Tema 4. Planificación de monoprocesadores

Dr. José Raúl Fernández del Castillo Díez

Dr. Juan José Sánchez Peña

Tipos de planificación (I)

Planificación a largo plazo	Decisión de añadir procesos al conjunto de procesos a ejecutar.
Planificación a medio plazo	Decisión de añadir procesos al conjunto de procesos que se encuentran parcial o completamente en la memoria.
Planificación a corto plazo	Decisión sobre qué proceso disponible será ejecutado en el procesador
Planificación de E/S	Decisión sobre qué solicitud de E/S pendiente será tratada por un dispositivo de E/S disponible

Tipos de planificación (II)

- Largo
  - Determina cuáles son los programas admitidos en el sistema.
  - Controla el grado de multiprogramación.
  - Cuantos más procesos se crean, menor es el porcentaje de tiempo en el que cada proceso se puede ejecutar.
- Medio
  - Forma parte de la función de intercambio.
  - Se basa en la necesidad de controlar el grado de multiprogramación.
- Corto.
  - También conocido como distribuidor.
  - Es el de ejecución más frecuente.
  - Se ejecuta cuando ocurre un suceso:
    - Interrupciones del reloj.
    - Interrupciones de E/S.
    - Llamadas al sistema operativo.
    - Señales.

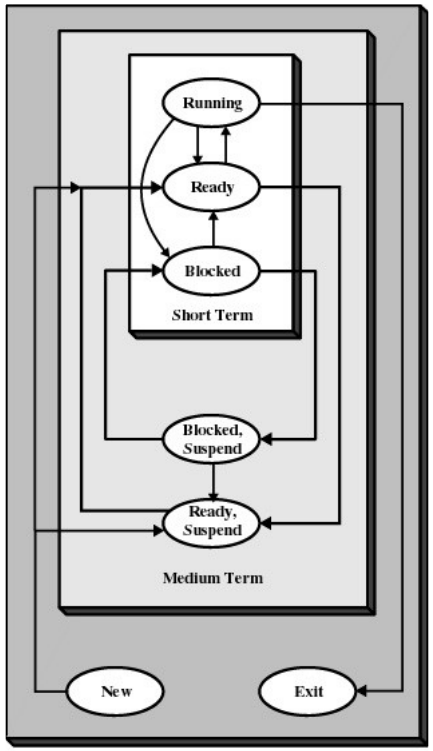


Figure 9.2 Levels of Scheduling

Objetivos y conceptos.

- Tiempo de respuesta.
  - Tiempo transcurrido entre el lanzamiento de un proceso y el momento en el que éste comienza a dar resultados (primera asignación).
  - Valor perceptible por el usuario.
- Tiempo de retorno.
  - Intervalo de tiempo transcurrido entre el lanzamiento de un proceso y su finalización.
  - Resulta de la suma de la espera (de ser planificado y en espera de los recursos) y del tiempo de ejecución real.
- Productividad.
  - Número de procesos terminados por unidad de tiempo.

Objetivos y conceptos: planificación corto plazo (I)

- Dos procesos.
  - Planificador: Scheduler.
  - Ejecutor de la planificación: Dispatcher.

  Se activan por IRQ, eventos (semáforos, I/O,etc.), fin de proceso.
- Directrices
  - Distribución equitativa del tiempo de CPU.
  - Tiempo de respuesta mínimo.
  - Productividad: nº máximo de trabajos por unidad de tiempo.
  - Eficiencia del procesador: CPU ocupada al 100%.
- Activación.
  - Interrupciones del reloj.
  - Interrupciones de E/S.
  - Llamadas al sistema operativo.
  - Señales.

Objetivos y conceptos: planificación corto plazo (II)

- También conocido como distribuidor.
- Es el de ejecución más frecuente.
- Optimiza el sistema.
- Criterios de actuación:
  - **Orientado al sistema.**
    - Uso máximo de la CPU.
    - Máximo número de procesos terminados.
  - **Orientados al usuario**
    - Tiempo mínimo de respuesta (al usuario).
    - Tiempo mínimo de asignación del proceso a la CPU.

Objetivos y conceptos: planificación corto plazo (III)

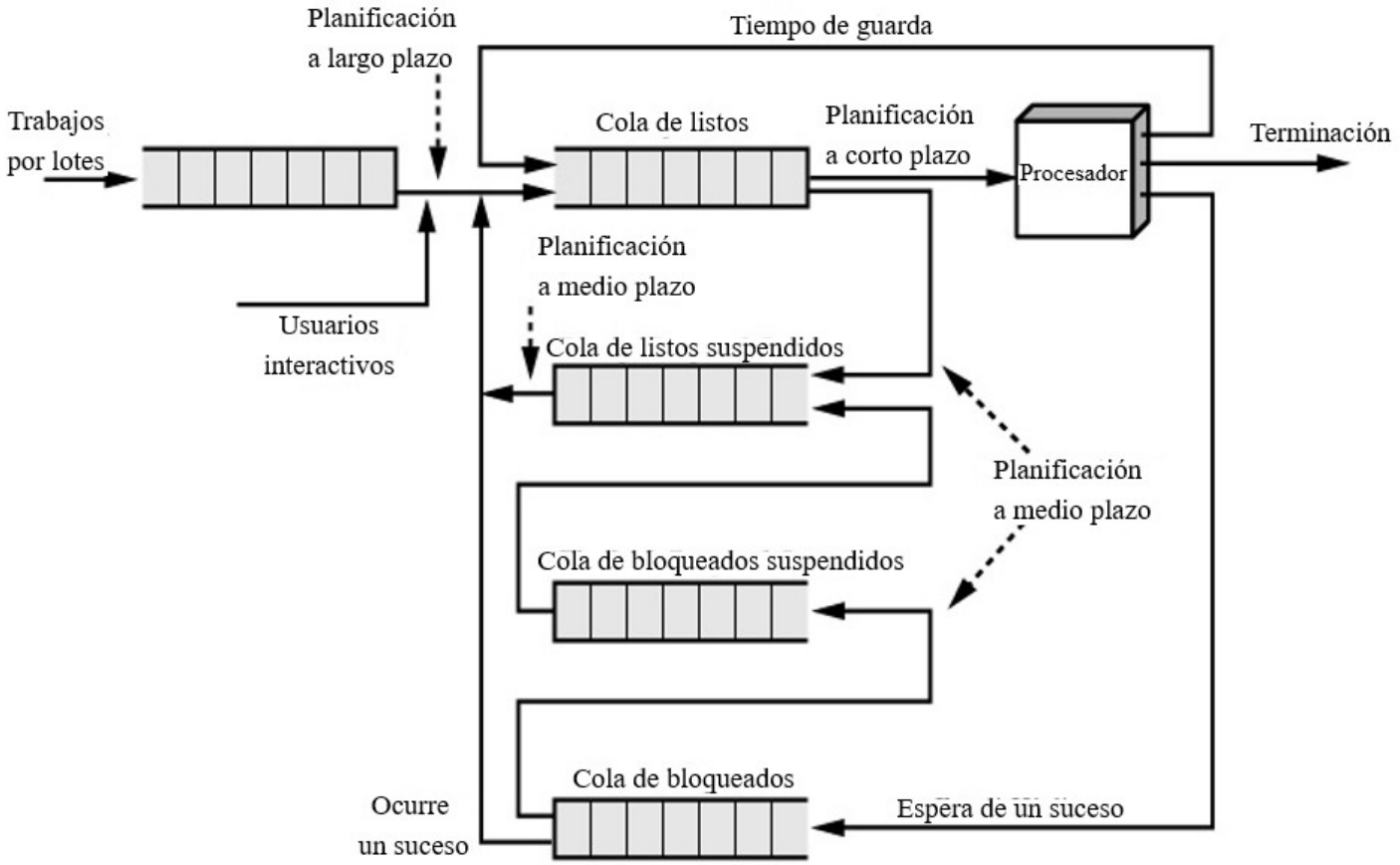


Diagrama de colas de planificación.

Algoritmos de asignación: tipos de asignación

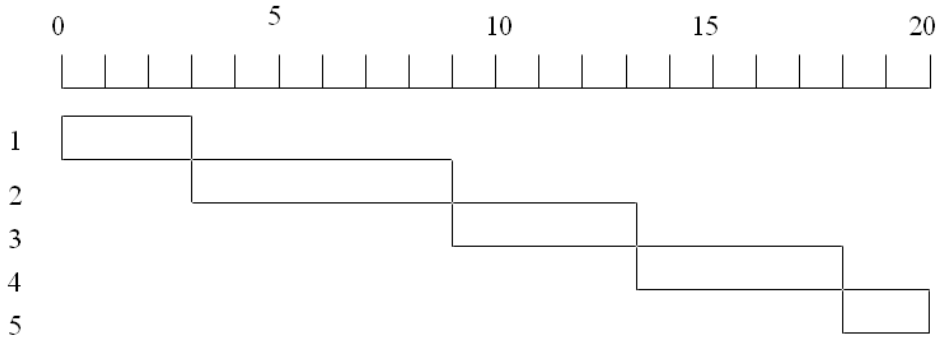
- No preferente:
  - Una vez que el proceso pasa al estado de Ejecución, continúa ejecutando hasta que termina o se bloquea en espera de una E/S.
- Preferente:
  - El proceso que se está ejecutando actualmente puede ser interrumpido y pasado al estado de Listos por el sistema operativo.
  - Permiten dar un mejor servicio ya que evitan que un proceso pueda monopolizar el procesador durante mucho tiempo.

    - First Come Firts Served (FCFS).
    - Turno rotatorio (Round Robin, RR).
    - Shortest Job First (SJF) o Primero en trabajo más corto.
    - Menor ráfaga restante.
    - Mayor tasa de respuesta.

Ejemplo de planificación de procesos

Proceso	Instante de llegada	Tiempo de servicio
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Primero en llegar, primero en servirse (FCFS)

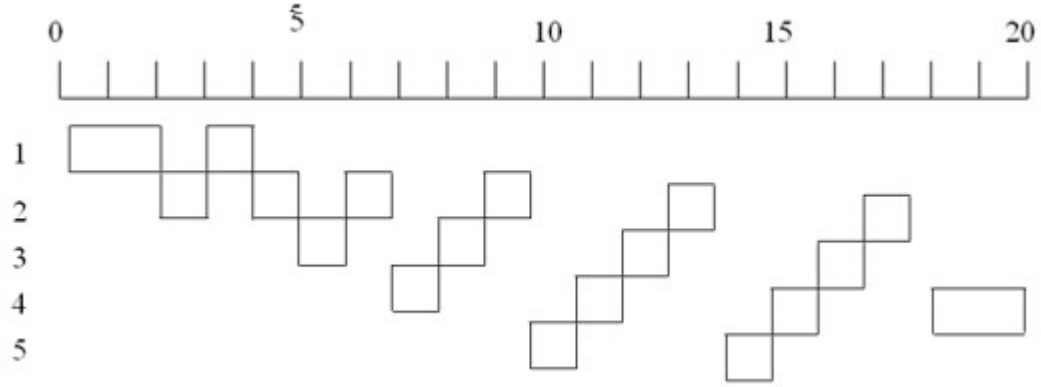


- **Cada proceso se incorpora a la cola de listos.**
- **Cuando el proceso actual cesa su ejecución, se selecciona el proceso más antiguo de la cola.**

- Puede que un proceso corto tenga que esperar mucho tiempo antes de que pueda ser ejecutado.
- No requisa → malo para sistemas interactivos.
- Favorece a los procesos con carga de CPU:
  - **Efecto convoy:** los procesos con carga de E/S tienen que esperar a que se completen los procesos con carga de CPU.

Turno rotatorio (Round Robin)

- Utiliza la requisa dependiente de un reloj. Periódicamente, se genera una interrupción de reloj.
- Se determina una cantidad de tiempo (***quanto***) que permite a cada proceso utilizar el procesador durante este periodo de tiempo.
- Cuando se genera la interrupción, el proceso que está en ejecución se sitúa en la cola de Listos y se selecciona el siguiente trabajo.

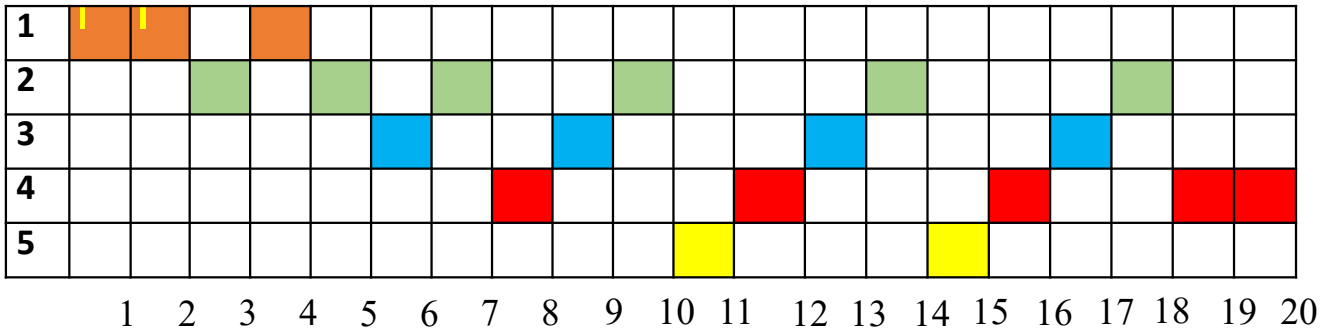


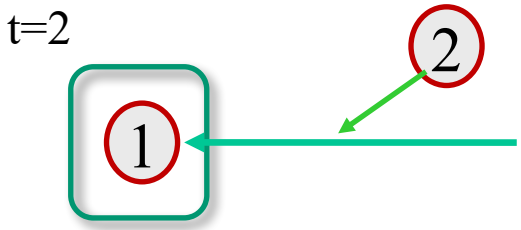
- Se conoce también como fracciones de tiempo.
- El cambio de contexto: Necesario equilibrio entre Quanto largo y corto.

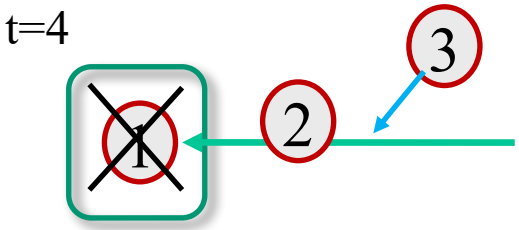
Ej.:

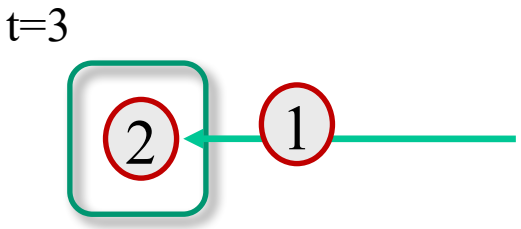
Q=20 mseg con c.contex.= 5 mseg ➔ 20% de tiempo en cambios.
Q=500 mseg con c.contex.= 5 mseg ➔ 1% de tiempo en cambios.
Pero si Q = 500 mseg ➔ tiempo de respuesta a usuario lento.

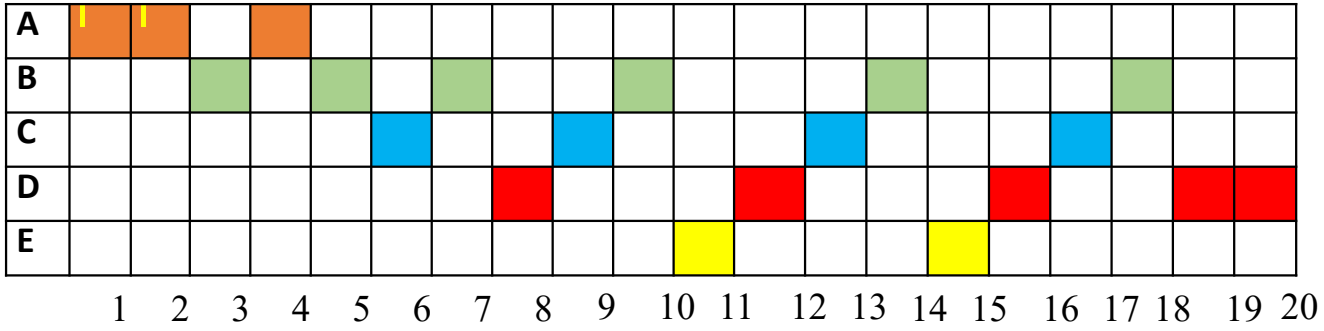
Ajuste del Q para que el 80% de las ráfagas de CPU sean menores que Q.

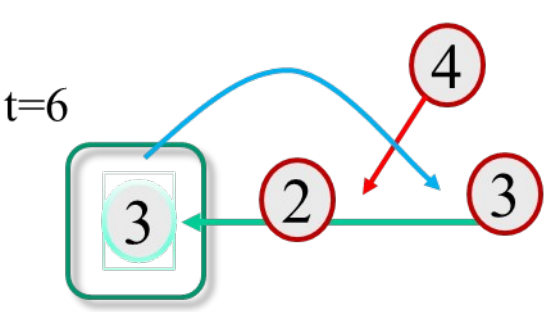

1
2
3
4
5
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

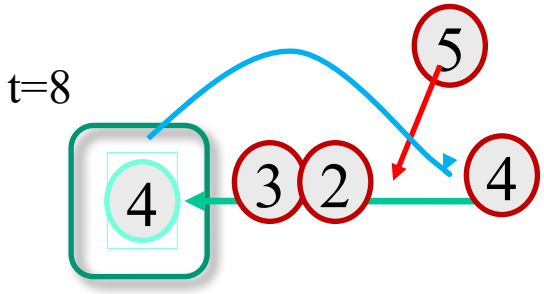

t=2
2
1


t=4
3
2
1

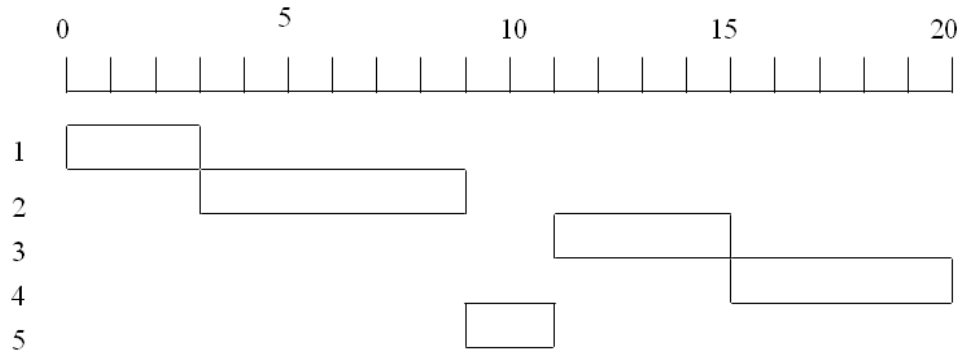

t=3
2
1


A
B
C
D
E
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20


t=6
4
3
2
3


t=8
5
4
3
2
4

Primero el proceso más corto



- Es una política no preferente.
- Se selecciona el proceso con menor tiempo esperado de ejecución.
- Un proceso corto saltará a la cabeza de la cola, sobrepasando a trabajos largos.

- Se reduce la previsibilidad de los procesos largos.
- Si la estimación de tiempo del proceso no es correcta, el sistema puede abandonar el trabajo.
- Posibilidad de inanición para los procesos largos.
- Estimación del futuro desde el pasado: media exponencial.

  $\tau_{n+1} = \alpha T_n + (1-\alpha)\,\tau_n$

  $\alpha = 0 \rightarrow \tau_{n+1} = \alpha T_n + (1-\alpha)\,\tau_n = \tau_n$

  $\alpha = 1 \rightarrow \tau_{n+1} = \alpha T_n + (1-\alpha)\,\tau_n = T_n$

$\tau_{n+1} = \alpha T_n + (1-\alpha)\,\tau_n = \tau_{n+1} = \alpha T_n + (1-\alpha)(T_{n-1} + (1-\alpha)\,\tau_{n-1})$

$\tau_n = \alpha T_{n-1} + (1-\alpha)\,\tau_{n-1}$

$\tau_{n-1} = \alpha T_{n-2} + (1-\alpha)\,\tau_{n-2}$

$\tau_{n-2} = \alpha T_{n-3} + (1-\alpha)\,\tau_{n-3}$

$\tau_{n+(n-2)} = \alpha T_{n-(n-1)} + (1-\alpha)\,\tau_{n-(n-1)}$

$\tau_1 = \alpha T_0 + (1-\alpha)\,\tau_0 = \alpha T_0 + (1-\alpha)\,(\alpha T_n + (1-\alpha)\,\tau_n)$

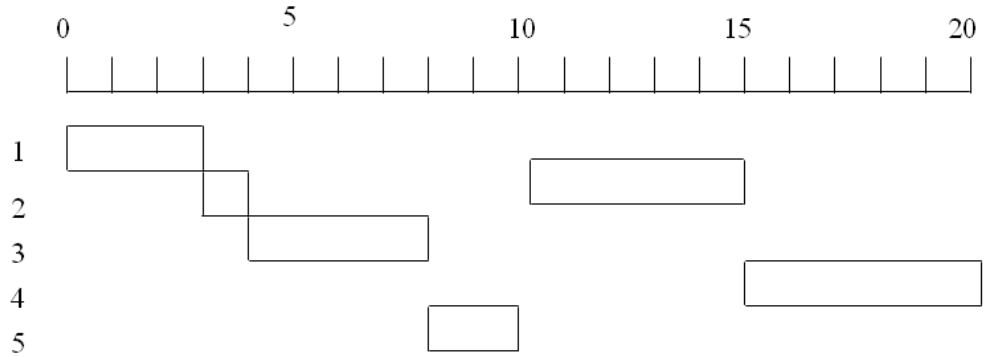
$\tau_{n+1} = \alpha T_n + (1-\alpha)\,\alpha\, T_{n-1} + (1-\alpha)^2\,\alpha\, T_{n-2} + (1-\alpha)^3\,\alpha\, T_{n-3} + (1-\alpha)^4\,\alpha\, T_{n-4}\,..$

$...+ (1-\alpha)^i\,\alpha\, T_{n-i} + ... + (1-\alpha)^n\,\alpha\, T_0$

Consideraciones:

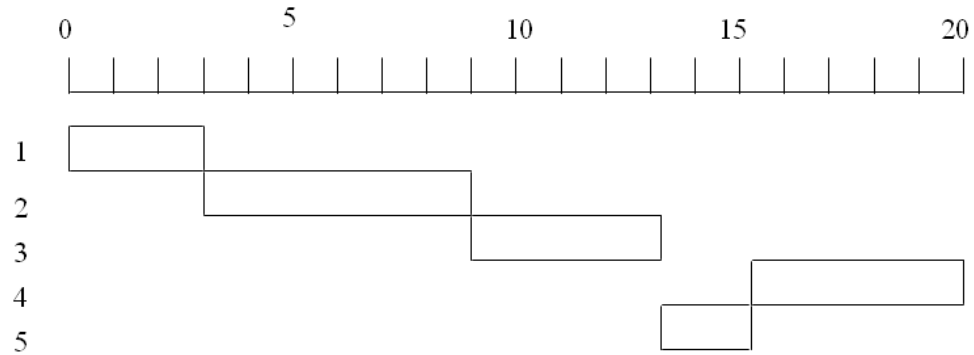
$(1-\alpha)<1 \rightarrow (1-\alpha)^n \rightarrow 0$ si n → infinito

Menor tiempo restante (SJF con requisa)



- Es una versión preferente de la política de primero el proceso más corto.
- Debe estimar el tiempo de proceso.

Primero el de mayor tasa de respuesta (HRRN)



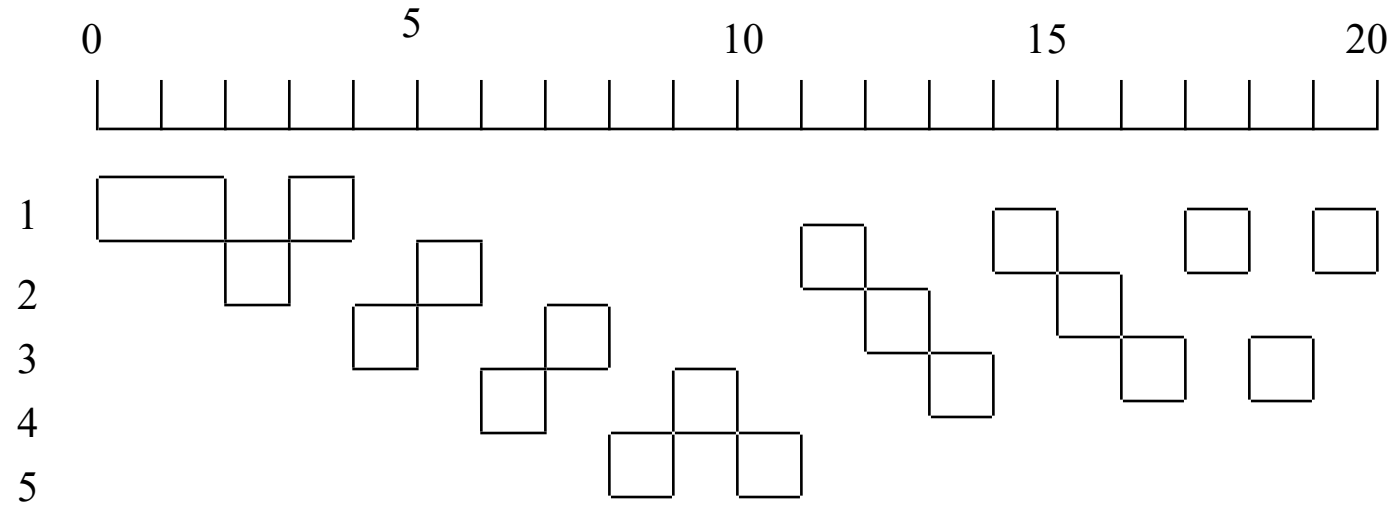
- Elige el proceso con la tasa más alta.

$$\frac{\text{tiempo consumido esperando al procesador + tiempo de servicio esperado}}{\text{tiempo de servicio esperado}}$$

t=9	T espera	T resp	Tasa de resp.
1			
2			
3	5	4	2.25
4	3	5	1.6
5	1	2	1.5

t=13	T espera	T resp	Tasa de resp.
1			
2			
3			
4	7	5	2.4
5	5	2	3.5

Realimentación



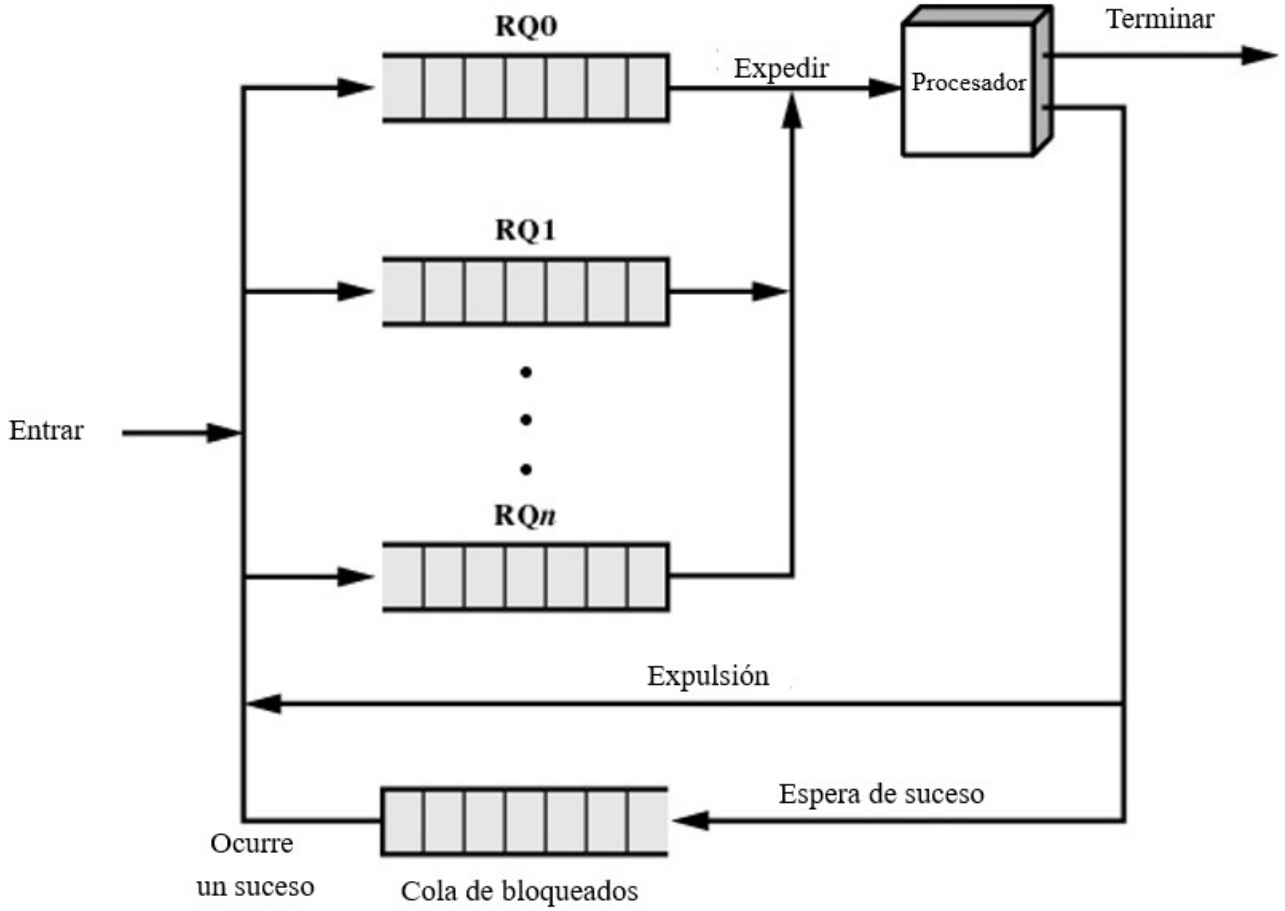
- No se conoce el tiempo de ejecución restante del proceso.
- Se estima el orden de planificación en función del pasado de los procesos.
- Penaliza a los trabajos que han estado ejecutándose durante más tiempo.

Planificación por prioridades (I)

- **Cada proceso recibe una prioridad.** Los procesos Listos se clasifican en colas dependiendo de la prioridad.
- El planificador seleccionará primero los procesos de mayor prioridad antes que a los de menor prioridad.
- Tiene múltiples colas de Listos para representar cada nivel de prioridad.
- Los procesos de prioridad más baja pueden sufrir inanición:
  - Permite que un proceso cambie su prioridad en función de su edad o su historial de ejecución.
- Existe planificación entre colas: reasignación dinámica.

- **Asignación de las prioridades:**
  - **Internamente**: Cantidad medible.
    - Requerimientos de MEM.
    - Nº de archivos / recursos.
    - Relación entre la I/O y el tiempo de CPU.
  - **Externamente**: Parámetros subjetivos.
    - Importancia del proceso / usuario.
    - Políticas.
    - Económicas.

Planificación por prioridades (II)



Colas de prioridad.

Planificación con colas múltiples

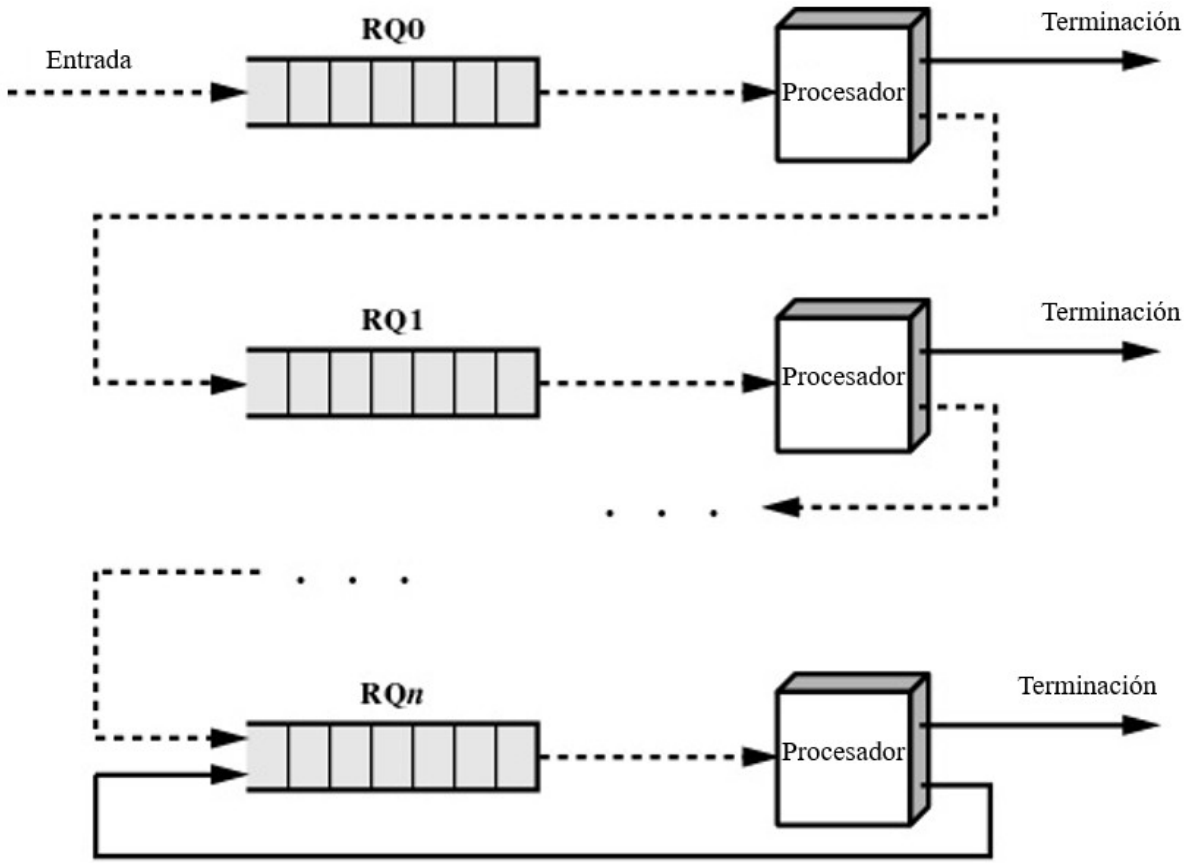
- Los procesos Listos se clasifican en colas dependiendo de alguna propiedad.
  - Tamaño de memoria.
  - Tipo de proceso.
  - Prioridad.
  - Anterior uso de la CPU.
  - Relación de I/O, etc.
- Necesario un mecanismo de planificación entre colas (prioridad más alta la cola de procesos con mayor prioridad).
- Cada cola su propio algoritmo de planificación.
- Ej.: Dos colas:
  - Prioridad alta: Procesos interactivos.
  - Prioridad baja: Procesos de segundo plano.

Planificación con colas múltiples realimentadas (I)

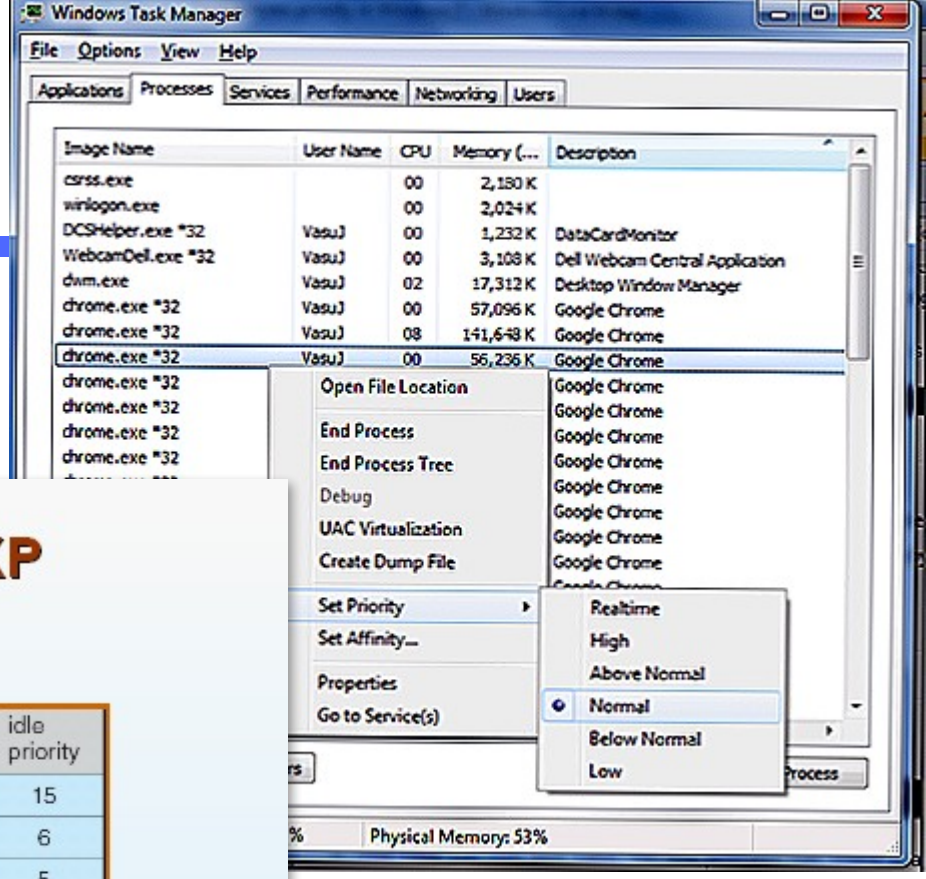
- La prioridad y/o la asignación a las colas cambia dinámicamente.
  - Mucho uso de CPU → baja la prioridad.
  - Mucha I/O e interactivos → sube la prioridad.
  - Mucha espera de CPU → sube la prioridad.
  - Según se hunden en las colas su Q sube. Se premian procesos cortos.

- Definición de los sistemas de colas:
  - Nº de colas.
  - Algoritmo de planificación entre colas a la baja.
  - Algoritmo de planificación entre colas al alza.
  - Algoritmo de planificación de cada cola.
  - Método de asignación inicial a la cola.

Planificación con colas múltiples realimentadas (II)



Planificación con realimentación.



Prioridades en Windows XP

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

- El algoritmo es de Colas Multinivel con Realimentación. Cada prioridad tiene asociada una cola con planificación RR.
- 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.

Planificación de tiempo real

- Hard real-time:
  - El sistema garantiza la finalización de los procesos en un tiempo dado.
  - El proceso indica explícitamente los recursos necesarios.
  - El sistema acepta o no al proceso.
- Soft real-time:
  - Se da prioridad a procesos críticos.
  - Posible asignación desfavorable de recursos: esperas, inanición.
  - Sistemas multimedia, gráficos, etc.

System type	Hard or soft real time?
Traffic light control	Hard RT – Critical
Automated teller machine	Soft RT – Non-Critical
Controller for radiation therapy machine	Hard RT – Critical
Car simulator for driver training	Hard RT – Non Critical
Highway car counter	Soft RT – Non-Critical
Missile control	Hard RT – Critical
Video games	Hard RT – Non Critical
Network chat	Soft RT – Non-Critical

Evaluación de los planificadores (I)

- La selección del algoritmo se realiza según:
  - Máximo uso de la CPU.
  - Máximo número de salidas.
- Métodos:
  - Deterministas.
  - Modelos de colas.
  - Simulaciones.
  - Implementación.

Evaluación de los planificadores (II)

Deterministas.

- Forma analítica según carga y tipo de algoritmo.
- Cálculo del tiempo de salida y atención del proceso.
- Simple, rápido, fácil de comparar algoritmos.
- Aplicación a condiciones similares.

Modelos (modelos de colas).

- Distribución de las ráfagas de CPU y I/O (probabilidad).
- Aproximaciones al caso real.
- Cálculos de:
  - Tiempos medios de salida.
  - Tiempos medios de espera.
  - Grado de utilización.

- Ej:

  <n> procesos en cola.
  w tiempo de espera.
  λ ritmo de llegada de los procesos.
  $\lambda * \omega$ marca el número de procesos nuevos.
  El estado estacionario será para $\lambda * \omega = n$

Evaluación de los planificadores (III)

Simulaciones

- Simula reloj del sistema, IRQ, dispositivos, procesos, planificador, etc.
- Datos:

1. Generados:
   - Aleatorios (Procesos, ráfagas, tiempos de llegada,etc.).
2. Distribuciones existentes:
   - Sistemas reales bajo estudio.
   - Aproximada: solo nº de procesos por unidad de tiempo.
3. Distribuciones reales:
   - Captura de datos de sistemas reales.
   - Conjunto real e idéntico de procesos para someterlos a pruebas.

Implementación

- Mejor solución.
- Problema ➔ Costes

Planificación clásica en UNIX

- Emplea realimentación multinivel usando turno rotatorio en cada una de las colas de prioridad.
- La prioridad de cada proceso se calcula cada segundo.
- La prioridad base (**PBase**) divide los procesos en bandas fijas de prioridad.
  - Swap.
  - Control dispositivos I/O bloque.
  - Gestión archivos.
  - Control dispositivos I/O carácter.
  - Procesos usuario.
- Se utiliza un factor de ajuste para impedir que un proceso salga fuera de la banda que tiene asignada.