

# Sistemas Operativos

## Planificación de los Procesos

### Objetivos

#### Que Ud. logre

Comprender el concepto de la Planificación.

Conocer las diferentes técnicas de Planificación de los Procesos.

Evaluar estas Planificaciones

Comprender como distribuye la carga de trabajo de procesos el Sistema Operativo.

Conocer los estados que un proceso puede tener en función de la Planificación.

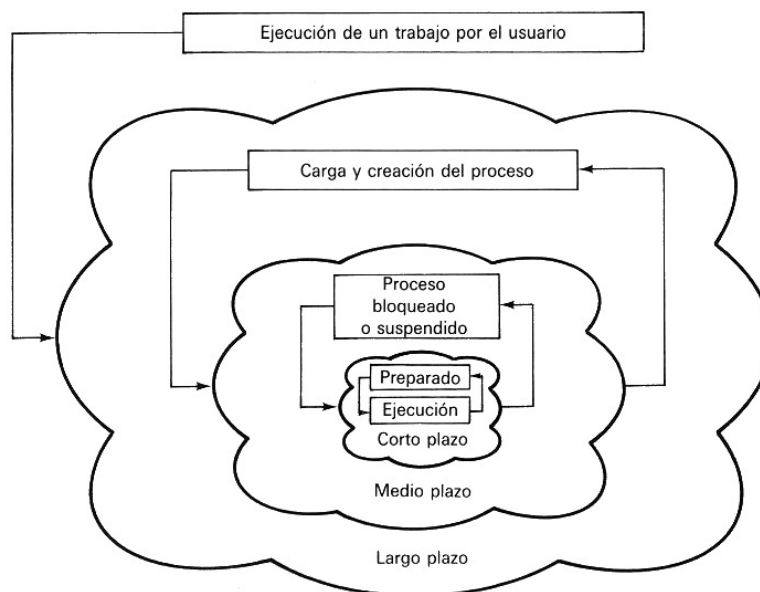
Integrar los conocimientos adquiridos en el tema Procesos con los de la Planificación.

### Planificación del procesador

#### Introducción

En este capítulo se estudian las distintas políticas y mecanismos más comunes que poseen los Sistemas Operativos actuales para realizar la gestión del procesador que se conoce con el nombre de planificación, cuyo objetivo principal es el de dar un buen servicio a todos los procesos que existan en un momento dado en el sistema.

En general, se distinguen varios niveles de planificación, como se refleja en la figura. Aunque la mayoría de los autores identifica tres.



#### Planificación a largo plazo (planificador de trabajos).

Decide cuál será el próximo trabajo que se va a ejecutar. Este nivel sólo existe en los sistemas de proceso por lotes donde la decisión se basa en las necesidades de recursos y su disponibilidad. En los sistemas de tiempo compartido tiene como única misión cargar los programas que se desean ejecutar en memoria. Este nivel es, por tanto, el encargado de crear los procesos.

Esta planificación determina cuáles son los programas admitidos en el sistema. De este modo se controla el grado de multiprogramación. Una vez admitido el programa se convierte en proceso y es añadido a la cola del planificador a corto plazo.

La decisión de cuándo crear un nuevo proceso viene dada, en general, por el grado de multiprogramación. Cuantos más procesos se crean, menor será el porcentaje de tiempo en el que cada proceso puede ejecutar simultáneamente. Con lo cual el planificador a largo plazo puede limitar el grado de multiprogramación para ofrecer un buen servicio al conjunto de procesos activos.

### **Planificación a medio plazo (planificador de swapping).**

Decide si un proceso que está en ejecución en estado bloqueado o suspendido debe ser extraído de la memoria temporalmente. Posteriormente, cuando el sistema se encuentre más descargado, devolverá dicho proceso a la memoria y al estado de ejecución.

Algunos sistemas operativos, como los de tiempo compartido, pueden presentar un nivel intermedio adicional de planificación. La idea clave de un planificador a medio plazo es que en ocasiones puede interesar eliminar procesos de la memoria y reducir de este modo el grado de multiprogramación.

Esta técnica se conoce con el nombre de swapping y será estudiada al tratar la gestión de la memoria. Sólo existe en sistemas de tiempo compartido y en aquellos que tienen gestión de memoria virtual. Este nivel, por tanto, gestiona los procesos suspendidos en espera de algún recurso no disponible en el momento de la suspensión.

### **Planificación a corto plazo (planificador del procesador).**

La ejecución de un proceso consiste en un ciclo de ejecución del procesador y una espera de E/S, y los procesos se alternan entre estos dos estados.

La ejecución del proceso se inicia con una ráfaga de procesador y a ésta le sigue una ráfaga de E/S, le sigue una ráfaga de procesador y una más de E/S, etc. Finalmente la última ráfaga de procesador terminará con una solicitud al sistema para que concluya la ejecución del proceso.

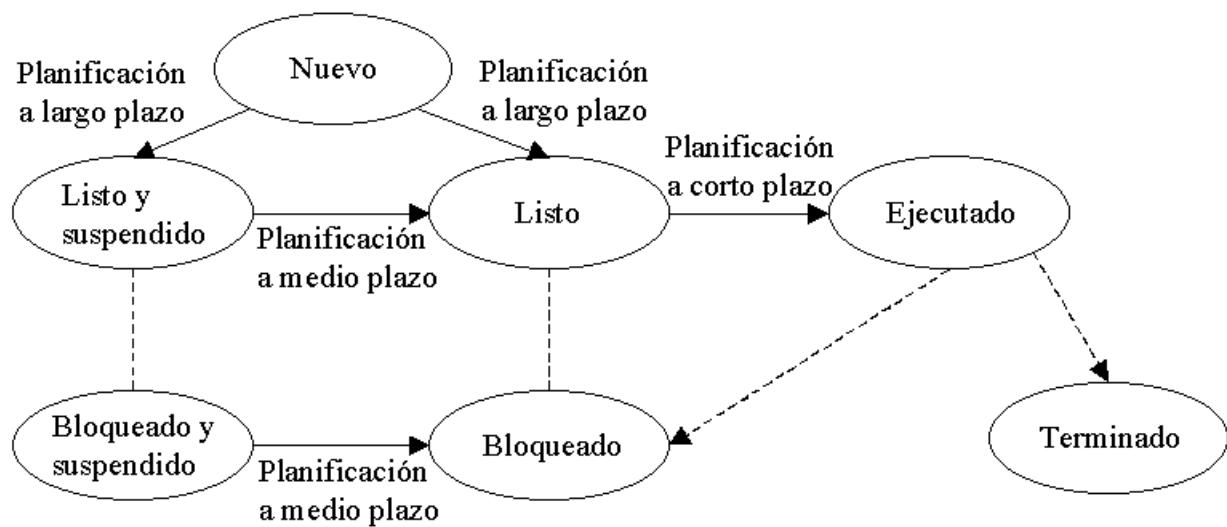
La duración de estas ráfagas de procesador y de E/S va a determinar el tipo de carga al sistema que proporciona cada proceso. Así por ejemplo un programa limitado por E/S, tendrá muchas ráfagas cortas de procesador, mientras que un programa limitado por el procesador tendrá pocas ráfagas de larga duración de procesador. La distribución de estos procesos puede ser muy importante al seleccionar un algoritmo de planificación de corto plazo.

Siempre que el procesador quede inactivo, el sistema operativo debe seleccionar para su ejecución uno de los procesos de la cola de procesos listos. El proceso de selección es realizado por el planificador a corto plazo. El planificador selecciona uno de los procesos en memoria que están listos y se lo asigna al procesador.

La cola de procesos listos no es necesariamente una cola primero en entrar primero en salir (FIFO – First In First Out), sino que como veremos cuando expliquemos los diferentes algoritmos de planificación, la cola de procesos listos puede ser implementada de formas diferentes. Sin embargo, conceptualmente todos los procesos de la cola de procesos listos están en fila esperando la oportunidad de ejecutarse en el procesador.

Es el encargado de decidir cómo y cuándo tendrá acceso al procesador un proceso que está preparado para utilizarlo. Por tanto, lleva a cabo las funciones de la multiprogramación, estando siempre residente en memoria y ejecutándose con mucha frecuencia; por ello, debe ser de ejecución muy rápida. En este nivel, que es el que vamos a estudiar en este capítulo, es donde se debe dar un buen servicio a los procesos interactivos para que el usuario no perciba, o lo haga en pequeño grado, que está compitiendo por el procesador junto con otros usuarios.

Sólo queda añadir que toda política de gestión debe ser justa al tiempo de ofrecer un rendimiento del sistema lo más aceptable posible.



En la figura superior observamos la relación existente entre los estados de los procesos y la relación con la planificación en los diferentes niveles.

### Objetivos de la Planificación

Las políticas de planificación intentan cubrir los siguientes objetivos:

**Justicia.** La política debe ser lo más justa posible con todo tipo de procesos, sin favorecer a unos y perjudicar a otros.

**Máxima capacidad de ejecución.** Debe dar un servicio aceptable para que todos los trabajos se realicen lo más rápidamente posible. Esto se logra disminuyendo el número de cambios de proceso.

**Máximo número de usuarios interactivos.** En los sistemas de tiempo compartido se tratará de que puedan estar trabajando el mayor número de usuarios simultáneamente.

**Predecibilidad.** La política de planificación debe concebirse de tal forma que en todo momento pueda saberse cómo será su ejecución.

**Minimización de la sobrecarga.** La computadora debe tener poca sobrecarga y a que ésta incide directamente sobre el rendimiento final del sistema: a menor sobrecarga, mayor velocidad de proceso. Por ello, los cambios de contexto deben minimizarse.

**Equilibrio en el uso de recursos.** Para obtener un buen rendimiento en el uso de los recursos y que éstos estén ocupados equitativamente el mayor tiempo posible.

**Seguridad de las prioridades.** Si un proceso tiene mayor prioridad que otro, éste debe ejecutarse más rápidamente.

Los objetivos enunciados pueden entrar en ocasiones en contradicción; por ello es necesario llegar a una situación de compromiso entre todos los objetivos para conseguir del sistema operativo un buen rendimiento y un buen servicio.

### Criterios a tener en cuenta

Los criterios que se deben tener en cuenta a la hora de elegir o diseñar un algoritmo de planificación son los siguientes:

**Tiempo de respuesta.** Velocidad con que el ordenador da respuesta a una petición. Depende mucho de la velocidad de los dispositivos de entrada/salida.

**Tiempo de servicio.** Es el tiempo que tarda en ejecutarse un proceso, donde se incluye el tiempo de carga del programa en memoria, el tiempo de espera en la cola de procesos preparados, el tiempo de ejecución en el procesador y el tiempo consumido en operaciones de entrada/salida.

**Tiempo de ejecución.** Es idéntico al tiempo de servicio menos el tiempo de espera en la cola de procesos preparados; es decir, es el tiempo teórico que necesitaría el proceso para ser ejecutado si fuera el único presente en el sistema.

**Tiempo de procesador.** Es el tiempo que un proceso está utilizando el procesador sin contar el tiempo que se encuentra bloqueado por operaciones de entrada/salida.

**Tiempo de espera.** Es el tiempo en que los procesos están activos pero sin ser ejecutados, es decir, los tiempos de espera en las distintas colas.

**Eficiencia.** Se refiere a la utilización del recurso más caro en un sistema, el procesador, que debe estar el mayor tiempo posible ocupado para lograr así un gran rendimiento.

**Rendimiento.** Es el número de trabajos o procesos realizados por unidad de tiempo, que debe ser lo mayor posible.

Ahora bien, ¿qué algoritmo de planificación se debe elegir para un sistema determinado? Será misión del diseñador del sistema operativo la elección de los mecanismos apropiados para que la política elegida partiendo de los criterios anteriores sea satisfactoria y ofrezca un alto rendimiento global.

## Medidas

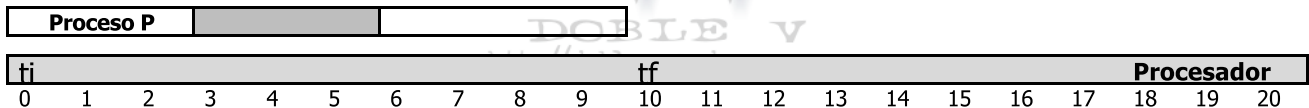
Para estudiar el comportamiento de las distintas políticas de planificación, definiremos dos medidas relacionadas entre sí que nos indiquen cómo estamos tratando un proceso concreto.

Considerando  $t$  como tiempo de un proceso  $P$ , necesita estar en ejecución para llevar a cabo trabajo,  $t_i$  el instante en que el usuario da la orden de ejecución del proceso y  $t_f$  el instante en que el proceso termina su ejecución. En función de estos datos, tendremos las siguientes medidas para cada proceso:

Tiempo de servicio (T):  $T = t_f - t_i$

Tiempo de espera (E):  $E = T - t$

Visualmente sería:



Para este ejemplo  $t_i$  es 0 ya que en ese instante se invoca el proceso para su ejecución. Es importante recordar que será la planificación quien decida si empezar inmediatamente su ejecución o no. Por tanto en  $t_i$  el proceso podría esperar.

En el gráfico la parte sombreada es el tiempo de espera del procesador ya que posiblemente este siendo utilizado por algún otro proceso.

Siguiendo con el ejemplo  $t_f$  es 10, en consecuencia  $T$  es 10. El tiempo de espera es 3. Ya que durante ese período el proceso está en espera para el uso del procesador.

A partir de los dos valores anteriores, podemos establecer una relación que nos permite evaluar la actuación de la política establecida en lo que se denomina índice de servicio (I).

$$I = t/T$$

En nuestro ejemplo el índice de servicio sería de un 70 %, ya que el tiempo  $T$  es 10 y el tiempo de ejecución es 7.

Este índice representa el porcentaje de tiempo que el proceso está en ejecución respecto al tiempo de vida del mismo en el sistema.

En caso de que sólo exista un proceso ejecutándose en el sistema, según el valor del índice de servicio, podemos decir que:

- Cuando  $I$  sea próximo a la unidad, el proceso está limitado por proceso. .
- Si  $I$  tiene un valor bajo próximo a cero, el proceso estará limitado por entrada/salida

En los casos más usuales en los que existe más de un proceso en el sistema, no podemos hacer las consideraciones anteriores puesto que puede desvirtuarse el verdadero comportamiento del sistema. Por este motivo se establecen las mismas medidas, pero con valores medidos obtenidos al considerar el conjunto de procesos presentes. Estas serán las medidas que nos reflejarán el verdadero comportamiento del sistema. Las medidas a las que nos referimos son:

- **Tiempo medio de servicio.**
- **Tiempo medio de espera. .**
- **Eficiencia** (índice medio de servicio).

Además de las anteriores, en la planificación del procesador suelen emplearse otras medidas de interés:

- **Tiempo del núcleo.** Es el tiempo consumido por el núcleo del sistema operativo para tomar las decisiones de planificación del procesador, donde se incluyen los tiempos de cambio de contexto y de proceso.
- **Tiempo de inactividad** (Idle). Es el tiempo consumido cuando la cola de procesos preparados está vacía y por tanto no puede realizarse ningún trabajo productivo.

## Algoritmos de Planificación

Como ya hemos visto, el planificador del procesador tiene como misión la asignación del mismo a los procesos que están en la cola de procesos preparados.

Esta cola es alimentada desde dos puntos distintos:

- ✓ Cada vez que un usuario inicie la ejecución de un programa, el planificador a largo plazo recibe la orden de ejecución, crea el proceso y lo pasa al planificador a corto plazo, colocándose en la cola de procesos preparados.
- ✓ Cuando un proceso deja de estar en estado de ejecución y no existen causas para su bloqueo, o deja de estar bloqueado, pasa nuevamente a la cola de procesos preparados.

Por otro lado, cuando un proceso termina su ejecución, deja de existir para el planificador.

Las políticas de planificación se agrupan en:

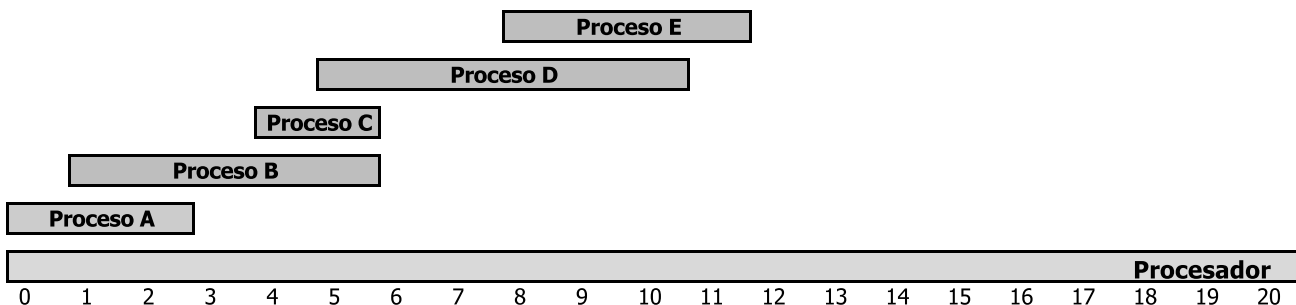
- **Políticas apropiativas.** Son las que producen un cambio de proceso con cada cambio de contexto; es decir, el proceso que está haciendo uso del procesador puede ser temporalmente suspendido y permitir que otro proceso se apropie del procesador. Se utilizan en sistemas operativos con tiempo compartido y tiempo real.
- **Políticas no apropiativas.** Son aquellas en las que un proceso no abandona nunca el procesador desde su comienzo hasta su fin. Se utilizan en sistemas de proceso por lotes.

Una política apropiativa trabaja con procesos no apropiativos mientras que una no apropiativa lo hace con procesos apropiativos.

En los sistemas operativos comerciales existen diversas políticas de planificación, de las cuales veremos algunas a continuación, puestas en práctica a través de algoritmos que en ningún caso son perfectos. Recordemos que los objetivos y criterios pueden ser contradictorios entre sí, de manera que si favorecemos a un tipo de procesos, normalmente perjudicaremos a otros.

Para el estudio de las diferentes políticas nos basaremos en la situación de un grupo de procesos existentes en un sistema, cuyos datos se encuentran en la tabla siguiente y su representación gráfica en la figura.

Nombre Proceso	Instante llegada	Tiempo ejecución	Prioridad
A	0	3	0
B	1	5	1
C	4	2	0
D	5	6	2
E	8	4	1



Además de figurar el instante de entrada en el sistema, también se ha representado el tiempo de servicio de cada proceso, es decir, el tiempo que necesita cada uno de ellos para ejecutarse si fuera el único presente en el sistema. Por otro lado, supondremos que estos procesos no necesitan realizar operaciones de entrada/salida (aunque esto sea una utopía) para facilitar el estudio.

La unidad de tiempo que utilizamos en el ejemplo es abstracta, es decir, puede ser cualquier unidad (milisegundos, segundos, etc.). No debemos olvidar que el valor más representativo del comportamiento de una política de planificación es el índice medio de servicio o eficiencia que es un valor adimensional y suele expresarse en tanto por ciento (%).

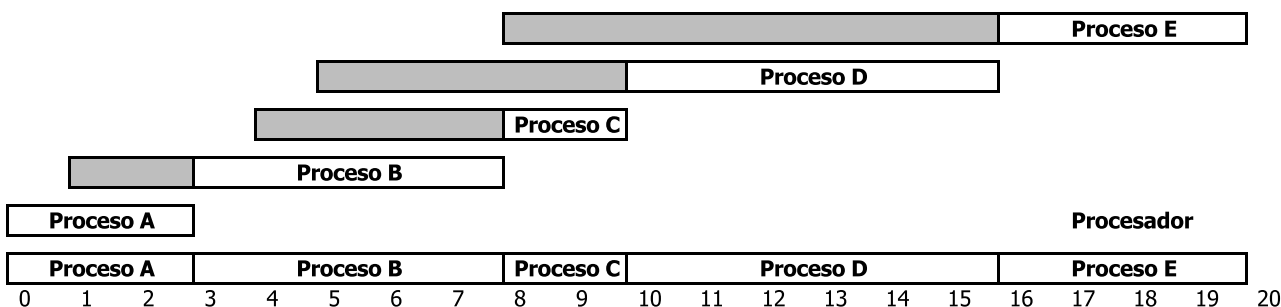
### Primero en llegar, primero en ser servido (FCFS)

En esta política de planificación FCFS (First Come, First Served), el procesador ejecuta cada proceso hasta que termina; por tanto, los procesos que entren en cola de procesos preparados permanecerán encolados en el orden en que lleguen hasta que les toque su ejecución. Este método se conoce también como <primero en entrar, primero en salir> (First Input, First Output – FIFO).



Se trata de una política muy simple y sencilla de llevar a la práctica, pero muy pobre en cuanto a su comportamiento. En la tabla se muestran los datos de los procesos para esta planificación y en la figura puede verse cómo despacha FCFS los procesos del ejemplo propuesto, donde las partes sombreadas indican el tiempo que ha estado cada proceso en espera de acceder al procesador.

Nombre Proceso	Instante llegada	Tiempo ejecución	Instante de Finalización	T	E	I
A	0	3	3	3	0	1.00
B	1	5	8	7	2	0.71
C	4	2	10	6	4	0.33
D	5	6	16	11	5	0.54
E	8	4	20	12	8	0.33
Media				7.8	3.8	0.58



Podemos observar que el índice de servicio mejora cuanto más largos son los procesos. Es decir, los procesos cortos que entren en el sistema después de uno o varios procesos largos tendrán que esperar un período de tiempo relativamente largo hasta su ejecución.

La cantidad de tiempo de espera de cada proceso depende del número de procesos que se encuentren en cola en el momento de su petición de ejecución y del tiempo que cada uno de ellos tenga en uso al procesador, y es independiente de las necesidades de ejecución del propio proceso.

**Las características de esta política son las siguientes:**

- No es apropiativa.
- Es justa, aunque los procesos largos hacen esperar mucho a los cortos.
- Es una política predecible.
- El tiempo medio de servicio es muy variable en función del número de procesos y su duración.

**Actividad N°1:**

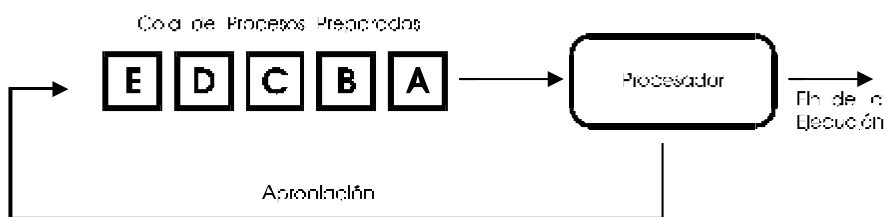
Obtenga los Tiempos de Servicio, Espera e Índices de la siguiente tabla de procesos ejecutando una planificación FCFS. Agregue el diagrama de Gantt de la ejecución de manera similar a lo observado.

Nombre Proceso	Instante llegada	Tiempo ejecución	Prioridad
A	0	6	0
B	2	5	1
C	2	6	2
D	3	7	2
E	6	4	1

**Round-Robin (RR)**

Esta política, cuya traducción podría ser asignación cíclica o planificación en rueda, es una mejora de la FCFS. Trata de ser más justa en cuanto a la respuesta tanto de los procesos cortos como de los largos.

Consiste en conceder a cada proceso en ejecución un determinado período de tiempo  $q$  (quantum), transcurrido el cual, si el proceso no ha terminado, se le devuelve al final de la cola de procesos preparados, concediéndose el procesador al siguiente proceso por su correspondiente quantum.



Esta interrupción periódica continúa hasta que el proceso termine su ejecución, formando una rueda de procesos que serán ejecutados cíclicamente hasta que terminen.

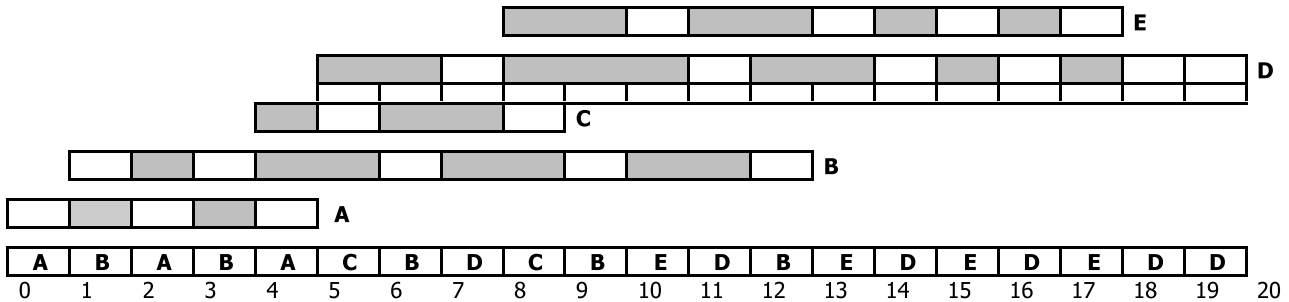
La gestión de la cola de procesos preparados se puede realizar de muy diversas maneras siendo las más comunes la FIFO o por prioridades, donde los procesos se ordenan según su prioridad.

Variando el parámetro  $q$  lograremos tener diferentes comportamientos de esta política de tal forma que si  $q$  es mayor que el tiempo que necesita para su ejecución el proceso más largo, se convertiría en una política FCFS. En cambio, si se aproxima a 0, la sobrecarga de sistema será muy grande puesto que la mayor parte del tiempo se consumiría en cambio de contexto.

Los valores de  $q$  varían entre 10 y 100 milisegundos, siendo recomendable que el 80 por 100 de los tiempos de respuesta de los procesos sean inferiores al quantum. En los sistemas actuales ronda los 200 milisegundos.

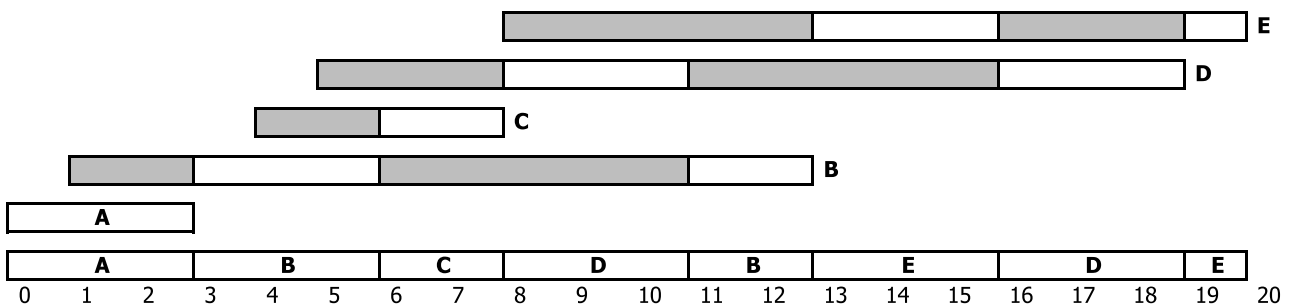
En la figura puede verse esta planificación para el ejemplo propuesto, con un valor de quantum  $q=1$ , con las siguientes condiciones:

- ✓ Si un proceso finaliza durante su quantum, inmediatamente se le concede el procesamiento a otro proceso, al que se le asigna el quantum completo.
- ✓ Al crearse un proceso y pasar a la lista de procesos preparados, se coloca al final de la lista.
- ✓ Si un proceso comienza su ejecución (creación) en el mismo momento en que un quantum finaliza, se supondrá que dicho proceso ha llegado a la cola de procesos preparados antes de la finalización del mencionado quantum.



Nombre Proceso	Instante llegada	Tiempo ejecución	Instante Finalización	T	E	I
A	0	3	5	5	2	0.60
B	1	5	13	12	7	0.42
C	4	2	9	5	3	0.40
D	5	6	20	15	9	0.40
E	8	4	18	10	6	0.40
Media				9.4	5.4	0.44

A continuación se repite la planificación RR para un valor de quantum de 3 y con las mismas condiciones que en el caso anterior. La tabla y la figura representan los datos y el gráfico respectivamente del ejemplo propuesto para este valor.



Nombre Proceso	Instante llegada	Tiempo ejecución	Prioridad	T	E	I
A	0	3	3	3	0	1.00
B	1	5	13	12	7	0.42
C	4	2	8	4	2	0.50
D	5	6	19	14	8	0.43
E	8	4	20	12	8	0.33
Media				9.0	5.0	0.54

En este caso el tiempo de servicio  $T$  se mantiene prácticamente constante. Se puede observar que el tiempo de espera  $E$  crece de acuerdo con el tiempo de ejecución de cada proceso.

**Las características de esta política de planificación son:**

- Baja sobrecarga si el cambio de contexto es eficiente y los procesos siempre están en la memoria principal.
- El tamaño óptimo del quantum depende de:
  - El tipo de sistema.
  - Las cargas que vaya a soportar el sistema.
  - El número de procesos en el sistema y su tipo.
  - Es la política más utilizada para tiempo compartido.
  - Ofrece un índice de servicio uniforme para todos los procesos.
  - Es una política apropiativa.

Este tipo de planificación es la que utilizan los sistemas operativos actuales gregando a la misma condiciones de prioridad.

**Actividad N°2:**

Obtenga los Tiempos de Servicio, Espera e Indices de la siguiente tabla de procesos ejecutando una planificación RR. Agregue el diagrama de Gantt . Utilice un quantum = 2.

Nombre Proceso	Instante llegada	Tiempo ejecución	Prioridad
A	0	6	0
B	2	5	1
C	2	6	2
D	3	7	2
E	6	4	1

**Actividad N3:**

Idem a la actividad anterior pero con quantum = 4.  
 ¿Qué diferencias nota?  
 ¿Mejora en algo el índice de servicio?

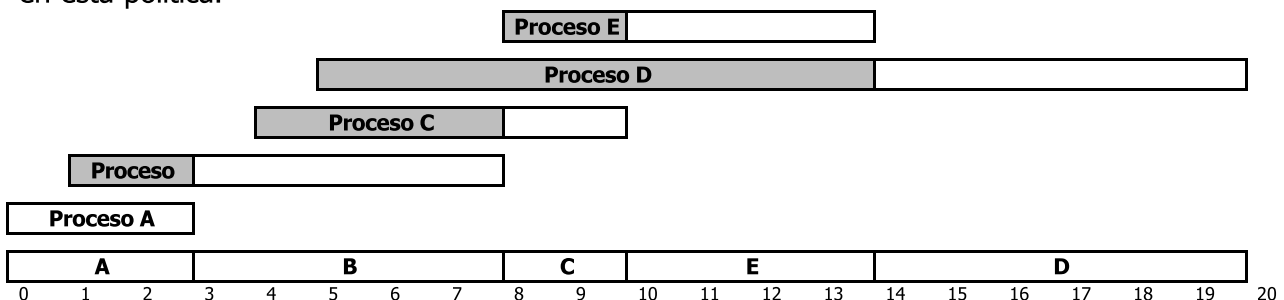
**El siguiente proceso, el más corto (SJN)**

Hemos visto que la política RR mantiene constante el índice de servicio de los procesos cortos basándose en la apropiación del procesador. El método SJN (Shortest Job Next) es una política de planificación no apropiativa que trata de cubrir los mismos objetivos que la RR.

Esta política toma de la cola de procesos preparados el que necesite menos tiempo de ejecución para realizar su trabajo. Para ello debe saber el tiempo de procesador que necesita cada proceso, lo cual es tarea nada fácil, pero posible a través de diversos métodos como pueden ser la información suministrada por el propio usuario, por el propio programa, basándose en la historia anterior (heurística), etc.

El tiempo de servicio  $T$  en esta política es bueno para los procesos cortos, saliendo perjudicados los procesos largos.

En la figura y la tabla pueden verse el gráfico y los datos, respectivamente, del ejemplo propuesto en esta política.



Nombre Proceso	Instante llegada	Tiempo ejecución	Instante Finalización	T	E	I
A	0	3	3	3	0	1.00
B	1	5	8	7	2	0.71
C	4	2	10	6	4	0.33
D	5	6	20	15	9	0.40
E	8	4	14	6	2	0.67
Media				7.8	3.8	0.62

**Las características de esta política de planificación son las siguientes:**

- No es apropiativa.
- El tiempo de espera aumenta de acuerdo con la longitud de los procesos, pero el tiempo medio de espera con respecto a otras políticas es óptimo.
- Es poco predecible.
- No es justa con los procesos largos.
- Buen tiempo de servicio.
- Resulta difícil de poner en práctica por los datos que necesita para realizarse la planificación.

**Actividad N°4:**

Obtenga los Tiempos de Servicio, Espera e Índices de la siguiente tabla de procesos ejecutando una planificación SJN. Agregue el diagrama de Gantt de la ejecución.

Nombre Proceso	Instante llegada	Tiempo ejecución	Prioridad
A	0	6	0
B	2	5	1
C	2	6	2
D	3	7	2
E	6	4	1

**Actividad N°5:**

Realice el mismo ejercicio pero considerando que todos los procesos llegan al mismo tiempo. Es decir que todos tengan instante de llegada 0 en la tabla.

**Próximo proceso, el de tiempo restante más corto (SRT)**

La política SRT (Shortest Remaining Time) es una mezcla de los dos métodos anteriores y trata de obtener las ventajas de ambos. Para ello esta técnica cambia el proceso que está en ejecución cuando se ejecuta un proceso (paso del planificador de largo plazo al de corto plazo), con una exigencia de tiempo de ejecución total menor que el que se está ejecutando en el procesador. El valor del tiempo de respuesta medio de los procesos largos mejora con respecto a SJN.

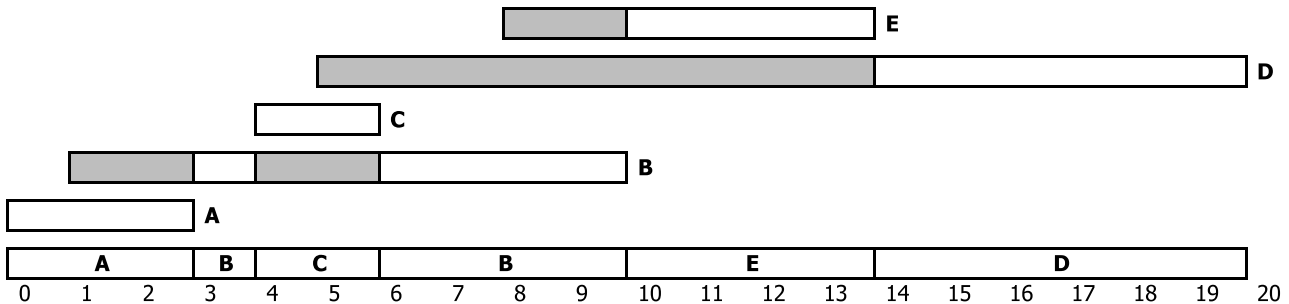
Presenta un excelente índice de servicio I y el tiempo de espera E es bastante corto para la mayoría de los procesos. SRT consigue una buena eficiencia, ya que logra que la lista de procesos preparados sea lo más corta posible.

En la tabla y la figura pueden verse los datos y el gráfico, respectivamente, del ejercicio propuesto para esta política.

Nombre Proceso	Instante llegada	Tiempo ejecución	Instante Finalización	T	E	I
A	0	3	3	3	0	1.00
B	1	5	10	9	4	0.55
C	4	2	6	2	0	1.00
D	5	6	20	15	9	0.40
E	8	4	14	6	2	0.67
Media				7.0	3.0	0.72

**Esta política presenta las siguientes características:**

- Es una variante de SJN, para hacerla apropiativa.
- Puede ser injusta, ya que un proceso corto puede echar a uno largo que esté haciendo uso del procesador y que además esté terminando.
- Presenta una mayor sobrecarga.
- Excelente tiempo medio de servicio.
- Es muy eficiente.



**Actividad N°6:**

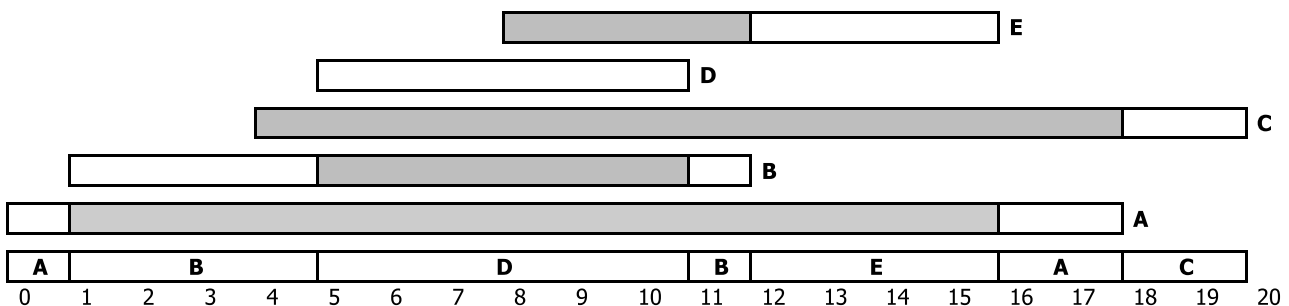
Obtenga los Tiempos de Servicio, Espera e Índices de la siguiente tabla de procesos ejecutando una planificación SRT. Agregue el diagrama de Gantt .

Nombre Proceso	Instante llegada	Tiempo ejecución	Prioridad
A	0	6	0
B	2	5	1
C	2	6	2
D	3	7	2
E	6	4	1

**Prioridad**

En esta política se asocia a cada proceso una prioridad, de manera que el procesador se asigna al proceso de mayor prioridad.

Las prioridades pueden ser definidas interna o externamente. En el primer caso, el sistema operativo se basa en una serie de informaciones medibles para el cálculo y asignación de dichas prioridades (tiempo necesitado de procesador, necesidad de memoria, etc.).



El principal problema de esta política es el bloqueo o postergación indefinida, ya que un proceso de baja prioridad puede estar esperando su turno indefinidamente. Para evitarlo se suele emplear lo que se denomina envejecimiento de las prioridades, que aumenta gradualmente las prioridades de los procesos que están a la espera de utilizar el procesador.

Cualquier algoritmo basado en esta política puede ser apropiativo o no apropiativo. En el primer caso, un proceso puede ser retirado del procesador si aparece otro de mayor prioridad en la cola de procesos preparados.

El comportamiento de esta política como apropiativa puede verse gráficamente, para el ejemplo propuesto, en la figura anterior, y los datos en la tabla siguiente.

Nombre Proceso	Instante llegada	Tiempo ejecución	Instante Finalización	T	E	I
A	0	3	18	18	15	0.17
B	1	5	12	11	6	0.45
C	4	2	20	16	14	0.13
D	5	6	11	6	0	1.00
E	8	4	16	8	4	0.50
Media				11.8	7.8	0.45

### Próximo el de más alto índice de respuesta (HRN)

HRN (High Response Next) es una política que trata de corregir las posibles injusticias de la política SJN con los procesos largos y las de la política FCFS con los procesos cortos

Se basa en hacer variable la prioridad de un proceso, calculándola constantemente por medio de la expresión:

$$P = (w+t)/t$$

**P** es la prioridad del proceso.

**w** es el tiempo de espera en la cola de procesos preparados.

**t** es el tiempo de ejecución del proceso.

En un principio P valdrá 1 e irá aumentando paulatinamente a medida que el proceso permanezca en la cola de procesos preparados (w favorece a los procesos largos), e irá disminuyendo cuanto más tiempo esté en ejecución (t favorece a los procesos cortos).

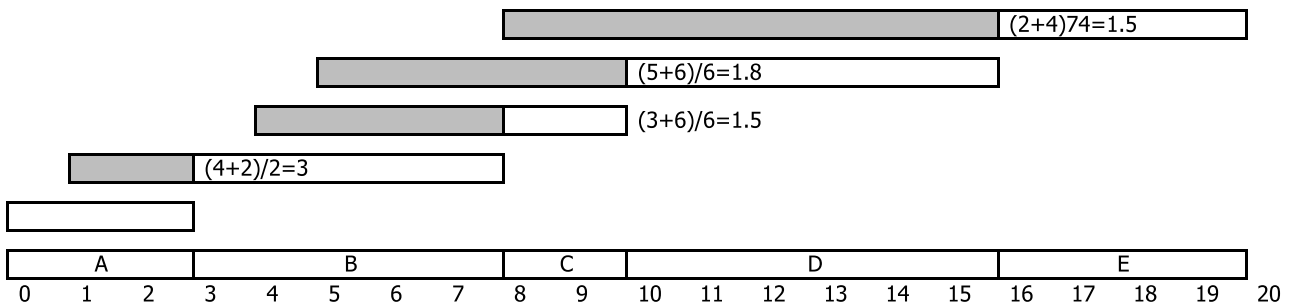
Cuando un proceso que está en ejecución abandona el procesador, ya sea por que ha terminado o por una operación de entrada/salida, el proceso preparado que tenga mayor prioridad (que más haya esperado en la cola), será el que se seleccione para su ejecución.

### Esta política presenta los siguientes inconvenientes:

- Si un usuario ejecuta un proceso corto inmediatamente después de que un proceso largo haya comenzado a utilizar el procesador, deberá sufrir una larga espera.
- Es muy costosa de llevar a la práctica, ya que la prioridad debe calcularse para todos los procesos en espera, cada vez que termine el proceso que está haciendo uso del procesador.
- Sobrecarga mucho el sistema debido a los cálculos que debe realizar.

El comportamiento de esta política y sus datos pueden verse en la tabla y la figura para el ejemplo propuesto.

Nombre Proceso	Instante llegada	Tiempo ejecución	Instante Finalización	T	E	I
A	0	3	3	3	0	1.00
B	1	5	8	7	2	0.71
C	4	2	10	6	4	0.33
D	5	6	16	11	5	0.54
E	8	4	20	12	8	0.33
Media				7.8	3.8	0.58



**Las características de esta política son:**

- No es apropiativa.
- Es justa.
- Es costosa de poner en práctica.
- Produce una gran sobrecarga en el sistema.

**Actividad N°7:**

Obtenga los Tiempos de Servicio, Espera e Índices de la siguiente tabla de procesos ejecutando una planificación HRN. Agregue el diagrama de Gantt de la ejecución de manera similar a lo observado.

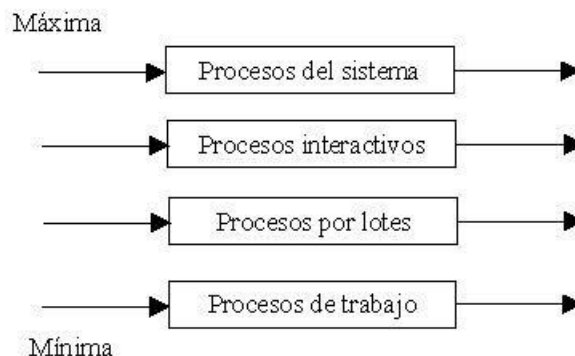
Nombre Proceso	Instante llegada	Tiempo ejecución	Prioridad
A	0	6	0
B	2	5	1
C	2	6	2
D	3	7	2
E	6	4	1

**Colas múltiples**

Cuando los procesos que van a ser ejecutados en una computadora se pueden agrupar en distintos grupos, podemos asignarlos a diferentes colas, cada una con distinta planificación, para darle a cada una de ellas la que realmente necesite.

Esta política divide la cola de procesos preparados en varias colas separadas, de manera que los procesos se asignan a una determinada cola según sus necesidades y tipo.

Para determinar en cada caso qué cola es la que suministrará un proceso para que acceda al procesador cuando éste deje a otro anterior, será controlada por un algoritmo de planificación entre las colas, que normalmente es apropiativo de prioridad fija.



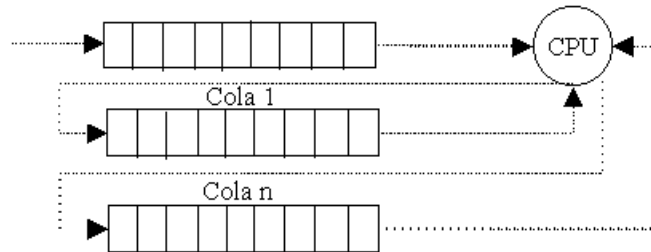
**Colas múltiples con realimentación (FB)**

Para dar un trato justo a los procesos, es necesario conocer previamente todos sus parámetros característicos: la longitud, si están limitados por entrada/salida o por procesador, la memoria que van a necesitar, etc. Como estos datos no suelen ser conocidos, es difícil determinar el trato que debe recibir cada proceso.

Después de analizar las políticas anteriores, es fácil concluir que se deben adoptar las siguientes cuestiones:

- Favorecer los procesos cortos.
- Favorecer los procesos limitados por entrada/salida.
- Determinar la naturaleza del trabajo a realizar.

El método de colas múltiples con realimentación divide los procesos en varias colas de procesos preparados: cola 0, cola 1, cola 2, y así sucesivamente, de manera que las de numeración más baja tendrán asignadas mayor prioridad.



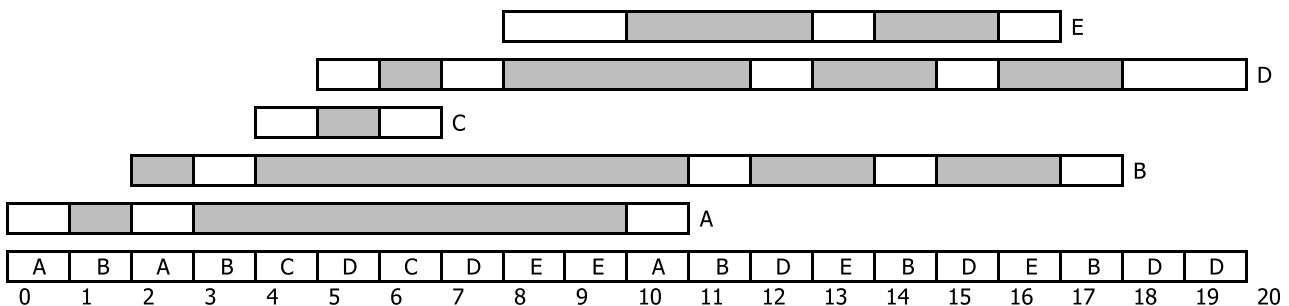
Cuando el proceso que está haciendo uso del procesador finaliza su quantum, se selecciona un nuevo proceso del principio de la cola del nivel más bajo que tenga algún proceso.

Una vez que un proceso haya consumido el quantum de su cola un determinado número de veces, sin haber finalizado su ejecución, será colocado al final de la de nivel inmediatamente superior al anterior.

La política FB (Feedback Multiple Queues) intenta dar un trato justo a los procesos por medio de separación de los mismos en categorías, para así darles el servicio que necesitan. Los procesos limitados por procesador irán a las colas de menor prioridad (nivel más alto) mientras los de mayor prioridad serán aquellos procesos muy interactivos.

En la tabla y en la figura pueden verse los datos y la gráfica, respectivamente de esta política, permitiendo un quantum a cada proceso de la cola antes de pasar a la siguiente.

Nombre Proceso	Instante llegada	Tiempo ejecución	Instante Finalización	T	E	I
A	0	3	11	11	8	0.27
B	1	5	18	17	12	0.29
C	4	2	7	3	1	0.67
D	5	6	20	15	9	0.67
E	8	4	17	9	5	0.80
Media				11.0	7.0	0.54



Recuerde las prioridades de la primera tabla para seguir el diagrama de Gantt que se presenta.

### Las características de esta política son:

- Soporta bien la sobrecarga.

- Es apropiativa.
- Es muy adaptable a las necesidades del sistema, ya que cada cola puede ser gestionado de forma diferente.

### Actividad N°8:

Obtenga los Tiempos de Servicio, Espera e Índices de la siguiente tabla de procesos ejecutando una planificación FB. Agregue el diagrama de Gantt.

Nombre Proceso	Instante llegada	Tiempo ejecución	Prioridad
A	0	6	0
B	2	5	1
C	2	6	2
D	3	7	2
E	6	4	1

## Evaluación de los Algoritmos de Planificación

Como hemos visto existen diversos algoritmos de planificación, cada uno con sus propios parámetros, por lo que ahora nos preguntaremos Cómo seleccionar un algoritmo de planificación para un sistema determinado.

Para seleccionar un algoritmo de planificación, primero debemos definir los criterios que se aplicarán para elegir el algoritmo, que podrán ser: Utilización del procesador, tiempo de respuesta, productividad; para luego determinar la importancia relativa de cada una de estas mediciones, como:

- ✓ Maximizar la utilización del procesador, con la restricción de cuantificar el tiempo máximo de respuesta.
- ✓ Maximizar la productividad.

Una vez definidos los criterios de selección, se querrá evaluar los distintos algoritmos, para lo cual hay varios métodos, como pueden ser, modelado determinista, modelos de colas, simulaciones e implementación. Comentaremos brevemente cada uno de ellos.

### Modelado determinista

Este método toma una carga de trabajo determinada y define el rendimiento de cada algoritmo para esa carga de trabajo. Este modelo es sencillo y proporciona números exactos, permitiendo comparar los algoritmos para un sistema determinado, sin embargo requiere números precisos en la entrada y los resultados solo son aplicables a esos casos.

Este modelo es adecuado para situaciones en que se está ejecutando una y otra vez el mismo conjunto de programas, y de esta forma poder seleccionar el algoritmo de planificación más adecuado.

### Modelos de colas

En la mayoría de los sistemas, los procesos que se ejecutan son muy heterogéneos, por lo que no existe un conjunto estático de procesos que puedan usarse para el modelo determinista. Lo que sí puede determinarse es la distribución de las ráfagas de CPU y de E/S. De esta forma se pueden establecer sus distribuciones y luego simplemente estimarse las correspondientes ráfagas. Se tendrá como resultado la probabilidad de una ráfaga de CPU y de E/S. También deberá determinarse la distribución de los tiempos de llegada de cada proceso. Con estas dos distribuciones (las ráfagas de CPU y tiempos de llegada) se puede calcular la productividad, la utilización del procesador, tiempo de espera, etc., para los diferentes algoritmos de planificación.

El problema que presenta estos modelos es la dificultad de establecer buenas distribuciones y aplicarlas a los diferentes algoritmos de planificación, por lo que frecuentemente se definen las distribuciones de tiempos de llegada y ráfagas de CPU de una manera poco realista, con lo cual el resultado será una mera aproximación del mundo real. Con lo que los resultados obtenidos con estos modelos podrán ser cuestionados.

## **Simulaciones**

Para conseguir una evaluación más precisa de los algoritmos de planificación podemos usar simulaciones, que implica programar un modelo del sistema de computación. El simulador tiene una variable que representa al reloj, tal que al aumentar el valor de esta variable, el simulador modifica el estado del sistema para reflejar las actividades de los dispositivos de E/S, los procesos y el planificador. Al ejecutar la simulación se guardan los datos que indican el rendimiento del algoritmo.

La forma de generarse los datos para la simulación suele ser mediante un generador de números aleatorios, programado para generar procesos, tiempos de ráfaga de CPU, llegadas de procesos, etc., según unas distribuciones probabilísticas, como pueden ser uniforme, exponencial, Poisson.

El problema de las simulaciones es que pueden ser muy costosas. Una simulación muy detallada ofrece resultados muy precisos, pero también requiere mayor tiempo de la computadora. Además, el diseño, codificación y depuración del simulador puede ser una tarea de gran magnitud.

## **Implementación**

Incluso una simulación tiene una precisión limitada. La única manera completamente exacta de evaluar un algoritmo de planificación es codificarlo y colocarlo en el sistema operativo para ver cómo funciona.

### **Actividad N°9:**

Según lo visto en este capítulo:

¿Cuál cree que es el mejor algoritmo de planificación?

Proponga un algoritmo propio y realice un ejemplo de manera similar a las actividades propuestas.

### **Actividad N°10:**

¿Podría modelizar mediante un programa con cualquier lenguaje los algoritmos vistos?

