

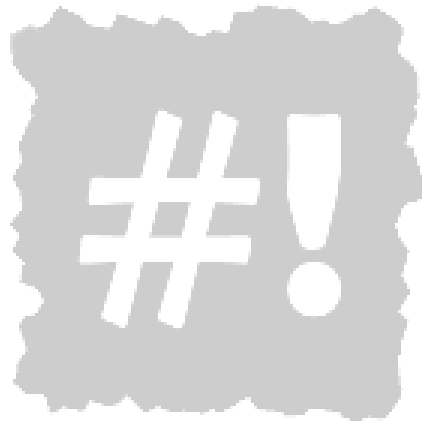


Preparador Informática

www.preparadorinformatica.com

<p>MANUAL 1</p> <p>PLANIFICACIÓN DE PROCESOS</p>
--

1. Procesos	2
1.1.Estados de un proceso.....	3
2. Planificación de procesos	4
2.1.Algoritmo FCFS	5
2.2.Algoritmo ROUND ROBIN.....	7
2.3.Algoritmo SJF ó SPN	9
2.4.Algoritmo SRT	11
2.5.Algoritmo POR PRIORIDADES	12



Preparador Informática

1. Procesos

Un proceso es un programa o un fragmento de programa en ejecución. Cada proceso necesita una serie de recursos, siendo el sistema operativo el encargado de proporcionárselos. En función del tipo de proceso, los recursos necesarios pueden ser diferentes, pero, por regla general, todos necesitan un espacio en memoria y un tiempo de uso del microprocesador (CPU).

En el sistema operativo, cada proceso se representa por medio de su propio bloque de control de proceso (PCB, Process Control Block). Un **PCB** es un registro de datos que contiene información básica del proceso como:

- Su estado actual.
- Su PID (número identificador de proceso).
- Valores de registros asociados a él, como el contador de programa, los punteros, acumuladores, etc.
- Valores de los recursos asignados: espacio en memoria, archivos, E/S, etc.

Los procesos, desde su creación, tienen la capacidad de comunicarse y sincronizarse con otros procesos y con recursos del sistema. Este hecho da lugar a diferentes tipos de procesos:

- **Independientes:** no se comunican con otros procesos. Estos tipos de procesos apenas existen.
- **Cooperativos:** se comunican y sincronizan para realizar una actividad común.
- **Competitivos:** necesitan hacer uso del mismo recurso y, por consiguiente, competirán por él.

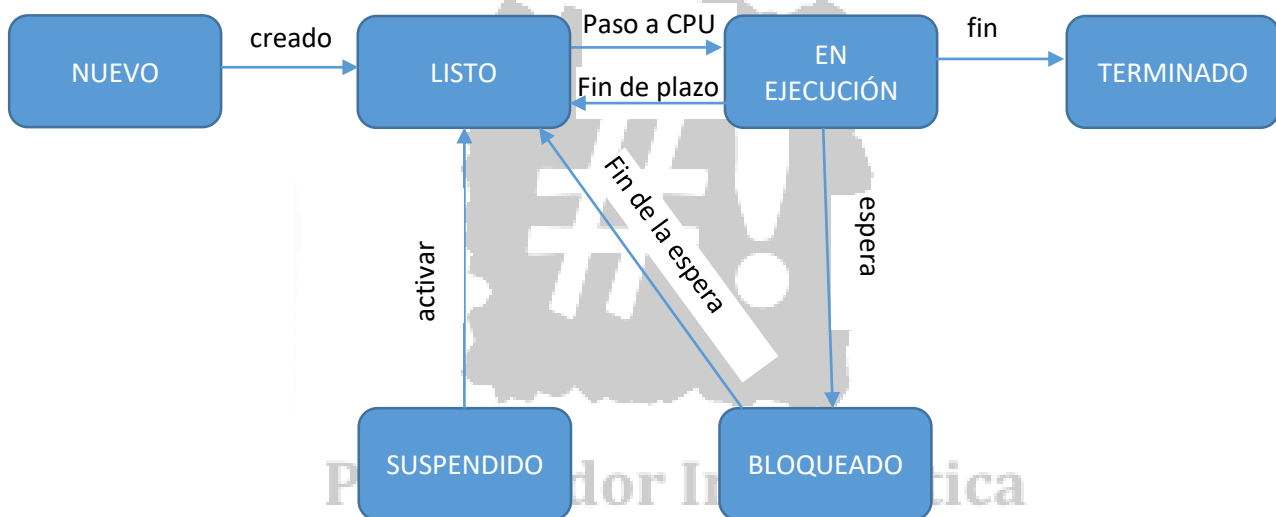
En cualquier caso, la ejecución de los procesos exigirá concurrencia, cualidad que deberá gestionar el sistema operativo gracias a la técnica denominada multiprogramación, que permite que dos o más procesos puedan ejecutarse "a la vez".

Preparador Informática

1.1. Estados de un proceso

Los diferentes estados en el ciclo de vida de un proceso son:

- **Nuevo.** Proceso nuevo creado.
- **Listo.** Proceso que está esperando la CPU para ejecutarse.
- **En ejecución.** Proceso que actualmente está ejecutándose en la CPU.
- **Bloqueado.** Proceso que está a la espera de que finalice una E/S.
- **Suspendido.** Proceso que se ha llevado a la memoria virtual para liberar memoria principal.
- **Terminado.** Proceso que ha finalizado.



2. Planificación de procesos

Los procesos se ejecutan según un orden establecido por el sistema operativo. La forma en la que el sistema operativo gestiona los procesos es lo que se conoce como **planificación** y la herramienta que lo hace recibe el nombre de **planificador** (scheduler). Para realizar esta tarea se utilizan **los algoritmos de planificación**, que se pueden clasificar en dos tipos:

- **Apropiativos:** los procesos pueden ser expulsados de la CPU.
 - Mayor coste de implementación
 - Mejor optimización del uso de la CPU
 - Utilizados en sistemas multiusuario y multitarea
- **No apropiativos:** los procesos se ejecutan hasta que terminan o se bloquean.
 - Sencillos
 - Rendimiento menor
 - Utilizados en sistemas batch o monousuario.

Debemos tener en cuenta una serie de conceptos y objetivos de los algoritmos de planificación, de manera sintetizada podemos expresarlo de la siguiente manera:

- **Utilización de CPU:**
 - Porcentaje de tiempo que se usa la CPU.
 - Objetivo: Maximizar.
- **Productividad:**
 - Número de trabajos terminados por unidad de tiempo.
 - Objetivo: Maximizar.
- **Tiempo de retorno (Tq):**
 - Tiempo que está un proceso en el sistema.
 - Instante final (Tf) menos instante inicial (Ti).
 - Objetivo: Minimizar.
- **Tiempo de servicio (Ts):**
 - Tiempo dedicado a tareas productivas (cpu, entrada/salida).
 - $T_s = T(\text{CPU}) + T(\text{E/S})$
- **Tiempo de espera (Te):**
 - Tiempo que un proceso pasa en colas de espera.
 - $T_e = T_q - T_s$
- **Tiempo de retorno normalizado (Tn):**
 - Razón entre tiempo de retorno y tiempo de servicio.
 - $T_n = T_q / T_s$
 - Indica el retardo experimentado.

Los principales algoritmos de planificación son:

2.1. Algoritmo FCFS

El algoritmo FCFS (First-Come, First-Served) es un algoritmo no apropiativo, que consiste en atender a los procesos por estricto orden de llegada a la cola de procesos listos. Cada proceso se ejecuta hasta que termina, o hasta que hace una llamada de E/S.

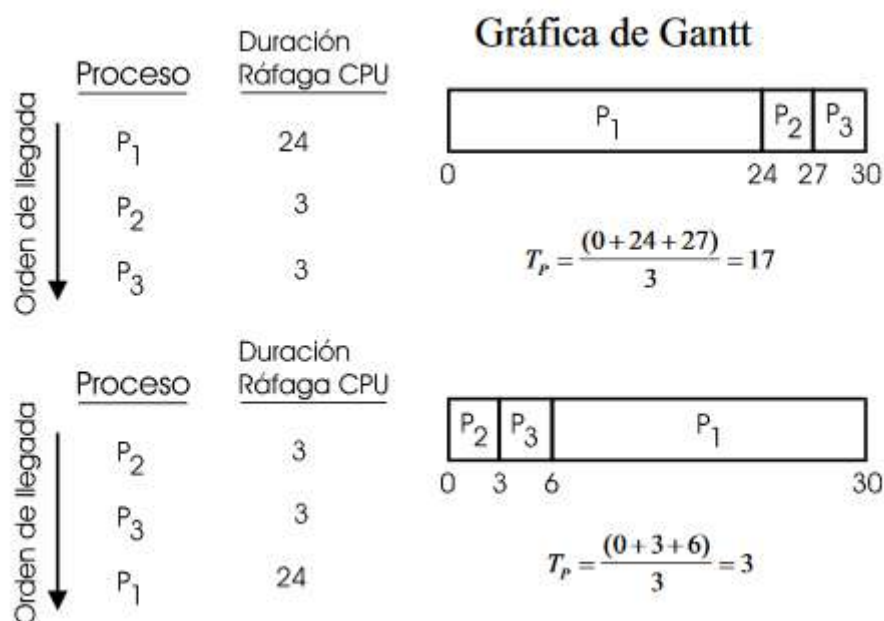
Es un algoritmo que penaliza a los procesos cortos y en este algoritmo el tiempo promedio de espera suele ser bastante largo.

Ejemplo 1:

Procesos	Tiempo de llegada	Tiempo de ejecución
A	0	5
B	1	3
C	2	1

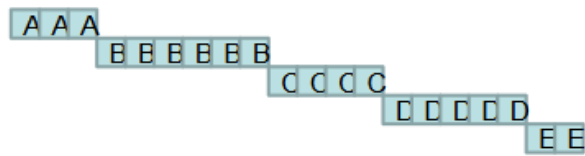
	0	1	2	3	4	5	6	7	8
A	X	X	X	X	X				
B						X	X	X	
C									X

Ejemplo 2 y 3: (Con otra forma de representarlo y calculando el tiempo promedio de espera)



Ejemplo 4: (Con otra forma de representarlo y cálculo de parámetros)

Proceso	Llegada	Servicio
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



Proceso	Llegada	Servicio	Inicio	Fin	Retorno	Espera	Retorno normalizado
A	0	3	0	3	3	0	$3/3=1$
B	2	6	3	9	7	1	$7/6=1.16$
C	4	4	9	13	9	5	$9/4=1.25$
D	6	5	13	18	12	7	$12/5=2.4$
E	8	2	18	20	12	10	$12/2=6$

- Tiempo medio de espera = 4,6.
- Tiempo medio de retorno normalizado = 2,4.

Preparador Informática

2.2. Algoritmo ROUND ROBIN

El algoritmo Round Robin es un algoritmo apropiativo que consiste en darle a cada proceso un intervalo de tiempo de ejecución (quantum). Cada vez que se agota el quantum el proceso que está en ejecución deja de ejecutarse y se sitúa al final de la cola de procesos listos. La cola de procesos listo se gestiona con una política FIFO.

Ejemplo 1: quantum = 2

Procesos	Tiempo de llegada	Tiempo de ejecución
A	0	5
B	1	3
C	2	1

	0	1	2	3	4	5	6	7	8
A	X	X				X	X		X
B			X	X				X	
C					X				

Ejemplo 2: (Con otra forma de representarlo y calculando el tiempo promedio de espera)

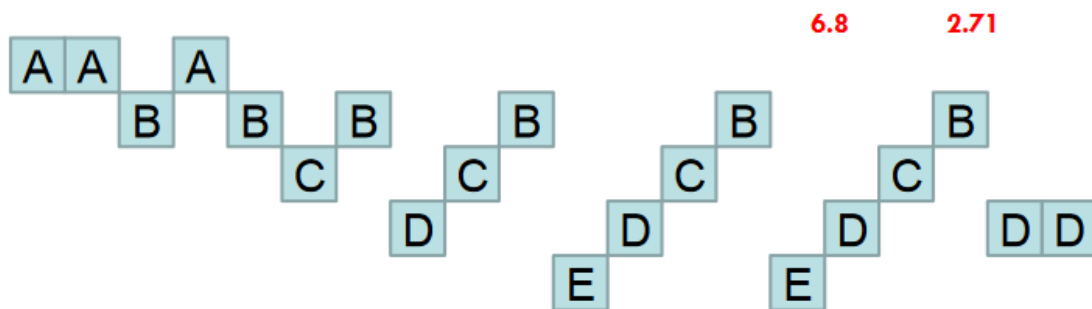
Proceso	Duración Ráfaga CPU	Cuanto de tiempo $Q = 4$	$T_p = \frac{(0+4+7+(10-4))}{3} = 5,66$
P ₁	24		
P ₂	3		
P ₃	3		

P ₁	P ₂	P ₃	P ₁					
0	4	7	10	14	18	22	26	30

Ejemplo 3: (Con otra forma de representarlo y cálculo de parámetros)

quantum = 1

Proceso	Llegada	Servicio	Inicio	Fin	Retorno	Espera	Retorno normalizado
A	0	3	0	4	4	1	$4/3=1.33$
B	2	6	2	18	16	10	$16/6=2.66$
C	4	4	5	17	13	9	$13/4=3.25$
D	6	5	7	20	14	9	$14/5=2.8$
E	8	2	10	15	7	5	$7/2=3.5$



6.8

2.71

Preparador Informática

2.3. Algoritmo SJF ó SPN

El algoritmo SJF (Shortest Job First) o también conocido como SPN (Shortest Process Next) es un algoritmo no apropiativo que consiste en ejecutar el proceso más corto de los que hay en la cola de procesos listos. Si hay dos con igual longitud se usa FCFS.

En este algoritmo existe la posibilidad de **inanición**: Si continuamente llegan trabajos cortos, los trabajos largos nunca llegan a ejecutarse.

Ejemplo 1:

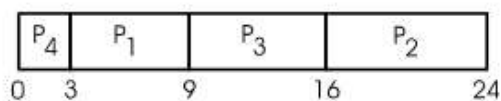
Procesos	Tiempo de llegada	Tiempo de ejecución
A	0	5
B	1	3
C	2	1

	0	1	2	3	4	5	6	7	8
A	X	X	X	X	X				
B							X	X	X
C						X			

Ejemplo 2: (Con otra forma de representarlo y calculando el tiempo promedio de espera)

(Aquí llegan todos los procesos a la vez).

Proceso	Duración Ráfaga CPU
P ₁	6
P ₂	8
P ₃	7
P ₄	3



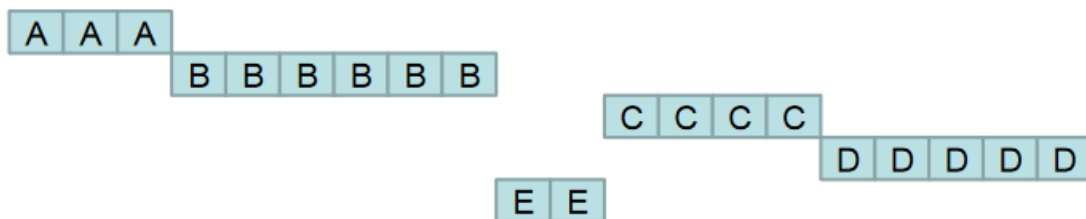
$$T_p = \frac{(0 + 3 + 9 + 16)}{4} = 7$$

Ejemplo 3: (Con otra forma de representarlo y cálculo de parámetros)

Proceso	Llegada	Servicio	Inicio	Fin	Retorno	Espera	Retorno normalizado
A	0	3	0	3	3	0	$3/3=1$
B	2	6	3	9	7	1	$7/6=1.16$
C	4	4	11	15	11	7	$11/4=2.75$
D	6	5	15	20	14	9	$14/5=2.8$
E	8	2	9	11	3	1	$3/2=1.5$

3.6

1.84



Preparador Informática

2.4. Algoritmo SRT

El algoritmo SRT (Shortest Remaining Time) es un algoritmo apropiativo que consiste en ejecutar el proceso que menos tiempo de CPU le quedé para acabar. De este modo, si se está ejecutando un proceso y llega otro más corto, el sistema operativo le quitará la ejecución de la CPU para asignársela al proceso más corto.

Ejemplo:

Procesos	Tiempo de llegada	Tiempo de ejecución
A	0	5
B	1	3
C	2	1

	0	1	2	3	4	5	6	7	8
A	X					X	X	X	X
B		X		X	X				
C			X						

Preparador Informática

2.5. Algoritmo POR PRIORIDADES

El algoritmo por prioridades consiste en ejecutar primero el proceso que tiene más prioridad. Cada proceso tiene asignada una prioridad. El planificador selecciona el proceso con prioridad más alta (a igual prioridad se selecciona con FCFS). Las prioridades pueden ser dinámicas (cambian con el tiempo) o estáticas (se mantienen).

Ejemplo 1: Supongamos que todos los procesos llegan en el mismo instante de tiempo. Empezaríamos por el que más prioridad tiene:

Procesos	Tiempo de ejecución	Prioridad
A	3	3
B	1	1
C	2	3
D	1	4
E	4	2

	0	1	2	3	4	5	6	7	8	9	10
A						X	X	X			
B	X										
C									X	X	
D											X
E		X	X	X	X						

Ejemplo 2: (Con otra forma de representarlo).

Proceso	Duración Ráfaga CPU	Prioridad
P ₁	10	3
P ₂	1	1
P ₃	2	3
P ₄	1	4
P ₅	5	2

P ₂	P ₅	P ₁	P ₃	P ₄
0	1	6	16	18 19

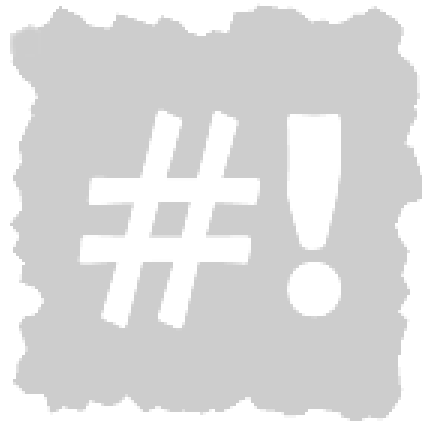
Las prioridades pueden definirse por:

- Factores externos al sistema: Importancia del proceso, del usuario, etc.
- Factores internos al sistema: Requisitos de memoria, límites de tiempo, número de archivos abiertos, etc.

La planificación puede ser **hasta terminación o expulsiva**.

En este algoritmo existe un problema, el denominado bloqueo indefinido o **inanición**: Un proceso con muy baja prioridad puede llegar a no ejecutarse nunca.

La solución para este problema es el envejecimiento: se aumenta la prioridad al aumentar el tiempo de espera en la cola de procesos listos.



Preparador Informática