

# UNIDAD 2



## SISTEMAS OPERATIVOS

### TEMA 3: Planificación de la CPU.

# ÍNDICE

|                                                                |           |
|----------------------------------------------------------------|-----------|
| <b>1. Unidad 2: Gestión de procesos.....</b>                   | <b>3</b>  |
| <i>Tema 3: Planificación de la CPU .....</i>                   | <i>3</i>  |
| <i>Objetivo: .....</i>                                         | <i>3</i>  |
| <i>Introducción:.....</i>                                      | <i>3</i>  |
| <b>2. Información de los subtemas .....</b>                    | <b>4</b>  |
| 2.1 Subtema 1: Conceptos.....                                  | 4         |
| 2.2 Subtema 2: Criterios de planificación.....                 | 7         |
| 2.3 Subtema 3: Algoritmos de planificación.....                | 8         |
| 2.4 Subtema 4: Planificación de sistemas multiprocesador ..... | 11        |
| <b>3. Preguntas de Comprensión de la Unidad .....</b>          | <b>13</b> |
| <b>4. Material Complementario .....</b>                        | <b>14</b> |
| <b>5. Bibliografía .....</b>                                   | <b>15</b> |

# 1. Unidad 2: Gestión de procesos

---

## Tema 3: Planificación de la CPU

### Objetivo:

Argumentar importancia de planificación de procesos dentro de la CPU.

### Introducción:

La base de los S.O multi-programados son las técnicas de planificación de la CPU, por medio de la conmutación de la CPU entre varios procesos, el S.O logra que el ordenador sea mucho más productivo. En esta unidad se presentará la planificación de la CPU empezando por conceptos básicos, criterios, diferentes algoritmos usados para esta finalidad y la planificación de los sistemas multiprocesador.

## 2. Información de los subtemas

### 2.1 Subtema 1: Conceptos

En un sistema monoprocesador, se ejecuta un proceso a la vez; entonces el otro proceso va a estar a la espera de que el CPU éste totalmente libre y se vuelva a planificar. La multi-programación tiene como objetivo tener simultáneamente varios procesos ejecutándose, con la finalidad de potenciar el uso del CPU.

Un proceso es ejecutado hasta el momento que tiene que quedar a la espera, por lo general porque se necesita efectuar una solicitud de E/S. En un sistema informático simple, el CPU se mantiene en reposo, por ende, el tiempo de espera es desperdiciado; ya que no se ejecuta ningún trabajo. Con la multi-programación, se usa aquel tiempo de forma productiva; ya que permanecen diferentes procesos a la vez en memoria, cuando algún proceso tiene que esperar, el S.O le cede el uso de la CPU a otro proceso. Aquel tipo de planificación es una de las funciones fundamentales del S.O, la mayor parte de los recursos del ordenador se van a planificar antes de ser usados. La CPU es un recurso fundamental en el ordenador, por ende, su planificación debe de ser correcta en el diseño del S.O.

#### Ciclo de ráfagas de CPU y de E/S

“La correcta planificación de la CPU va a depender de la prioridad de los procesos: la ejecución de un proceso tiene un ciclo de ejecución en la CPU y una espera de E/S; los procesos se van alternando entre los ambos estados.”  
(RAMEZ, 2010)

Al ejecutarse un proceso empieza con una ráfaga de CPU, continuando con una de E/S, por consiguiente, otra de CPU, después otra de E/S, etc. En definitiva, la ráfaga final de CPU finaliza con una petición al sistema y así concluir la ejecución.

La duración de las ráfagas de la CPU va a variar ampliamente de un proceso a otro y de un ordenador a otro, presentan una curva de tipo exponencial, con una cantidad mayor de ráfagas de CPU cortas y ráfagas largas con una cantidad menor. Por lo general, un aplicativo restringido por E/S muestra muchas ráfagas de CPU cortas.

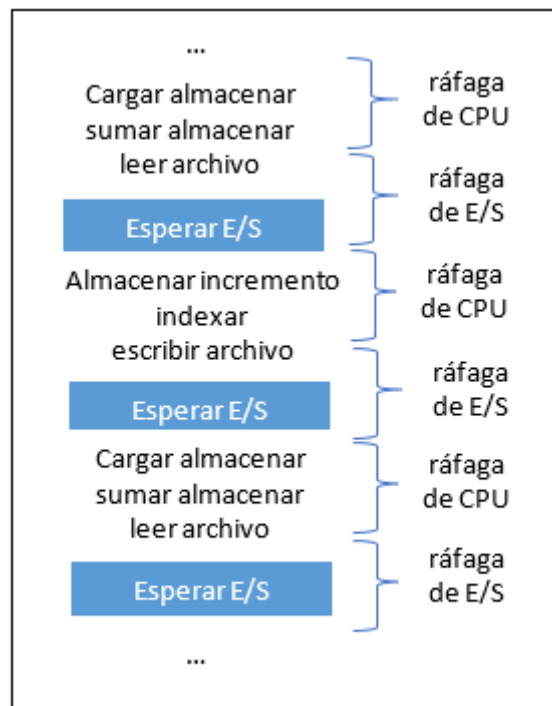


Gráfico 1: Ráfagas de CPU y de E/S. (MCHOES, 2011)

## Planificar de la CPU

En el momento que la CPU queda inactiva, el S.O tiene que seleccionar un proceso de los que están en cola de los procesos preparados para su ejecutarse. El planificador de la CPU escoge el proceso. El planificador selecciona un proceso de los que se encuentran en memoria ya preparados para así ejecutarlo y le estipula a la CPU el proceso.

Se consideran los diferentes algoritmos de planificación, una cola de procesos ya preparados puede realizarse en una cola FIFO, ya sea un árbol o lista enlazada desordenada, una cola prioritaria. No obstante, todos los procesos que están en la cola de los procesos ya preparados se colocan en una fila la espera de la oportunidad de poder ser ejecutados en el CPU. Los registros almacenados en las diferentes colas por lo general son bloques de proceso que van a detallar los procesos.

## Planificación apropiativa

Teniendo las circunstancias que se mencionaron puede que exista la necesidad de decidir sobre la planificación de la CPU, (SILBERSCHATZ, 2006) nos menciona algunas:

1. “Cuando un proceso cambia de estado de ejecución al estado de espera.
2. Cuando un proceso cambia de estado de ejecución al estado preparado.
3. Cuando un proceso cambia de estado de espera al estado preparado.
4. Cuando un proceso finaliza.”

En las circunstancias 1 y 4, se debe hacer lo siguiente en la planificación: se debe seleccionar un nuevo proceso para ejecutarse. En las circunstancias 2 y 3 hay la alternativa de planificar un nuevo proceso o no.

Si la decisión de planificación sólo tiene cabida en las situaciones 1 y 4, entonces, el esquema de planificación resulta ser cooperativo; en el caso contrario, se trataría de uno apropiativo. En una planificación cooperativa, desde el momento que se le asigna la CPU a un proceso, dicho proceso se va a mantener en el CPU hasta que se libere ya sea por la finalización del proceso o por el cambio al estado de espera. Este método fue usado en el S.O Microsoft Windows 3.x, desde el Windows 95 introdujo la planificación apropiativa y sus posteriores versiones del S.O Windows han utilizado este tipo de planificación. La planificación apropiativa tiene un coste relacionado con el acceso a datos compartidos. La planificación cooperativa afecta al diseño del kernel del S.O.

## Despachador

En la planificación de la CPU hay otro componente que está implicado es el despachador, el cual es el módulo que se encarga de brindar el control del CPU a los procesos que han sido escogidos por parte del planificador de la CPU o planificador a corto plazo. Se involucra lo siguiente:

- Cambio de contexto.
- Cambio al modo usuario.
- Salto a la posición correcta dentro del aplicativo de usuario y así el reinicio del mismo.

El despachador debe ser rápido, ya que es invocado en cada cambio de estado de los procesos. El tiempo que el despachador demora en interrumpir un proceso y ejecutar otro se denomina latencia de despacho.

## 2.2 Subtema 2: Criterios de planificación

### Su Criterios de Planificación

Todos los algoritmos de planificación de la CPU tienen diferentes propiedades, y la selección de algún algoritmo puede que favorezca a una clase de procesos por encima de otros. Para su elección se debe tener en cuenta las propiedades de los diferentes algoritmos.

Los siguientes criterios los menciono (SILBERSCHATZ, 2006):

- **“Utilización de la CPU:** se quiere conservar la CPU lo más ocupada que sea posible. En el concepto, el uso de la CPU es definido entre el rango de 0 y el 100%; en un sistema, va a variar entre el 40% en un sistema cargado ligeramente y el 90% para un sistema sumamente manipulado.
- **Tasa de procesamiento:** Si la CPU está ocupada con la ejecución de procesos, entonces está realizando alguna tarea. La medida de la cantidad de tarea o trabajos es el número de los procesos que son completados por unidad de tiempo, a esto se conoce como tasa de procesamiento.
- **Tiempo de ejecución:** En un proceso individual, un criterio muy importante es cuánto demora la ejecución de aquel proceso. El intervalo que va desde el momento en que se establece un proceso para ser ejecutado hasta el momento que se ha completado se denomina como tiempo de ejecución.
- **Tiempo de espera:** El algoritmo de planificación de la CPU afecta al período de tiempo que un proceso dedica en esperar en la cola de procesos preparados. El tiempo de espera es la suma de todos estos períodos dedicados a esperar.
- **Tiempo de respuesta:** Es el tiempo que el proceso demora en comenzar a responder, no el tiempo que demora en remitir a la salida los datos de respuesta, este tiempo se limita por la rapidez del dispositivo de salida.”

El objetivo es potenciar el uso de la CPU y la tasa de procesamiento y disminuir el tiempo de ejecución, espera y respuesta. Por lo general en el mayor de los casos, se optimiza algún tipo de valor promedio, No obstante, en circunstancias específicas, es favorable optimizar los valores máximos y mínimos.

## 2.3 Subtema 3: Algoritmos de planificación

La planificación de la CPU afronta la cuestión de la decisión de qué proceso de la cola de procesos ya preparados se tiene que asignar a la CPU. Hay muchos algoritmos relacionados a la planificación de la CPU:

### Planificación FCFS

El algoritmo FCFS (primero en llegar, primero en ser servido; first come-first served) es el más simple, en este esquema se le va a asignar primero la CPU al proceso que lo solicite primero. La ejecución del algoritmo FCFS se asimila con una cola FIFO. En el momento que un proceso entra a la cola de procesos ya preparados, el PCB se va a colocar en al último. En el momento que la CPU está liberada, se le fija al proceso que se encuentra al inicio de la cola y aquel proceso que sigue a la ejecución procede a eliminarse de la cola. El tiempo de espera promedio es bastante largo.

Ejemplo definido por (MCHOES, 2011):

“Calcular el tiempo de espera, tiempo de retorno y tiempo medio de espera aplicando el algoritmo FCFS suponiendo que los siguientes procesos llegan al mismo tiempo y en el siguiente orden: P1, P2, P3. Y en el orden de llegada: P2, P3, P1.”

| PROCESO | DURACIÓN |
|---------|----------|
| P1      | 8        |
| P2      | 3        |
| P3      | 2        |

Tabla 1: Ejemplo planificación FCFS. (MCHOES, 2011)

Diagrama de Gantt



Gráfico 2: Diagrama de Gantt del ejemplo. (MCHOES, 2011)

Tiempo de espera: P1=0; P2=8; P3=11 milisegundos

Tiempo de retorno: P1=8; P2=11; P3=13 milisegundos

Tiempo de espera medio:  $(0+8+11) / 3 = 6.3$  milisegundos

Si P1 llegaba al último, los tiempos serían sido mejores y su espera media seria  $(0+3+5) / 3 = 2.6$  milisegundos.



Gráfico 3: Diagrama de Gantt del ejemplo.  
(MCHOES, 2011)

### Planificación SJF

El algoritmo de planificación con selección del trabajo más corto, relaciona con cada proceso el tiempo de duración de la sucesiva ráfaga de CPU del proceso. En el momento que la CPU se encuentra libre se le va asignar al proceso que tiene la sucesiva ráfaga de CPU más corta. Si las ráfagas de CPU siguientes de dos procesos llegan a ser parejas, entonces se hace uso del algoritmo FCFS para romper la igualdad.

Ejemplo:

| PROCESO | TIEMPO DE RÁFAGA |
|---------|------------------|
| P1      | 5                |
| P2      | 7                |
| P3      | 6                |
| P4      | 2                |

Tabla 2: Ejemplo planificación SJF  
(MCHOES, 2011)

Diagrama de Gantt:

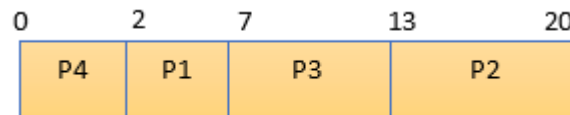


Gráfico 4: Diagrama de Gantt del ejemplo  
(MCHOES, 2011)

El tiempo medio de espera:  $(0+2+7+13) / 4 = 5.5$  milisegundos. Si se hubiera usado el algoritmo FCFS el tiempo medio de espera es de 8.75 milisegundos. Por ende, el algoritmo SJF es posiblemente recomendable en relación con el tiempo medio de espera mínimo para un conjunto de procesos dados.

“La planificación SJF es usada como mecanismo de planificación a largo plazo, esta planificación resulta ser optima, pero no puede ser implementada en el nivel de la planificación de la CPU a corto plazo porque no hay manera de conocer cuál es el tiempo de duración de la sucesiva ráfaga de CPU.” (MCHOES, 2011)

### Planificación por prioridades

“El algoritmo SJF es un caso especial de algoritmos de planificación por prioridades general, a cada proceso se le va asociar una prioridad y la CPU se le asigna al proceso que tenga la prioridad más alta. Los procesos que tengan la misma prioridad se

planifican con el algoritmo FCFS. En el algoritmo SJF mientras más larga sea la ráfaga de CPU, menor será la prioridad y viceversa.” (SILBERSCHATZ, 2006)

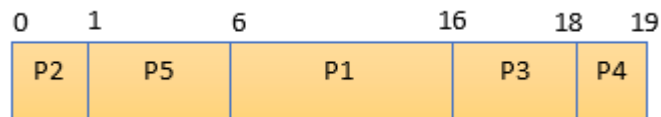
Por lo general las prioridades son indicadas por medio de un rango de números fijos, ejemplo de 0 a 7 o de 0 a 4095. No obstante, no hay un acuerdo general sobre si 0 es la prioridad más alta o la más baja, algunos sistemas emplean los números bajos para indicar una prioridad alta, en este caso asumiremos los números bajos como prioridad alta.

Ejemplo definido por (SILBERSCHATZ, 2006): “Teniendo el siguiente conjunto de procesos y suponiendo que llegan en el instante 0 en este orden: P1, P2,...,P5, la duración de ráfagas de CPU están especificadas en milisegundos.”

| PROCESO   | TIEMPO DE RÁFAGA | PRIORIDAD |
|-----------|------------------|-----------|
| <b>P1</b> | 10               | 3         |
| <b>P2</b> | 1                | 1         |
| <b>P3</b> | 2                | 4         |
| <b>P4</b> | 1                | 5         |
| <b>P5</b> | 5                | 2         |

*Tabla 3: Ejemplo planificación por prioridades  
(SILBERSCHATZ, 2006)*

Diagrama de Gantt:



*Gráfico 5: Diagrama de Gantt del ejemplo  
(SILBERSCHATZ, 2006)*

El tiempo medio de espera:  $(0+1+6+16+18) / 5 = 8.2$  milisegundos.

Las prioridades pueden determinarse internamente o externamente:

- Internamente: utilizan un valor mensurable para el cálculo de la prioridad de un proceso.
- Externamente: se especifican en función a criterios externos del S.O.

Un problema que es importante de estos algoritmos es el bloque indefinido, un proceso que está preparado para su ejecución, pero está esperando a acceder a la CPU se puede considerar como bloqueado, el algoritmo podría desertar a varios procesos de baja prioridad en una espera indefinida. Para solucionar este inconveniente se aplica el mecanismo de envejecimiento, radica en incrementar lentamente la prioridad de los procesos que están a la espera con un tiempo muy largo.

## 2.4 Subtema 4: Planificación de sistemas multiprocesador

Si se tiene variados procesadores, se comparte la carga; no obstante, la planificación es un poco compleja, se presentarán diferentes cuestiones referentes a la planificación multiprocesador. En los procesadores iguales, es decir homogéneos, se pueden utilizar cualquiera de los procesadores que se encuentran disponibles para la ejecución de cualquier proceso de la cola, pero inclusive en estos procesadores pueden existir limitaciones que llegan a afectar la planificación.

### Métodos de planificación en los sistemas multiprocesador

Un método radica en que las decisiones de la planificación, procesamiento de E/S y otras actividades van a ser tramitadas por el mismo procesador, el cual será el servidor maestro, y los otros procesadores sólo van a ejecutar código de usuario.

“Este método conocido como el **multiprocesamiento asimétrico** es simple, porque hay un solo procesador que va acceder a las estructuras de datos, y así se reduce la necesidad de compartir datos.” (RAMEZ, 2010)

Otro método usa el **multiprocesamiento simétrico**, en donde cada uno de los procesadores se auto-planifica, todos podrían encontrarse en una cola frecuente, o cada uno puede tener su cola, independientemente la planificación se realiza cuando el planificador de cada procesador revise la cola de los procesos y realice la selección de un proceso para ser ejecutado. Si se tiene muchos procesadores que estén intentando acceder a una misma estructura de datos para actualizarla, el planificador debe programarse con mucho cuidado para asegurar que dos de sus procesadores no elijan el mismo proceso y que no se pierdan procesos de la cola.

### Afinidad al procesador

Cuando un proceso se ha ejecutado en un procesador: los datos a los que el proceso recientemente ha accedido se va almacenar en la cache del procesador, dando como resultado que los siguientes accesos a memoria por parte del proceso se satisfagan solo consultando la memoria caché. Si el proceso va a migrar a otro procesador: el contenido de la memoria caché del procesador de origen tiene que invalidarse y la caché del procesador de destino tiene que ser rellenada. Invalidar y rellenar las memorias caché tienen un alto coste, debido a esto los sistemas de multiprocesamiento simétrico intentan que se evite la migración de proceso de un procesador a otro, en su lugar hacen el

intento de mantener cada proceso en ejecución con el mismo procesador. Esto se denomina afinidad al procesador.

La afinidad al procesador toma diferentes maneras. En el momento que el S.O tiene la orden de intentar que un proceso se mantenga en ejecutado en el mismo procesador, pero no se garantiza que lo realice, entonces esto se considera afinidad suave. Unos sistemas como Linux brindan llamadas al sistema que resisten la afinidad dura, está le va a permitir a un proceso determinar, que no tienen que migrar a otros procesadores.

### Equilibrio de carga

En los sistemas de multiprocesamiento simétrico es sumamente importante que se mantenga equilibrada la carga de trabajo entre todos los procesadores. Hay dos métodos para el equilibrio de carga:

- **Migración comandada:** una tarea específica corrobora periódicamente la carga de cada procesador y en el caso de que encuentre un desequilibrio, se encarga de distribuir igualitariamente la carga, moviendo procesos de los procesadores que se encuentran sobrecargados a los procesadores que estén menos ocupados.
- **Migración solicitada:** se produce cuando un procesador que está inactivo extrae de un procesador cargado una tarea que está en espera.

Las migraciones comandadas y solicitadas no deben de ser mutuamente excluyentes y regularmente se van a implementar en paralelo en los sistemas de equilibrio de carga.

### Mecanismos multihilos simétricos

Los sistemas de multiprocesamiento simétricos permiten que varios hilos se logren ejecutar de manera simultánea, ya que brindan varios procesadores físicos. Una alternativa estratégica es el brindar varios procesadores lógicos, en lugar de los procesadores físicos. Esta alternativa estratégica se denomina mecanismo multihilo simétrico, también se conoce como tecnología hiperhilo.

La idea de este mecanismo es crear varios procesadores lógicos sobre un procesador físico, de esta manera se presente una vista de varios procesadores lógicos al S.O, inclusive en un sistema con un procesador físico. Cada procesador lógico es responsable de su tratamiento de las interrupciones, esto significa que las interrupciones son dadas y son gestionadas por los procesadores lógicos.

### 3. Preguntas de Comprensión de la Unidad

---

**¿Cuáles son las circunstancias para la toma de decisión de la planificación sobre la CPU?**

El autor (SILBERSCHATZ, 2006) menciona algunas:

- “Cuando un proceso cambia de estado de ejecución al estado de espera.
- Cuando un proceso cambia de estado de ejecución al estado preparado.
- Cuando un proceso cambia de estado de espera al estado preparado.
- Cuando un proceso finaliza.”

**¿Qué involucra el despachador?**

- Cambio de contexto.
- Cambio al modo usuario.
- Salto a la posición correcta dentro del aplicativo de usuario y así el reinicio del mismo.

**¿En qué consiste el tiempo de ejecución?**

En un proceso individual, un criterio muy importante es cuánto demora la ejecución de aquel proceso. El intervalo que va desde el momento en que se establece un proceso para ser ejecutado hasta el momento que se ha completado se denomina como tiempo de ejecución.

**¿Cuáles son los algoritmos de planificación mencionados?**

1. Planificación FCFS
2. Planificación SJF
3. Planificación por prioridades

**¿De qué se trata la migración comandada?**

Una tarea específica corrobora periódicamente la carga de cada procesador y en el caso de que encuentre un desequilibrio, se encarga de distribuir igualitariamente la carga, moviendo procesos de los procesadores que se encuentran sobrecargados a los procesadores que estén menos ocupados.

## 4. Material Complementario

---

*Los siguientes recursos complementarios son sugerencias para que se pueda ampliar la información sobre el tema trabajado, como parte de su proceso de aprendizaje autónomo:*

**Videos de apoyo:**

<https://www.youtube.com/watch?v=lWs-RJAsdlw>

**Bibliografía de apoyo:**

JULIA, P. (2012). *Sistemas Operativos*. Universidad Técnica Particular de Loja.

**Links de apoyo:**

<https://www.infor.uva.es/~figonzalez/apuntes/Tema5.pdf>

## 5. Bibliografía

---

- » MCHOES, A. M. (2011). *Sistemas Operativos*. México: Cengage Learning.
- » RAMEZ, E. (2010). *Sistemas Operativos*. México: Mc Graw Hill.
- » SILBERSCHATZ, A. (2006). *Fundamentos de sistemas operativos*. México: McGraw-Hill.