

Tema 4

Planificación de Procesos

Índice

- Concepto planificación
- Criterios de rendimiento
- Algoritmos de planificación
- Colas multinivel
- Planificador Linux

Concepto planificación

■ Definición:

- La planificación consiste en distribuir el acceso a los recursos del sistema entre los procesos que los solicitan.

■ Objetivo:

- Optimizar el rendimiento de los recursos.
- Priorizar el acceso a los recursos disponibles.

■ Recursos que requieren planificación:

- Dispositivos E/S (discos)
 - Procesador
 - Memoria
- } Planificación de procesos

Planificación de procesos

■ Multiprogramación:

- El SO gestiona múltiples procesos en memoria principal de forma simultánea.
- Los procesos deben compartir el acceso al procesador.

■ Planificación de procesos:

- Decidir acerca de:
 - ¿Qué trabajos se admiten en el sistema?
 - ¿Qué procesos se mantienen en memoria principal?
 - ¿Qué proceso se hace cargo de la CPU cuando ésta queda disponible?
- El planificador de procesos es el responsable de tomar éstas decisiones, repartiendo el uso de la memoria y procesador entre los procesos activos del sistema.

Tipos de planificadores

- **Planificador a largo plazo.**

- Es el encargado de controlar el grado de multiprogramación del sistema (número procesos que se ejecutan al mismo tiempo).
- Admite nuevos trabajos en el sistema, convirtiéndolos en procesos

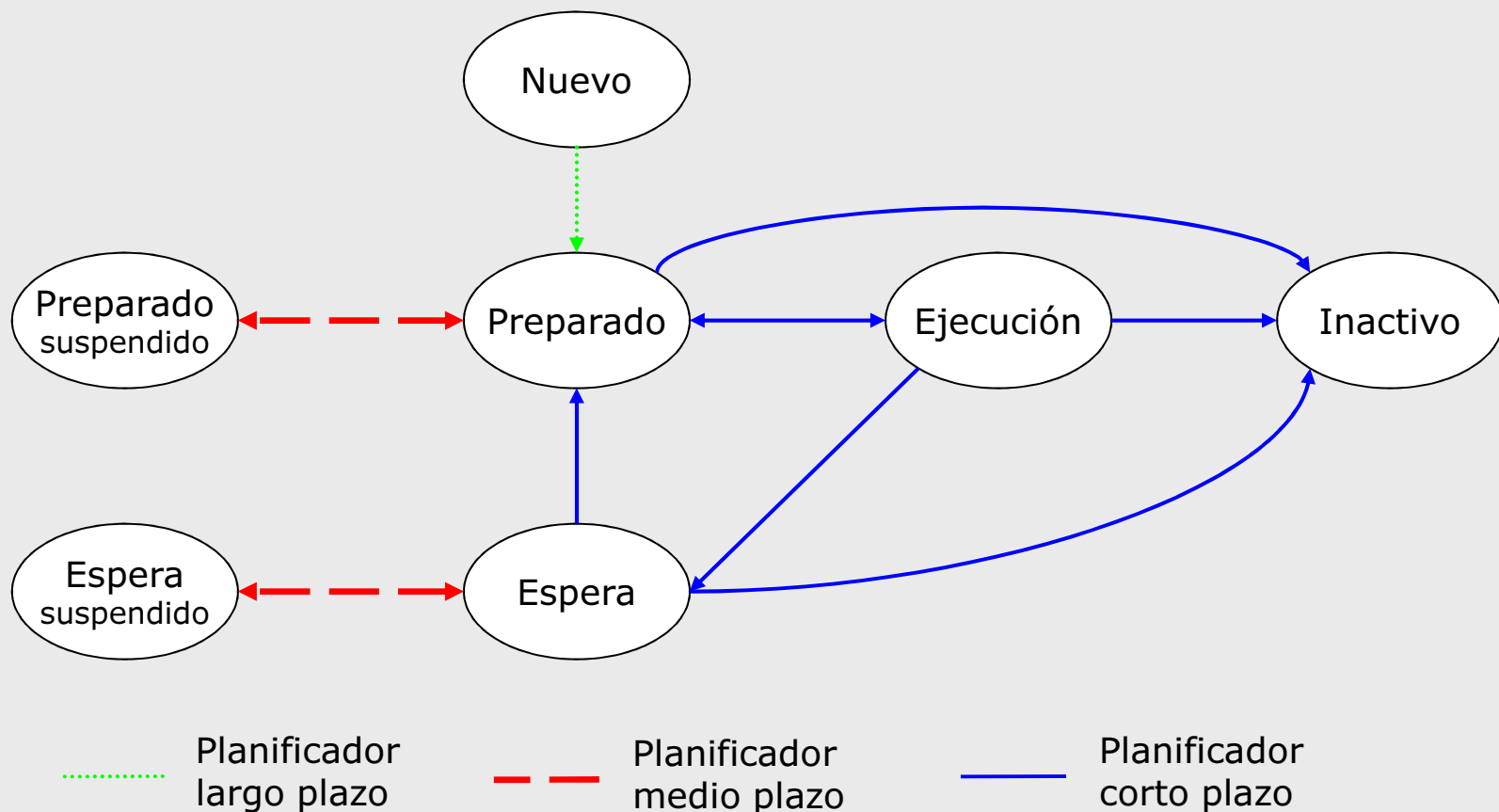
- **Planificador a medio plazo.**

Es el encargado de escoger los procesos que serán descargados totalmente ó parcialmente de **memoria** para ser llevados al disco (suspendidos), con el objetivo de mantener el **rendimiento** del sistema

- **Planificador a corto plazo.**

Asigna la CPU entre los procesos preparados en memoria.

Relación entre planificadores



Colas de planificación procesos

- Durante el ciclo de vida de un proceso y a medida que va cambiando de estado el proceso se ubica dentro de diferentes colas del SO.
 - **Cola de trabajos.**
Incluye todos los procesos del sistema.
 - **Cola de procesos preparados.**
Contiene todos los procesos que están en memoria principal, listos y esperado para ejecutarse.
 - **Colas de dispositivo.**
Conjunto de procesos que están esperando por un dispositivo de E/S.

Módulos del planificador a corto plazo

- **Planificador (algoritmo de planificación)**

Selecciona el proceso para su ejecución atendiendo a un determinado criterio.

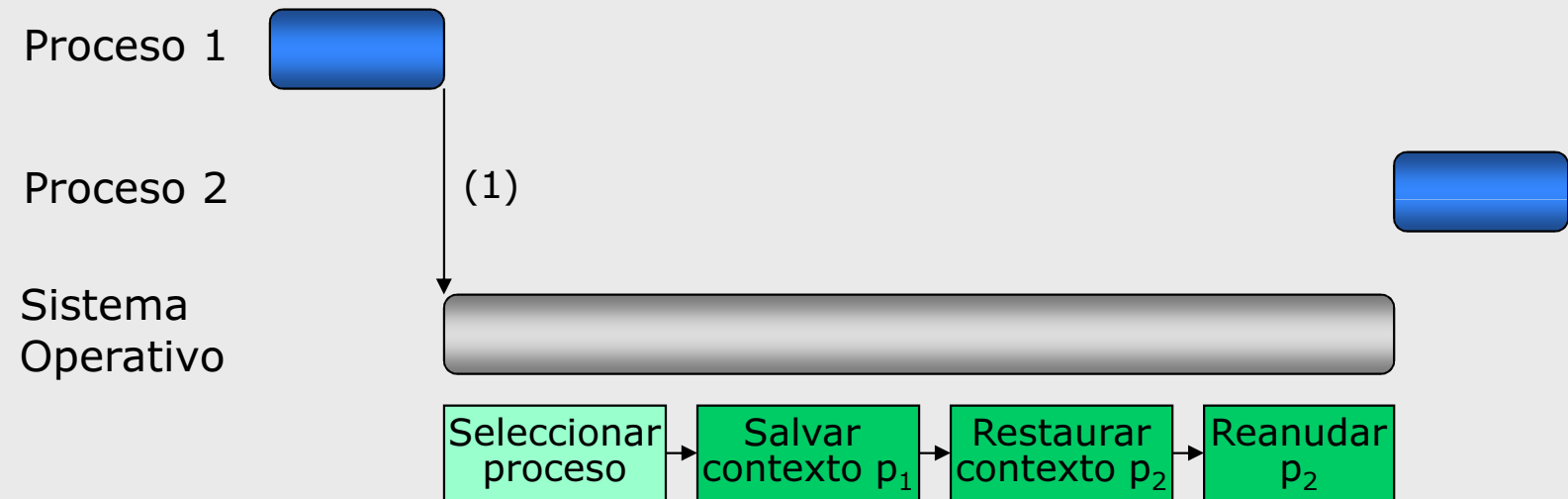
- **Dispatcher (despachador)**

Es el módulo que cede el control de la cpu al proceso seleccionado por el planificador de corto plazo.

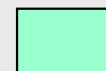
- Salvar contexto de proceso saliente
- Restaurar contexto proceso entrante
- Reanudar la ejecución del proceso
 - Cambiar a modo usuario.
 - Saltar al punto apropiado del programa

} Cambio
contexto

Intercambio procesos



(1) Finalización tiempo de ejecución ó el proceso se bloquea a la espera de un recurso que necesita



Planificador



Despachador

Objetivos Planificación (I)

- Reparto equitativo del procesador de forma que se optimicen uno ó varios de los siguientes criterios:
 - Grado de utilización de la CPU.
 - Productividad (throughput).
Número de procesos completados por unidad de tiempo
 - Tiempo de retorno.
Tiempo transcurrido desde que se lanza un proceso (entra en la cola de preparados) hasta que finaliza su ejecución
 - Tiempo de espera.
Tiempo que el proceso permanece en la cola de preparados
 - Tiempo medio de espera.
Tiempo medio que todos los procesos deben esperar

Objetivos planificación (II)

- **Tiempo de respuesta.**

Tiempo que transcurre desde que se arranca el proceso hasta que empieza a responder (coge por primera vez la CPU)

- **Tiempo de servicio.**

Tiempo esperado para la finalización del proceso (CPU+E/S)

- **Tiempo de retorno normalizado.**

Razón entre tiempo de retorno y tiempo de servicio

Indica el retardo de un proceso en relación a su duración.

Política planificación ideal

- El planificador de CPU ideal es aquel que consigue maximizar el uso de CPU y la productividad y minimizar los tiempos de retorno, respuesta y espera.
- No existe ninguna política de planificación óptima que permita cumplir todos los criterios anteriores al mismo tiempo.
- La política de planificación idónea depende:
 - Tipo de proceso.
 - Criterio que se intenta optimizar.

Algoritmos de planificación apropiativos

- Algunas políticas de planificación pueden funcionar en modo no apropiativo ó modo apropiativo.

- **Modo no apropiativo:**

El proceso que tiene la CPU solo la libera cuando quiere (cuando finaliza o hace E/S)

- No necesita soporte hardware adicional
- Un proceso puede acaparar la CPU
- No son convenientes para entornos de tiempo compartido.

- **Modo apropiativo:**

El planificador puede desalojar un proceso de la CPU antes de tiempo.

- Apropiación inmediata o diferida.
- Mayor coste, pero se evita que un proceso monopolice la CPU

Algoritmos de planificación a corto plazo

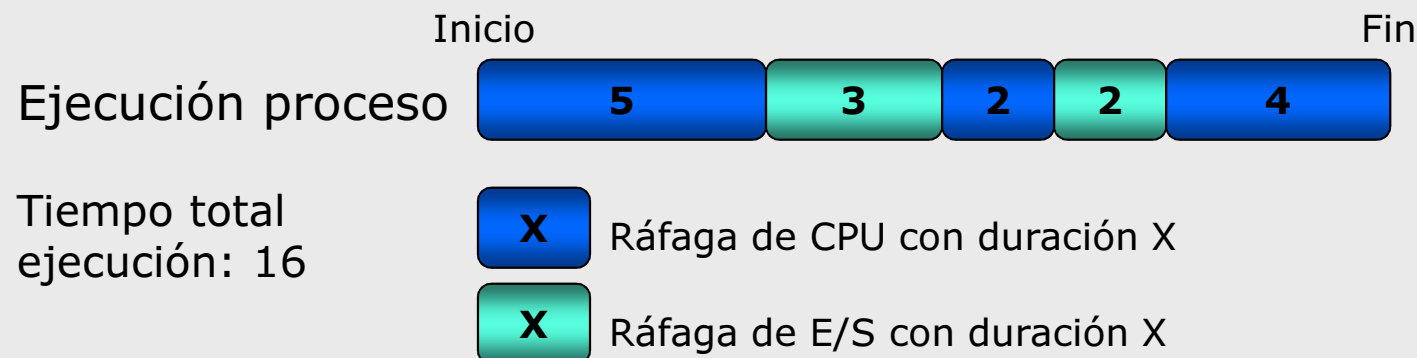
- FCFS ó FIFO
- SJF (Shortest Job First)
- SRT (SJF apropiativo)
- Por prioridades
- Turno rotativo (Round-Robin)
- Colas multinivel

Simplificaciones

- Para los siguientes algoritmos vamos a suponer que:
 - No hay suspensión de procesos
 - Tiempo de intercambio de procesos es nulo
 - Todas las operaciones de entrada / salida de los distintos procesos se pueden solapar entre si

Modelado procesos

- Desde el punto de vista del planificador de CPU los procesos se puede representar como una sucesión de ráfagas de dos tipos:
 - Ráfagas de CPU.
El proceso está ejecutando instrucciones.
 - Ráfagas de E/S.
El proceso utiliza ó espera por E/S.



Tipos de procesos

- Procesos intensivos en CPU

Las ráfagas de CPU son mayores que las de E/S.

Proceso CPU intensivo:



- Procesos intensivos en E/S

Las ráfagas de E/S son mayores que las de CPU.

Proceso E/S intensivo:



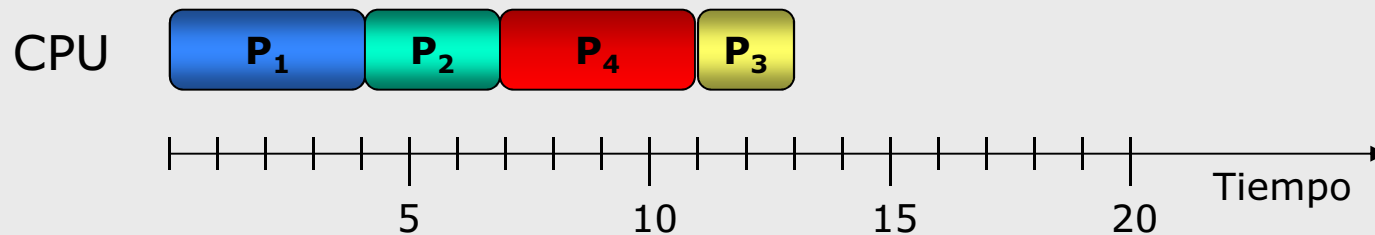
Ejemplos planificación

- En los ejemplos de los algoritmos de planificación vamos a suponer la existencia de 3 procesos con las siguientes características:

Proceso	Tiempo de llegada	Ráfagas del proceso
Proceso A	0	7 _{CPU}
Proceso B	2	4 _{CPU}
Proceso C	3	2 _{CPU}

Diagramas de Gant de Planificación

- Los procesos no tienen E/S:



- Los procesos tienen E/S:

	0	1	2	3	4	5	6	7	8	9	10
P ₁						P	P	P	P	E	P
P ₂			P	E	E	E	W	P	E	F	
P ₃		E	E	W	P	P	P	E	P	P	E
P ₄				P	P	P	E	W	W	P	P

P: Preparado (Ready)

E: Ejecución (Run)

W: E/S (Blocked)

F: Finalizado (End)

Algoritmo FCFS (First-Come First-Served)

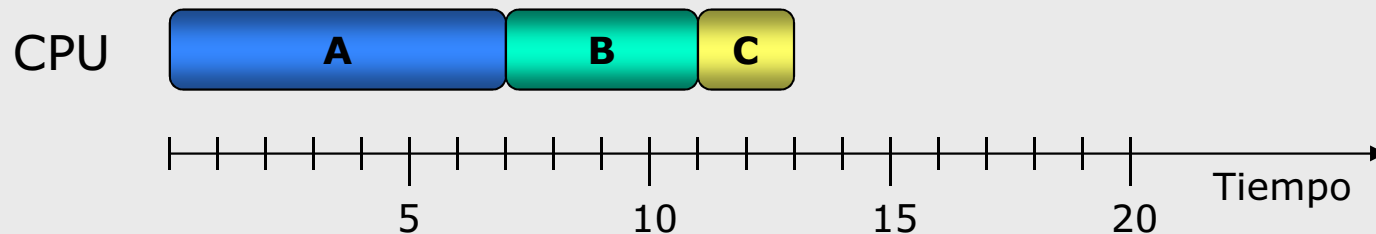
■ Funcionamiento:

- El procesador se asigna siguiendo el orden de llegada de los procesos a la cola de preparados.
- El proceso que tiene la CPU no la libera hasta que acaba su ejecución ó hasta que se tiene que bloquear por una operación de E/S.

■ Implementación:

- La cola de preparados se implementa mediante una cola FIFO (First-In First-Out).

Diagrama de Gant FCFS



$$Utilización_{cpu} = \frac{TCPU_{ocupada}}{Tiempo} = \frac{13}{13} = 1 = 100\% \quad Pr oductividad = \frac{n^{\circ} procesos}{tiempo} = \frac{3}{13} = 0,23$$

$$TEspera_{medio} = \frac{TEspera_A + TEspera_B + TEspera_C}{n^{\circ} procesos} = \frac{0 + 5 + 8}{3} = 4,3$$

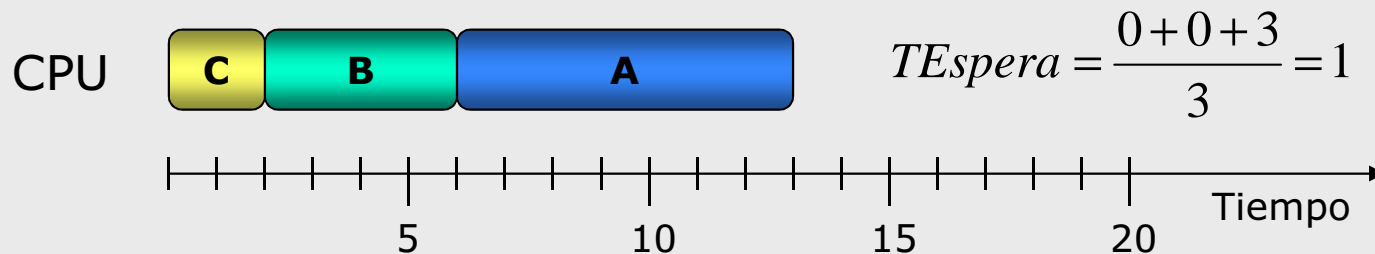
$$T Retorno_{medio} = \frac{T Retorno_A + T Retorno_B + T Retorno_C}{n^{\circ} procesos} = \frac{7 + 9 + 10}{3} = 8,6$$

$$T RetornoN_{medio} = \frac{T RetornoN_A + T RetornoN_B + T RetornoN_C}{n^{\circ} procesos} = \frac{7/7 + 9/4 + 10/2}{3} = 2,75$$

Características FCFS

- 👍 Simple de implementar
- 👎 Muy sensible al orden de llegada de los procesos.

Por ejemplo, si los procesos llegasen en tiempo 3, 2 y 0, tendríamos:



- 👎 Altos tiempos de espera
- 👎 Efecto convoy: tiende a favorecer a los procesos con mucha carga de CPU, perjudicando a los procesos intensivos en E/S

Algoritmo SJF (Shortest Job First)

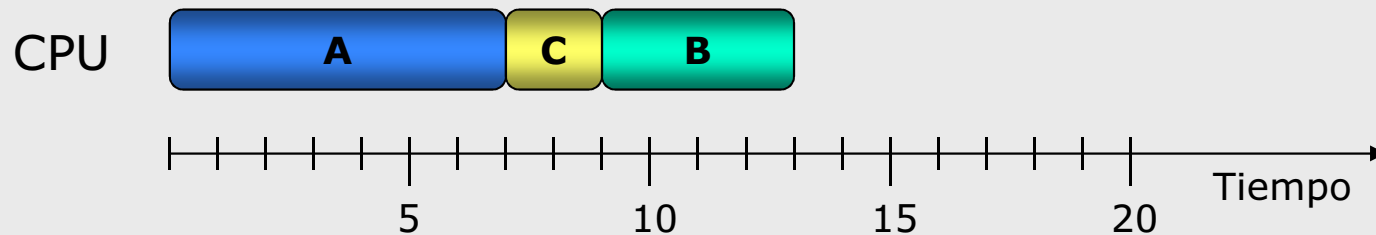
■ Funcionamiento:

- El procesador se asigna al proceso con la ráfaga de CPU más breve.
- En caso de empate se aplica otro algoritmo (normalmente el FCFS).
- No apropiativo (el proceso que tiene la CPU no la libera hasta que acaba su ejecución ó hasta que se tiene que bloquear).

■ Implementación:

- Ordenar la cola de preparados en función del tiempo de la siguiente ráfaga de CPU de los procesos.

Diagrama de Gant SJF



$$TEspera_{medio} = \frac{TEspera_A + TEspera_B + TEspera_C}{n^{\circ} \text{ procesos}} = \frac{0 + 7 + 4}{3} = 3,6$$

$$T \text{ Retorno}_{medio} = \frac{T \text{ Retorno}_A + T \text{ Retorno}_B + T \text{ Retorno}_C}{n^{\circ} \text{ procesos}} = \frac{7 + 11 + 6}{3} = 8$$

$$T \text{ Retorno}N_{medio} = \frac{T \text{ Retorno}N_A + T \text{ Retorno}N_B + T \text{ Retorno}N_C}{n^{\circ} \text{ procesos}} = \frac{\frac{7}{7} + \frac{11}{4} + \frac{6}{2}}{3} = 2,25$$

Características SJF

- 👍 Reduce el tiempo de espera medio
- 👍 Amortigua el efecto convoy
- 👎 Es difícil determinar a priori cual será la duración de la siguiente ráfaga de CPU de los procesos.

$$EstRaf_{n+1} = \alpha \cdot EstRaf_n + (1 - \alpha) \cdot Raf_n$$

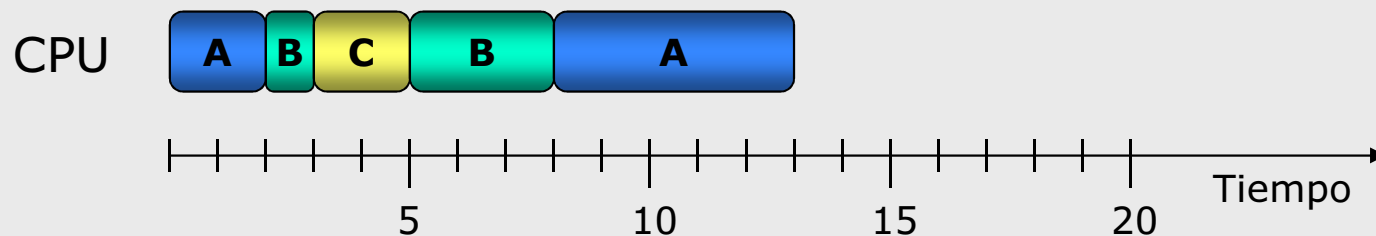
- 👎 Riesgo de inanición de los procesos intensivos en CPU.

Ejercicio:

- Realizar la planificación de los siguientes 3 procesos utilizando el algoritmo de planificación SJF apropiativo ó SRT (Short Remaining Time).

Proceso	Tiempo de llegada	Ráfagas del proceso
Proceso A	0	7 _{CPU}
Proceso B	2	4 _{CPU}
Proceso C	3	2 _{CPU}

Diagrama de Gant SJF apropiativo (ó SRT)



$$TEspera_{medio} = \frac{TEspera_A + TEspera_B + TEspera_C}{n^{\circ} \text{ procesos}} = \frac{6 + 2 + 0}{3} = 2,6$$

$$T \text{ Retorno}_{medio} = \frac{T \text{ Retorno}_A + T \text{ Retorno}_B + T \text{ Retorno}_C}{n^{\circ} \text{ procesos}} = \frac{13 + 6 + 2}{3} = 7$$

$$T \text{ Re spuesta}_{medio} = \frac{T \text{ Re spuesta}_A + T \text{ Re spuesta}_B + T \text{ Re spuesta}_C}{n^{\circ} \text{ procesos}} = \frac{0 + 0 + 0}{3} = 0$$

$$T \text{ Retorno}N_{medio} = \frac{T \text{ Retorno}N_A + T \text{ Retorno}N_B + T \text{ Retorno}N_C}{n^{\circ} \text{ procesos}} = \frac{13/7 + 6/4 + 2/2}{3} = 1,45$$

Algoritmo por prioridades

■ Funcionamiento:

- Cada proceso tiene asignada un valor entero que representa su prioridad
- El planificador escoge el proceso de la cola de preparados que tiene la mayor prioridad.

■ Implementación:

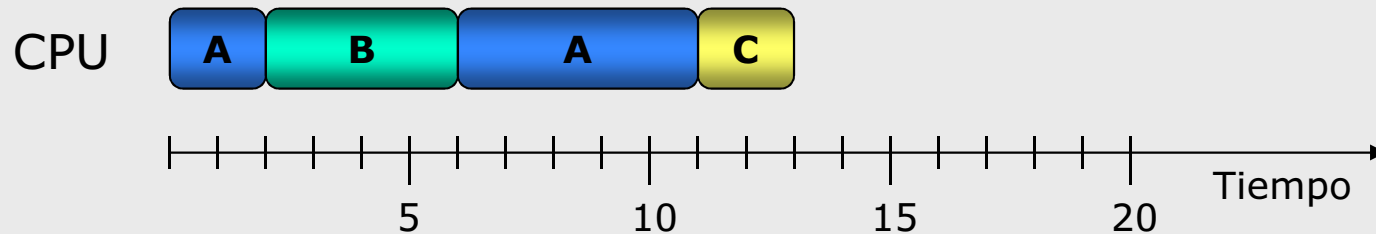
- Cola de preparados ordenada por la prioridad de los procesos.

■ Opciones:

- La política puede ser apropiativa ó no.
- Las prioridades se pueden definir de forma interna (por el SO) ó de forma externa (por el usuario).
- Prioridades estáticas ó dinámicas.

Diagrama de Gant prioridades apropiativo

- Suponemos que los procesos tienen las siguientes prioridades A=5, B=1 y C=6 (1 la prioridad más alta y 9 la más baja).



$$TEspera_{medio} = \frac{TEspera_A + TEspera_B + TEspera_C}{n^{\circ} \text{ procesos}} = \frac{4 + 0 + 8}{3} = 4$$

$$T \text{ Retorno}_{medio} = \frac{T \text{ Retorno}_A + T \text{ Retorno}_B + T \text{ Retorno}_C}{n^{\circ} \text{ procesos}} = \frac{11 + 4 + 10}{3} = 8,33$$

$$T \text{ Respuesta}_{medio} = \frac{T \text{ Respuesta}_A + T \text{ Respuesta}_B + T \text{ Respuesta}_C}{n^{\circ} \text{ procesos}} = \frac{0 + 0 + 8}{3} = 2,66$$

Envejecimiento

- Problema algoritmos basados en prioridades:
 - La inanición de los procesos con una menor prioridad.
- Solución:
 - Envejecimiento de la prioridad de los procesos.
 - Se aumenta periódicamente la prioridad a los procesos preparados y que no consiguen ejecutarse.

Algoritmo Round-Robin (turno rotativo)

■ Funcionamiento:

- Se asigna la CPU a cada proceso durante un intervalo de tiempo prefijado (denominado *quantum*) y de forma rotativa.
- El orden de asignación de la CPU es generalmente por orden de llegada (FIFO).

■ Implementación:

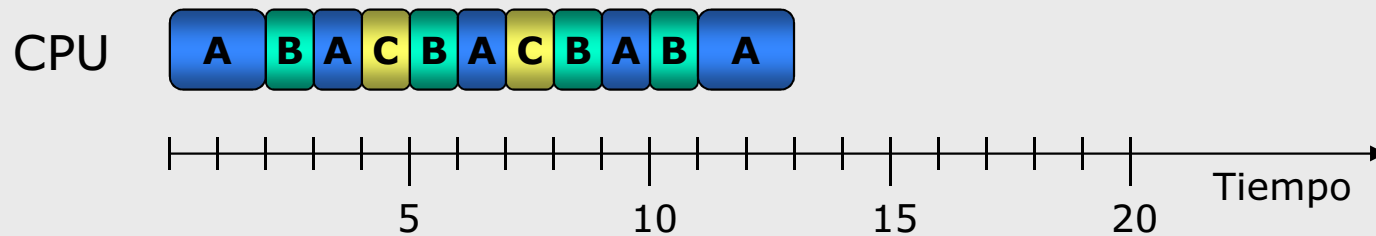
- Se requiere de un temporizador que cada *quantum* unidades de tiempo invoque al planificador.

■ Características:

- Permite acotar el tiempo de respuesta de los procesos.
- Algoritmo ideal para sistemas de tiempo compartido ó procesos interactivos.

Diagrama de Gant Round-Robin

- Suponemos quantum=1



$$TEspera_{medio} = \frac{TEspera_A + TEspera_B + TEspera_C}{n^{\circ} procesos} = \frac{6 + 5 + 3}{3} = 4,66$$

$$T Retorno_{medio} = \frac{T Retorno_A + T Retorno_B + T Retorno_C}{n^{\circ} procesos} = \frac{13 + 9 + 5}{3} = 9$$

$$T Re spuesta_{medio} = \frac{T Re spuesta_A + T Re spuesta_B + T Re spuesta_C}{n^{\circ} procesos} = \frac{0 + 0 + 1}{3} = 0,33$$

$$T RetornoN_{medio} = \frac{T RetornoN_A + T RetornoN_B + T RetornoN_C}{n^{\circ} procesos} = \frac{13/7 + 9/4 + 5/2}{3} = 2,20$$

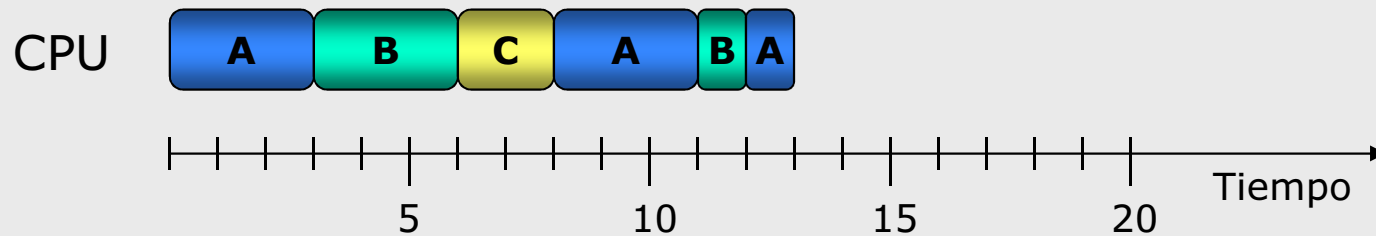
Ejercicio:

- Realizar la planificación de los 3 procesos del ejemplo utilizando un algoritmo de planificación Round-Robin con $\text{quantum}=3$.

Proceso	Tiempo de llegada	Ráfagas del proceso
Proceso A	0	7 _{CPU}
Proceso B	2	4 _{CPU}
Proceso C	3	2 _{CPU}

Diagrama de Gant Round-Robin

- Suponemos quantum=3



$$TEspera_{medio} = \frac{TEspera_A + TEspera_B + TEspera_C}{n^{\circ} \text{ procesos}} = \frac{6 + 6 + 3}{3} = 5$$

$$T \text{ Retorno}_{medio} = \frac{T \text{ Retorno}_A + T \text{ Retorno}_B + T \text{ Retorno}_C}{n^{\circ} \text{ procesos}} = \frac{13 + 10 + 5}{3} = 9,33$$

$$T \text{ Re spuesta}_{medio} = \frac{T \text{ Re spuesta}_A + T \text{ Re spuesta}_B + T \text{ Re spuesta}_C}{n^{\circ} \text{ procesos}} = \frac{0 + 1 + 3}{3} = 1,33$$

Resumen algoritmos de planificación

	Tiempo Espera	Tiempo Retorno	Tiempo Respuesta	% utilización (CC=0,25)
FCFS	4,3	8,6	4,3	96%
SJF	3,6	8	3,6	96%
Prioridades	4	8,33	2,66	94%
Round-Robin (Q=1)	4,66	9	0,33	81%

- No hay un algoritmo que saque los mejores resultados en todas las métricas.
- La elección del algoritmo depende de las necesidades concretas del tipo de proceso:
 - Procesos interactivos: Minimizar tiempo de respuesta.
 - Procesos SO: Ejecución prioritaria.
 - Procesos por lotes: minimizar tiempo espera y maximizar rendimiento.

Ejercicio:

- Realizar la planificación de los siguientes 3 procesos utilizando los algoritmos:
 - a) Reparto de tiempo (round-robin) con quantum de 3 ciclos.
 - b) Menor tiempo primero (SJF)
- Calcular el tiempo medio de espera, tiempo de respuesta y de retorno.

Proceso	Tiempo de llegada	Ráfagas del proceso
Proceso A	7	5 _{CPU} , 2 _{E/S} , 4 _{CPU}
Proceso B	3	5 _{CPU} , 1 _{E/S} , 1 _{CPU} , 1 _{E/S} , 1 _{CPU}
Proceso C	1	2 _{CPU} , 1 _{E/S} , 3 _{CPU} , 2 _{E/S} , 1 _{CPU}

Solución (Round-Robin):

	0	1	2	3	4	5	6	7	8	9	10	11	12
A								E	E
B				E	E	E	.	.	.	E	E	W	.
C		E	E	W	.	.	E	E	E	W	W	.	.

13	14	15	16	17	18	19	20	21	22	23	24
E	.	.	E	E	W	W	E	E	E	E	F
.	.	E	W	.	E	F					
.	E	F									

Tiempo de espera medio = $(6+7+5)/3 = 6$ ut

Tiempo de respuesta medio = $(4+0+0)/3 = 1,33$ ut

Tiempo de retorno medio = $(17+16+14)/3 = 15,67$ ut

Solución (SJF):

	0	1	2	3	4	5	6	7	8	9	10	11	12
1								E
2				E	E	E	E	E	W	.	.	E	W
3		E	E	W	E	E	E	W	W

13	14	15	16	17	18	19	20	21	22	23	24
E	E	E	E	W	W	E	E	E	E	F	
.	.	.	.	E	F						
.	E	F					

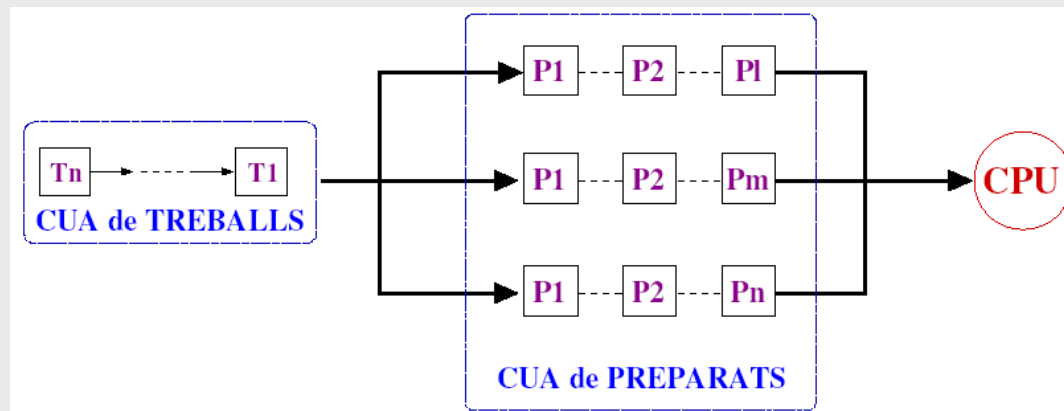
Tiempo de espera medio = $(5+6+9)/3 = 6,67$ ut

Tiempo de respuesta medio = $(5+0+0)/3 = 1,67$ ut

Tiempo de retorno medio = $(16+15+18)/3 = 16,33$ ut.

Colas multinivel sin realimentación

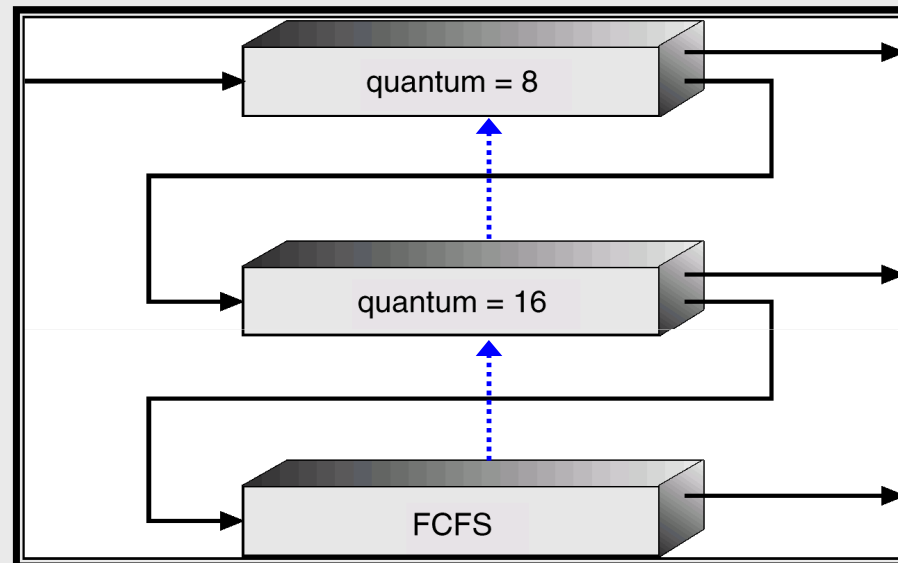
- Se dispone de varias colas de preparados, cada una gestionada mediante una política diferente.
- Los procesos en función de sus características son asignados a una cola específica y no se pueden mover entre colas (sin realimentación)
- Las colas se reparten la CPU entre ellas utilizando alguna política:
 - Por prioridad absoluta.
 - Un porcentaje de tiempo (quantum) para cada cola.



Colas multinivel con realimentación

- Se permite la posibilidad de que un proceso se mueva entre las colas.
- Objetivo: Separar los procesos según su comportamiento dinámico
 - Procesos intensivos en E/S quedan en colas más prioritarias
 - Procesos intensivos en CPU en colas menos prioritarias
- Criterios para cambiar de cola un proceso:
 - El proceso cambia su comportamiento durante su ejecución.
 - Un proceso que ha sido planificado varias veces, puede ser movido a una cola menos prioritaria para que no acapare la CPU y viceversa.

Ejemplo colas multinivel con realimentación



- Todos los procesos entran en la cola más prioritaria y a medida que van consumiendo CPU se mueven a colas menos prioritarias.
- Para evitar la inanición de los procesos situados en colas menos prioritarias, éstos pueden envejecer su prioridad con el paso del tiempo ser promocionados a una cola más prioritaria.

Ejercicio: Colas Multinivel (I)

- En un sistema multiprogramado que dispone de **1 procesador** se ejecutan **6 procesos** con las características siguientes:

Proceso	Tipo de Proceso	Prioridad (1 mejor)	Cola	Tiempo llegada	Ráfagas del proceso
Proceso A	Tiempo real	3	1	5	4 _{CPU} , 1 _{E/S} , 5 _{CPU}
Proceso B	Tiempo real	1	1	7	1 _{CPU} , 3 _{E/S} , 2 _{CPU} , 5 _{E/S} , 1 _{CPU}
Proceso C	Interactivo	2	2	2	1 _{CPU} , 5 _{E/S} , 2 _{CPU}
Proceso D	Interactivo	4	2	4	1 _{CPU} , 2 _{E/S} , 3 _{CPU} , 1 _{E/S} , 1 _{CPU}
Proceso E	Proceso batch	9	3	1	3 _{CPU} , 1 _{E/S} , 4 _{CPU}
Proceso F	Proceso batch	7	3	5	1 _{CPU} , 1 _{E/S} , 1 _{CPU}

- Se pide:
 - a) Planificar la ejecución de los procesos sobre la siguiente tabla e indicar el estado de cada proceso (E: en ejecución, W: entrada / salida, B: bloqueado P: preparado, F: finalizado, si hacen falta más estados añadirlos).
 - b) Calcular grado de utilización de la CPU, la productividad, el tiempo medio de espera para cada una de las colas, y el tiempo de retorno normalizado para cada uno de los procesos.

Ejercicio: Colas Multinivel (II)

- El planificador dispone de un algoritmo de planificación a corto plazo con 3 colas multinivel:
 - La cola 1 planifica los procesos de tiempo real y utiliza un algoritmo de prioridad apropiativo. Esta cola es la más prioritaria.
 - La cola 2 planifica los procesos de usuario y utiliza un algoritmo de reparto de tiempo (Round-Robin) con un quantum de 2 ciclos. Esta cola tiene una prioridad intermedia entre las otras dos colas.
 - La cola 3 planifica los procesos Batch y utiliza un algoritmo SRT (el que le quede menos tiempo primero). Esta cola es la menos prioritaria.
- La planificación entre las colas sigue un algoritmo prioritario, de manera que solo se puede ejecutar un proceso de una cola menos prioritaria si no hay ninguno preparado en las colas más prioritarias.
 - Los procesos más prioritarios se **apropian inmediatamente** del procesador en el momento que pueden ejecutarse.
- En caso de empate total a la hora de planificar los procesos utilizar la prioridad como último criterio.

Planificación en sistemas multiprocesador

- En el caso de haya que planificar más de un procesador se dispone de diversas opciones:
 - Colas de planificación independientes.
Problema: Desbalanceo de carga.
 - Cola de planificación compartida.
Problema: Actualización simultánea de la estructura de datos compartidas.
 - Un procesador es el encargado de la planificación.
- Los algoritmos de planificación en entornos multiprocesador son los mismos que en entornos monoprocesador.

Ejercicio:

- Repetir el ejercicio anterior pero esta vez para un sistema con 2 procesadores que funciona bajo el esquema de una única cola de preparados.
- Realizar la planificación de los siguientes 3 procesos utilizando los algoritmos:
 - a) Reparto de tiempo con quantum de 3 ciclos.
 - b) Menor tiempo primero (SJF)
- Calcular el tiempo medio de espera, tiempo de respuesta y de retorno.

Proceso	Tiempo de llegada	Ráfagas del proceso
Proceso A	7	5 _{CPU} , 2 _{E/S} , 4 _{CPU}
Proceso B	3	5 _{CPU} , 1 _{E/S} , 1 _{CPU} , 1 _{E/S} , 1 _{CPU}
Proceso C	1	2 _{CPU} , 1 _{E/S} , 3 _{CPU} , 2 _{E/S} , 1 _{CPU}

Solución (Round-Robin):

	0	1	2	3	4	5	6	7	8	9	10	11	12
A								E	E	E	E	E	W
B				E	E	E	E	E	W	E	W	E	F
C		E	E	W	E	E	E	W	W	P	E	F	

13	14	15	16	17	18	19	20	21	22	23	24
W	E	E	E	E	F						

Tiempo de espera medio = $(0+0+1)/3 = 0,33$ ciclos

Tiempo de respuesta medio = $(0+0+0)/3 = 0$ ciclos

Tiempo de retorno medio = $(11+9+10)/3 = 10$ ciclos

Solución (SJF apropiativo):

	0	1	2	3	4	5	6	7	8	9	10	11	12
A								E	E	P	E	E	E
B				E	E	E	E	E	W	E	W	E	F
C		E	E	W	E	E	E	W	W	E	F		

13	14	15	16	17	18	19	20	21	22	23	24
W	W	E	E	E	E	F					

Tiempo de espera medio = $(1+0+0)/3 = 0,33$ ciclos

Tiempo de respuesta medio = $(0+0+0)/3 = 0$ ciclos

Tiempo de retorno medio = $(12+9+9)/3 = 10$ ciclos

Planificación en Unix (I)

- Objetivos planificador de Linux:
 - Tiempo de respuesta rápido para los procesos.
 - Buena productividad de los trabajos en background
 - Evitar la inanición de los procesos.
 - Reconciliar necesidades de los procesos de alta y baja prioridad.
- Linux da preferencia a los procesos intensivos en E/S (interactivos).
 - Procesos que utilizan mucho la CPU se les baja la prioridad.
 - Procesos que no utilizan asiduamente la CPU se les sube la prioridad.

Planificación en Unix (II)

- La planificación se basa en la técnica de reparto de tiempo (Round-Robin) basado en prioridades.
 - Algoritmo de planificación apropiativo.
- Los procesos se clasifican de acuerdo a su prioridad:
 - Procesos **Tiempo Real**: todos aquellos que tienen una prioridad entre 1 y 99.
 - Prioridad estática que no se modifica.
 - Procesos **Convencionales**: resto de procesos con un prioridad superior entre 100 y 139.
 - Prioridad Dinámica: La prioridad de los procesos comunes es dinámica y se reajusta periódicamente:
 - Prioridad base del proceso ($120 + \text{nice}$)
 - Quantum que el proceso no ha utilizado en la última etapa de planificación

Planificación en Unix (III)

- Planificador basado en colas multinivel no realimentadas:
 - Cola procesos tiempo-real: FCFS basado en prioridades (*SCHED_FIFO*)
 - Algoritmo no apropiativo.
 - Cola procesos tiempo-real: Round-Robin basado en prioridades (*SCHED_RR*).
 - Algoritmo apropiativo.
 - Cola resto procesos: Round-Robin (*SCHED_OTHER* - *SCHED_NORMAL*).

Planificación en Unix (IV)

- El algoritmo de planificación divide el tiempo de CPU en épocas.
 - Al inicio de cada época se recalcula las prioridades dinámicas y con éstas el quantum de cada proceso.
 - $Quantum_i = Quantum_Prio + Quantum_i$
 - El planificador controla el quantum que le queda pendiente a cada proceso.
 - Cuando un proceso agota su quantum, se envían a la cola de proc. expirados y no se vuelve a planificar hasta la siguiente época
 - Un proceso que hace E/S y no agota el quantum puede volver a la cola de preparados.
 - Cuando un proceso crea un proceso hijo, su quantum pendiente se divide entre él y el hijo
- Las épocas finalizan cuándo todos los procesos ejecutables (preparados) han acabado su quantum.

Planificación en Windows

Clases de Prioridad (procesos)

Modificadores (hilos)		real-time	high	above normal	normal	below normal	idle priority
	time-critical	31	15	15	15	15	15
	highest	26	15	12	10	8	6
	above normal	25	14	11	9	7	5
	normal	24	13	10	8	6	4
	below normal	23	12	9	7	5	3
	lowest	22	11	8	6	4	2
	idle	16	1	1	1	1	1

- Colas Multinivel con Realimentación. Cada prioridad tiene asociada una cola con planificación Round-Robin.
- Prioridades 0-15 variables, 16-31 fijas (tiempo real).
- A los hilos que agotan su quantum se les reduce la prioridad. Cuando un hilo pasa de espera a listo se aumenta su prioridad

Colas multinivel en Windows

