Gestión de Procesos

Concepto

Para definir lo que es un proceso, hay que establecer la diferencia con el concepto de programa:

- **Un programa** : Es una entidad pasiva compuesta únicamente por un código y unos datos, es decir, tiene un listado fijo.
- **Un proceso** : Es una entidad activa, es el "programa" en ejecución.
- **Idle** : Un procesador de computadora es descrito como "idle" cuando no está siendo utilizado por ningún programa.

Cuando los programas hacen uso del tiempo Idle de la CPU, significa que estos se ejecutan en baja prioridad, de esta manera no impactan sobre otros programas que se ejecutan en prioridad normal.

La mayoría de los sistema operativos actuales muestran la tarea idle (en Windows es el proceso inactivo del sistema, mientras que en Linux suele ser Id zero), que es una tarea o proceso que abre el sistema cuando la computadora no tiene nada para hacer. Esta tarea tiene la prioridad más baja posible.

Concepto y criterios de planificación

La planificación hace referencia a un conjunto de políticas y mecanismos incorporados al SO que gobiernan el orden en que se ejecutan los trabajos que deben ser completados por el sistema informático. Un planificador es un módulo del SO que selecciona el siguiente trabajo a admitir en el sistema y el siguiente proceso que tomar el control sobre el procesador. El objetivo primario de la planificación es optimizar el rendimiento del sistema de acuerdo con los criterios considerados más importantes por los diseñadores del mismo.

Entre las medidas de rendimiento y los criterios de optimización mas habituales que los planificadores utilizan para llevar a cabo su labor se encuentran los siguientes:

Utilización del procesador:

La utilización del procesador es la fracción de tiempo promedio durante la cual el procesador está ocupado, es decir, la fracción de tiempo durante la cual el procesador se encuentra activo ejecutando algún proceso, bien de usuario, bien del propio SO. Con esta interpretación, la utilización del procesador puede ser medida con relativa facilidad, por ejemplo mediante un proceso nulo especial que se ejecute cuando ningún otro proceso pueda hacerlo. Una alternativa es considerar únicamente la operación en modo usuario y, por tanto, excluir el tiempo empleado para el SO.

En cualquier caso, el objetivo es mantener al procesador ocupado tanto tiempo como sea posible. De esta forma, se conseguirá que los factores de utilización de los restantes componentes también sean elevados obteniéndose con ello buenas medidas de rendimiento.

Productividad

La productividad se refiere a la cantidad de trabajo completada por unidad de tiempo. Un modo de expresarla es definiéndola como el número de trabajos de usuario ejecutados por una unidad de tiempo. Cuanto mayor sea este número, más trabajo aparentemente esta siendo ejecutado por el sistema.

Tiempo de retorno

El tiempo de retorno TR se define como el tiempo que transcurre desde el momento en que un trabajo o programa es remitido al sistema hasta que es totalmente completado por el mismo. Es decir, el tiempo de retorno TR es el tiempo consumido por el proceso dentro del sistema y puede ser expresado como la suma del tiempo de servicio o tiempo de ejecución + el tiempo de espera. TR = TS + TE .

Tiempo de espera

El tiempo de espera TE es el tiempo que un proceso o trabajo consume a la espera de la asignación de algún recurso o de que tenga lugar algún evento. En este tiempo también se incluyen el periodo de espera por la obtención del propio procesador debido a la competencia con otros procesos en un sistema con multiprogramación. Este tiempo es la penalización impuesta por compartir recursos con otros procesos y puede expresarse como el tiempo de retorno - el tiempo de ejecución efectivo. El tiempo de espera TE elimina la variabilidad debida a las diferencias en tiempos de ejecución del trabajo.

Tiempo de respuesta

El tiempo de respuesta en sistemas interactivos se define como el tiempo que transcurre desde el momento en que se introduce el ultimo carácter de una orden que desencadena la ejecución de un programa o transacción hasta que aparece el primer resultado en el terminal. Generalmente también se le denomina tiempo de respuesta de terminal.

En sistemas en tiempo real, el tiempo de respuesta es esencialmente una latencia y se define como el tiempo que transcurre desde el momento en que un suceso interno o externo es señalado hasta que se ejecuta la primera instrucción de su correspondiente rutina de servicio. A este tiempo suele denominarsele tiempo de respuesta al proceso.

Tipos de planificadores

En un SO complejo pueden coexistir tres tipos de planificadores: A corto, a medio y a largo plazo.

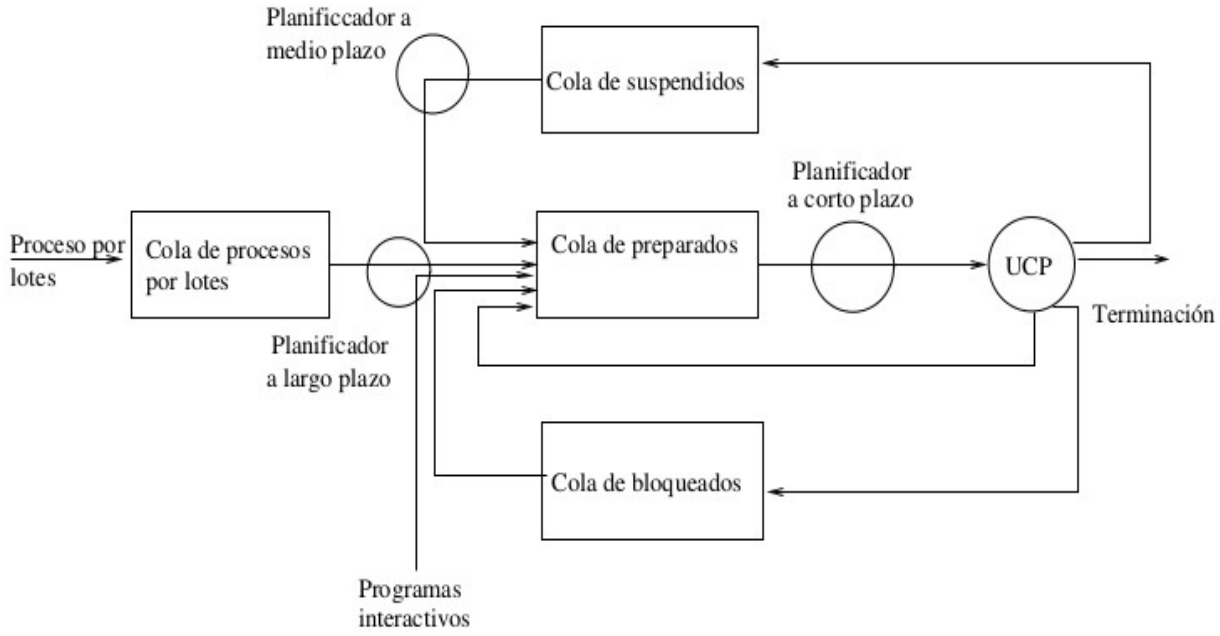
Planificador a largo plazo (PLP)

Su misión consiste en controlar la admisión de procesos nuevos al sistema. Cuando está presente este tipo de planificador, su objetivo principal es proporcionar una mezcla equilibrada de trabajos. El PLP decide cuando se da entrada al sistema a un nuevo proceso para que este sea ejecutado. Este proceso puede proceder de la respuesta al envío de un trabajo por lotes o bien a la orden de ejecución realizada por el usuario. En cierto modo, el PLP actúa como una válvula de admisión de primer nivel para mantener la utilización de recursos al nivel deseado. Es importante conseguir una administración equilibrada para saber cómo conjugar procesos interactivos que tienen retardos especiales con procesos por lotes que son una simple de cola de cálculo.

Por ejemplo, cuando la utilización del microprocesador puede admitir más trabajos, el planificador puede dar entrada al sistema a nuevos procesos y aumentar con ello la probabilidad de asignación de alguno de estos procesos al procesador. Por el contrario, cuando el tiempo para la utilización del procesador resulte alto y así se refleje en el deterioro en el tiempo de espera, el PLP puede optar por reducir la frecuencia de admisión de procesos a situación de preparado. El PLP es invocado generalmente cada vez que un trabajo completado abandona el sistema.

La frecuencia de llamada al PLP es, por tanto, dependiente del sistema y de la carga de trabajo pero generalmente es mucho más baja que para los otros dos tipos de planificadores.

Como resultado de esta no demasiada frecuente ejecución, el PLP puede incorporar algoritmos relativamente complejos y computacionalmente intensivos para admitir trabajos al sistema. En términos del diagrama de transición de estados de procesos, el PLP quedar´ a cargo de las transiciones del estado nuevo al estado preparado o listo.



Planificador a corto plazo (PCP)

Este planificador decide que procesos toman el control de la CPU. El PCP asigna el procesador entre el conjunto de procesos preparados residentes en memoria. Su principal objetivo es maximizar el rendimiento del sistema de acuerdo a con el conjunto de criterios elegidos. Al estar a cargo de la transición de estado preparado a ejecución, el PCP deberá ser invocado cuando se realice una operación de conmutación de procesos para seleccionar el siguiente proceso a ejecutar. En la práctica el PCP es llamado cada vez que un suceso interno o externo hace que se modifique alguna de las condiciones que definen el estado actual del sistema. Algunos de los sucesos que provocan una replanificación en virtud de su capacidad de modificar el estado del sistema son:

1. Tics de reloj, es decir, interrupciones basadas en el tiempo.
2. Interrupciones y terminaciones de operaciones de E/S.
3. Llamadas de operación al sistema operativo frente a llamadas de consulta.
4. Envío y recepción de señales.
5. Activación de programas interactivos.

En general, cada vez que ocurre uno de estos sucesos, el SO llama al PCP para determinar si deberá planificarse otro proceso para su ejecución.

Planificador a medio plazo (PMP)

El PMP tiene por misión traer procesos suspendidos a la memoria principal. Este planificador controla la transición de procesos en situación de suspendidos a situación de preparados. El PMP permanecerá inactivo mientras se mantenga la condición que dio lugar a la suspensión del proceso, sin embargo, una vez desaparecida dicha condición el PMP intenta asignar al proceso la cantidad de memoria principal que requiera y volver a dejarlo en situación de preparado. Para funcionar adecuadamente, el PMP debe disponer de información respecto a las necesidades de memoria de los procesos suspendidos, lo cual no es complicado de llevar a la práctica ya que el tamaño real del proceso puede ser calculado en el momento de suspenderlo almacenándose en el BCP.

Este planificador será invocado cuando quede espacio libre en memoria por la terminación de un proceso o cuando el suministro de procesos preparados quede por debajo de un límite especificado.

EJERCICIO 2

Si el tiempo de retorno o regreso (**R**) de un proceso P1 es 30 ms y el tiempo de ejecución real (**U**) es 10 ms.

¿Cuál es su tiempo de espera (E), la eficacia (E_f) y el rendimiento de este sistema (P)?

SOLUCION

En este ejercicio: R = T = S

A pesar de ser un solo proceso los valores de R y U son diferentes.

(S= Suma de tiempos de retorno de todos los procesos; T= tiempo de uso Total de la CPU)

(N = número de procesos = 1)

La expresión que relaciona el tiempo de retorno con el tiempo de espera es:

$$R = E + U$$

Por tanto,

$$E = R - U = 30 - 10 = \mathbf{20\ ms}$$

La eficacia será:

$$E_f = (U/T) \times 100 = (10/30) \times 100 = \mathbf{33.33\%}$$

El rendimiento será:

$$P = N/S = 1/30 = \mathbf{0.033}$$

EJERCICIO 3

Se tienen dos procesos P1 y P2 de tiempo de ejecución 25 y 30 ms respectivamente. El planificador a corto plazo actúa según el algoritmo **RR** o de prioridad circular con cuanto de 10 ms.

¿Cuál será el tiempo de retorno o regreso de P1 y P2?

SOLUCION

P1 = 25 ms P2=30 ms CUANTO = 10ms

	0 - 10	10 - 20	20 - 30	30 - 40	40 - 45	45 - 55
←	**P1**	**P2**	**P1**	**P2**	**P1**	**P2**
	10	10	10	10	5	10

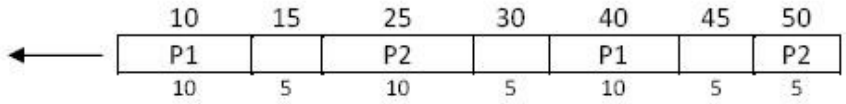
$R_{(P1)} = 10 + 10 + 10 + 10 + 5 = 45$ ms

$R_{(P2)} = 10 + 10 + 10 + 10 + 5 + 10 = 55$ ms

EJERCICIO 5

Sean dos procesos P1 (20ms) y P2 (15ms). El planificador a corto plazo actúa según el algoritmo de prioridad circular con cuanto de 10ms y tiempo de conmutación entre cada tarea de 5 ms.
Marcar el tiempo de retorno de P1 y P2.

SOLUCION



R(P1) = 40 ms
R(P2) =50 ms

Bloque de control de procesos o descriptor de un proceso

Un bloque de control de proceso (PCB o BCP) o descriptor de proceso es una estructura de datos que contiene cierta información importante acerca del proceso, incluyendo, entre otra:

- **Estado del proceso** : Determina la situación actual del proceso.
- **Identificador de proceso** : Es un número i entre 0-N (no se admiten negativos). Se le denomina PID y es el mismo durante toda la vida del proceso.
- **Registros de la CPU** : Almacenan información necesaria del proceso cuando se producen interrupciones como consecuencia de un cambio de estado, y así poder recuperarla para volver al estado original.
- **Límites de memoria** : Indican los límites de memoria utilizados por un determinado proceso para evitar que otros los invada. Además existe una zona de memoria donde se colocan los BCP.
- **Información del ´status´ de las operaciones de E/S** : Se refiere a las operaciones de E/S que el proceso está realizando, sobre que dispositivos si son de Entrada o de Salida. Estas operaciones, según van siendo atendidas, van siendo eliminadas del BCP.
- **Información del planificador de procesos** : Sobre el tipo de algoritmo de planificación que se esté utilizando.

El PCB es un almacenamiento central de información que permite al sistema operativo localizar toda la información clave sobre el proceso. Cuando el sistema operativo cambia la atención de la CPU entre los procesos, utiliza las áreas de preservación para mantener la información que necesita para reiniciar el proceso cuando consiga de

nuevo la CPU.

Debido a que los PCB deben ser manipulados con rapidez por el sistema operativo, muchos sistemas de computación contienen un registro de hardware que apunta siempre al PCB del proceso que está en ejecución, el proceso en curso.

Estados de un proceso

Durante su existencia, un proceso pasa por una serie de estados discretos. Varias circunstancias pueden hacer que un proceso cambie de estado.

Los estados más importantes en que puede encontrarse un proceso son:

- **Ejecutable (o activo)** : Si el proceso tiene asignada en ese momento la CPU.

- **Listo (preparado)** : Cuando el proceso podría usa una CPU, si hubiera una disponible.

- **Bloqueado (en espera)** : Si el proceso espera que ocurra algo (la terminación de una E/S por ejemplo) para poder ponerse en marcha.

Transiciones entre estados de un proceso

El S.O inicia la ejecución de un proceso nuevo que esta listo. Cuando este proceso no va a utilizar la CPU durante un tiempo : INTERRUPCIÓN (hay una llamada a E/S etc.) el S.O lo pone bloqueado hasta que acaba esa operación. Mientras tanto pasa a la CPU la ejecución de otro proceso que estuviera listo y realiza los mismos pasos: ejecuta hasta que este deja de usar la CPU, lo pone bloqueado si es necesario y pasa a ejecutar otro.

Cuando la acción del proceso que está bloqueado termina (acaba la E/S, etc.) el S.O lo pone otra vez en estado de listo, así hasta que termina con todo el proceso.

1. El proceso se bloquea en la entrada.
2. El planificador elige otro proceso.
3. El planificador elige este proceso.
4. La entrada se vuelve disponible.

El S.O deberá decidir que proceso de los que están listos pasará a ejecutable cuando la CPU quede libre, esta decisión lo hará a través de un Planificador de procesos (DISPACHER).

Interrupciones *(son necesarias para que un proceso pase de un estado a otro)*

- Permite interrumpir la ejecución de un proceso ante el acontecimiento de un suceso determinado, tomando el control el S.O.
- El Procesador puede ejecutar otras instrucciones mientras se está realizando alguna operación de E/S.

  Ante una interrupción, la forma de actuar es (FLIH o manejador de interrupciones de primer nivel):

  - Procesador registra situación exacta de la ejecución del trabajo actual.
  - Se desvía a la rutina de tratamiento de la interrupción.
  - La procesa según su naturaleza y, posteriormente, reanudar el mismo u otro trabajo de usuario.
  - La rutina de tratamiento de la interrupción forma parte del S.O. Este programa determina la naturaleza de la interrupción y ejecuta cuantas acciones sean necesarias.

Tipos de Interrupciones:

- Hardware.- Generadas por los dispositivos cuando finalizan una E/S.
- Software.- Generadas por el propio proceso. Hay 2 tipos:
  - Llamadas al sistema.- Se producen mediante la invocación de una instrucción que existe en casi todos los procesadores (Trap al S.O.). Se usan para pedir servicios al S.O. y se gestionan como si fuesen una interrupción.
  - Excepciones.- Se producen cuando el proceso produce un error grave que impide continuar con su ejecución (por ejemplo “división por cero”) → Finaliza proceso y cambiar a otro.

Operaciones sobre un proceso

Los S.O. con multitarea permiten numerosas operaciones dedicadas a la gestión de procesos. Entre ellas, las más importantes : creación, eliminación, obtención de información, modificación, retardo y activación.

- Creación de procesos : crear (id_proceso, atributos) ; El S.O. primero comprobará que no existen errores en la llamada (por ejemplo, comprueba que el procedimiento indicado no exista). A continuación se crea el proceso, se pasan los atributos como parámetros, se reserva memoria para el proceso (tanto para el BCP como para el código y los datos) y se añade a la cola de preparado.

- Eliminación de procesos : eliminar (id_proceso) ; Para eliminar un proceso es necesario que este sea hijo del proceso eliminador, ya que de no ser así podría volverse inconsistente el sistema. Una vez realizada la llamada, el S.O. verifica que no existen errores para a continuación liberar los recursos retenidos por el proceso. Finalmente se destruye el BCP.
- Obtención de información : inf_proc (id_proceso,est_BCP) ; Devolverá una copia del BCP del proceso requerido. El S.O. debe comprobar que no existen errores en los parámetros.
- Modificación de la información de un proceso : mod _inf (id_proceso, est_BCP) ; El proceso modificador debe enviar como parámetros el PID del proceso que modifica y un nuevo BCP que sustituya al actual. El S.O. comprobará los posibles errores producidos.
- Retardar un proceso : retardar (tiempo) ; El proceso que realiza esta llamada se autodetiene durante el tiempo indicado y pierde el control de la CPU durante ese tiempo. Los ciclos de reloj de espera se anotan en el BCP (utilizados posteriormente en la planificación de procesos). Finalmente, cuando el tiempo transcurre, el núcleo del S.O. introduce al proceso en la cola de procesos preparados para intentar ejecutarlo inmediatamente.
- Activar procesos retardados : activar (id_proceso); Esta función es privilegiada. El mecanismo para despertar procesos se activa en cada ciclo de reloj, recorriéndose la cola de procesos retardados para activarlos o disminuir en una unidad el número de pulsos de espera. Devuelve un código de error si el PID que se pasa no existe.

Planificación de procesos

Concepto

Es el encargado de pasar trabajos o procesos a la memoria principal y a la CPU, decidiendo qué procesos de los que están listos pasarán a ejecución cuando la CPU se libere de algún otro proceso (cuando este pase a estar bloqueado).

Sigue los pasos siguientes:

0. Decidir si se cambia el proceso.
1. Decidir mediante un algoritmo entre los procesos ejecutables, cual es el siguiente al que le asigna el procesador.
2. Salvar la información del proceso que se estaba ejecutando (si no lo hizo el tratamiento de interrupciones)
3. Cargar toda la información necesaria del proceso elegido.
4. Pasar el control al siguiente proceso.

Algoritmos de planificación

Esta planificación puede ser:

- Con desplazamiento(No apropiativo): Aquellos en los que unos procesos tienen prioridad sobre otros y cuando acaban de estar bloqueados y pasan a listos hacen que el S.O les dé paso a la CPU que abandona el proceso que tenía en ejecución, lo pone listo y pasa a ejecutar el que tiene prioridad.
- Sin desplazamiento (Apropiativo): Aquella en la cual, la CPU ejecuta un proceso hasta que lo termina o por alguna razón éste pasa a estar bloqueado.

Para evaluar el comportamiento de los distintos algoritmos se definen, entre otros, los siguientes parámetros:

- Tiempo de respuesta: Tiempo transcurrido desde que se hace una petición de CPU, hasta que es satisfecha.
- Tiempo de espera: Tiempo que está en estado de listo.
- Penalización: tiempo de respuesta= tiempo de finalización / tiempo de CPU
- Rendimiento: Número de trabajos terminados por unidad de tiempo.

Aquí tenemos la definición de los algoritmos.

- **FCFS (First Come First Served)** : Los procesos son servidos en orden a su llega a la cola de listos. Dichos procesos se ejecutarán hasta que terminen o se bloqueen. Su implementación es una cola FIFO.
  - Un problema de este algoritmo es el efecto *convoy*, se trata de que si los procesos mas largos son los primeros que llegan a la cola de espera y los mas pequeños llegan de ultimo, estos igual tendran que esperar a que se ejecuten los primeros que llegaron, que fueron los mas grandes.
  - Ejemplo:

Diagrama de Gantt

Tiempo de respuesta del caso 1 : 0 + 12 + 15 = 27 / 3 = 9

Tiempo de respuesta del caso 2 : 0 + 3 + 9 = 12 / 3 = 4

Caso del efecto Convoy:

Tiempo de respuesta caso 1 : 0 + 100 + 101 + 102 = 303 / 4 = 75,75

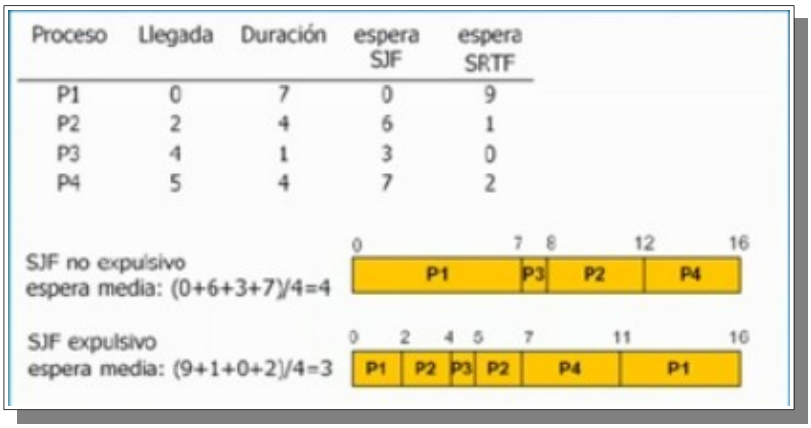
Tiempo de respuesta caso 2 : 0 + 1 + 2 + 3 = 6 / 4 = 1,5

- **SJF (Shortest Job First) :** Primero el más corto La CPU se asigna al proceso al que le queda menos tiempo para completar su ejecución y a igualdad de condiciones al primero que hay hecho la petición (FIFO).
  - Ejemplo :

    Calcular el tiempo medio de espera que resulta de aplicar:

    1. Un algoritmo SJF no expulsivo
    2. Un algoritmo SJF expulsivo

Proceso	Llegada	Duración
P1	0	7
P2	2	4
P3	4	1
P4	5	4

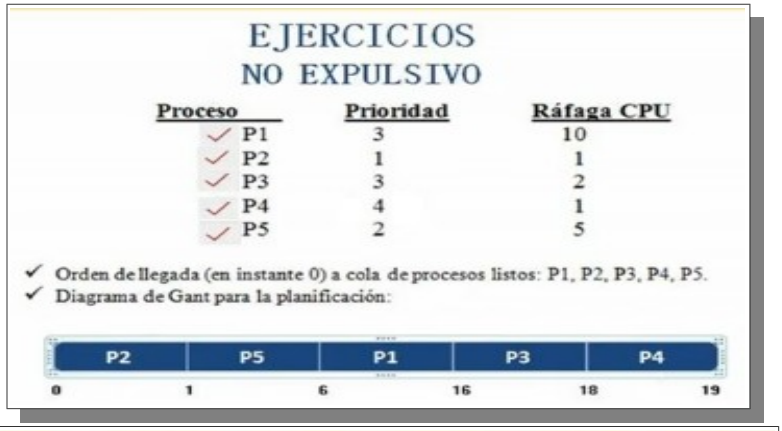


Proceso	Llegada	Duración	espera SJF	espera SRTF
P1	0	7	0	9
P2	2	4	6	1
P3	4	1	3	0
P4	5	4	7	2

- **PRIORIDAD** : A cada proceso se le asigna un determinado valor de prioridad, definida interna o externamente.

  La CPU se asigna al de mayor prioridad y a igualdad se sigue el algoritmo FCFS.

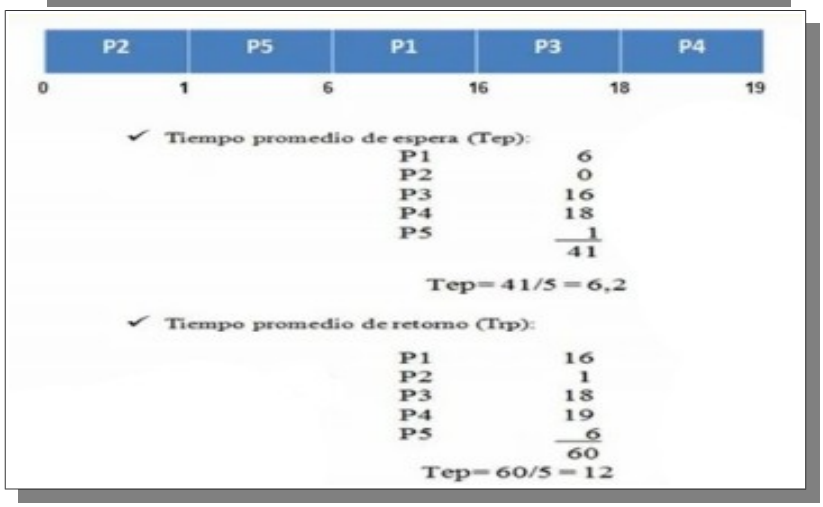
  En este ejemplo, el proceso de menor número (En la columna prioridad), es el de mayor prioridad.



EJERCICIOS
NO EXPULSIVO

Proceso	Prioridad	Ráfaga CPU
P1	3	10
P2	1	1
P3	3	2
P4	4	1
P5	2	5

- Orden de llegada (en instante 0) a cola de procesos listos: P1, P2, P3, P4, P5.
- Diagrama de Gant para la planificación:



- Tiempo promedio de espera (Tep):

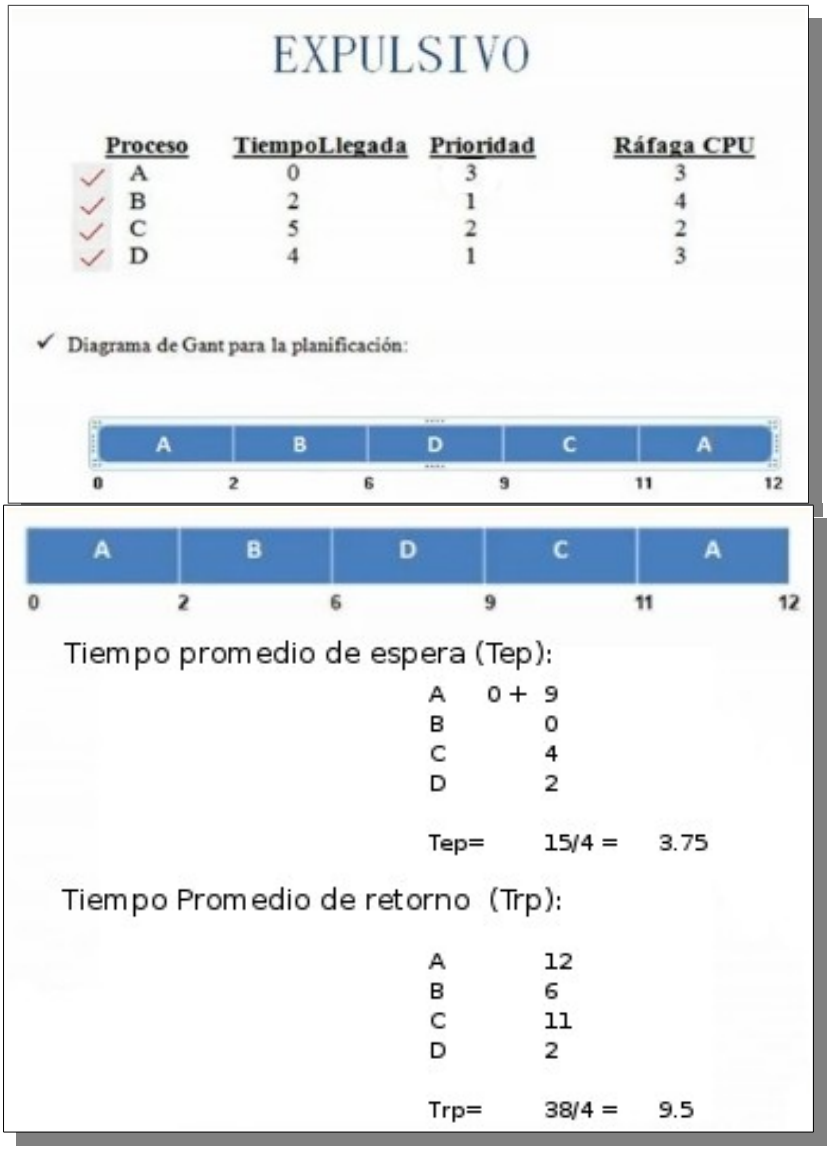
P1	6
P2	0
P3	16
P4	18
P5	1
	41

Tep= 41/5 = 6,2

- Tiempo promedio de retorno (Trp):

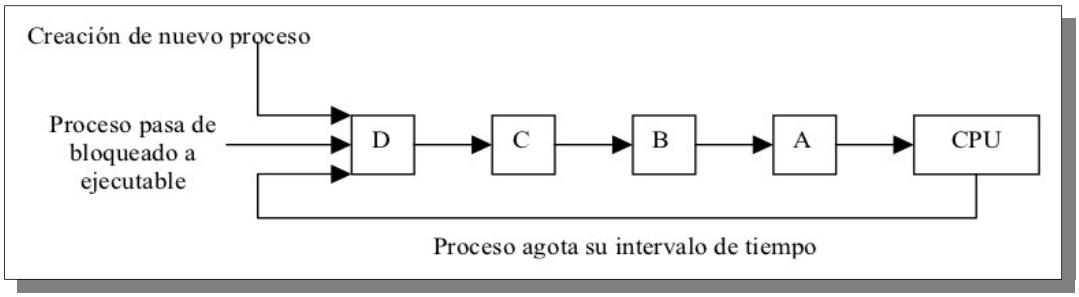
P1	16
P2	1
P3	18
P4	19
P5	6
	60

Tep= 60/5 = 12



- **EL MÁS CORTO PRIMERO CON DESPLAZAMIENTO** : Igual que el más corto primero, excepto que cuando un proceso llega a la cola de listos compara su tiempo con el que se esta ejecutando y si es menor se realiza un cambio de contexto.
- **ROUND ROBIN :** La CPU se asigna a los procesos en intervalos de tiempo llamados QUANTUM. Si el proceso finaliza o se bloquea antes de agotar el quantum, se toma el siguiente en la lista y se le asigna un nuevo intervalo de tiempo, si por el contrario agota su quantum pasa al final de la lista.

  La lista se trata como una cola FIFO.



- Ejemplo :

Proceso	Ráfaga CPU
P1	10
P2	5
P3	8
P4	3

q=4

✓ Diagrama de Gant para la planificación:

Tiempo promedio de espera:

P1 = 0 + 11 + 5 = 16

P2 = 4 + 11 = 15

P3 = 8 + 8 = 16

P4 = 12

P1 + P2 + P3 + P4 = 54

Tpe = 59 / 4 = 14,75

Tiempo promedio de retorno (tiempo en el que termina de ejecutarse un proceso:

P1 = 26

P2 = 20

P3 = 25

P4 = 15

P1 + P2 + P3 + P4 = 86

Tpr = 85 / 4 = 21,5

- **COLAS MULTIPLES** : La cola de ejecutables se divide en varias colas, los

trabajos son asignados permanentemente a una cola, según algún criterio (cada cola puede tener un algoritmo de asignación, además debe existir un algoritmo de asignación entre las colas).

- **COLAS MULTIPLES CON TRASPASO** : Usa el sistema de colas múltiples pero con la diferencia siguiente: Los trabajos pueden pasar de unas colas a otras.