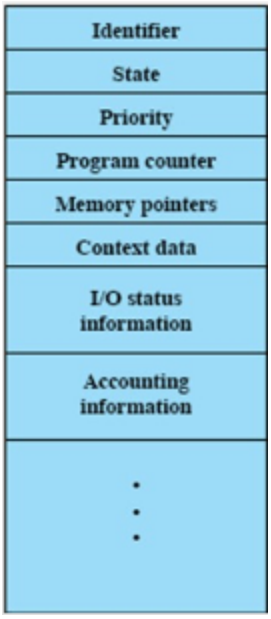
Practica 4

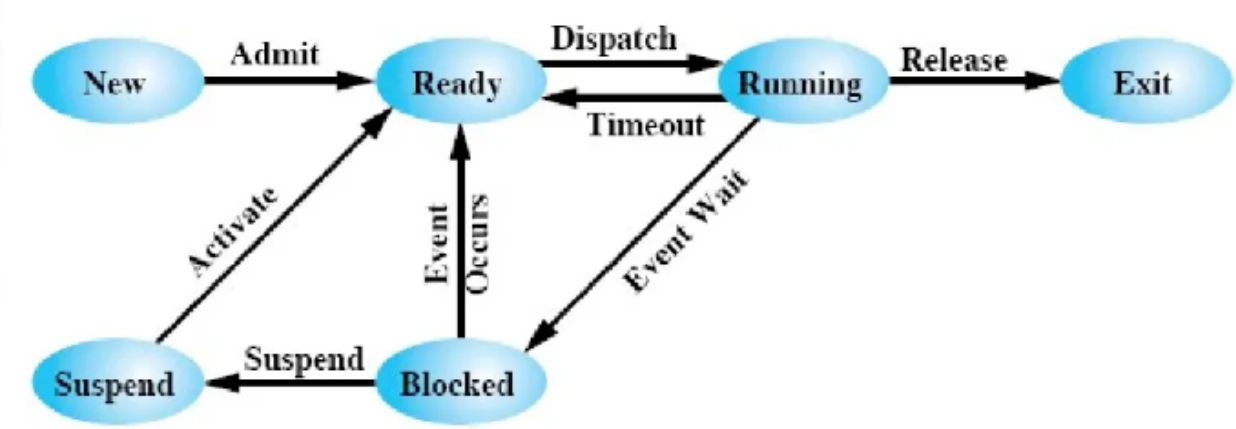
- Un **proceso** es un programa en ejecución. Los términos **tarea**, **job** y **proceso** son equivalentes. Según su uso de recursos, los procesos se clasifican en:
  - **CPU-Bound**: Requieren más tiempo de CPU.
  - **I/O-Bound**: Dependen más de operaciones de entrada/salida.

- Un programa es **estático**, no posee contador de programa y existe desde que se crea hasta que se elimina.
- Un proceso es **dinámico**, posee contador de programa y su vida inicia al ejecutarse y termina al finalizar.

El PCB es una estructura que:

- Existe una por cada proceso.
- Contiene toda la información del proceso (ID, estado, prioridad, contador de programa, punteros de memoria, entre otros).
- Es lo primero que se crea al hacer un `fork` y lo último que se libera al finalizar el proceso.





En resumen esta imagen muestra el diagrama de estados de un proceso en un sistema operativo. Los estados representan las diferentes fases por las que puede pasar un proceso mientras se ejecuta.

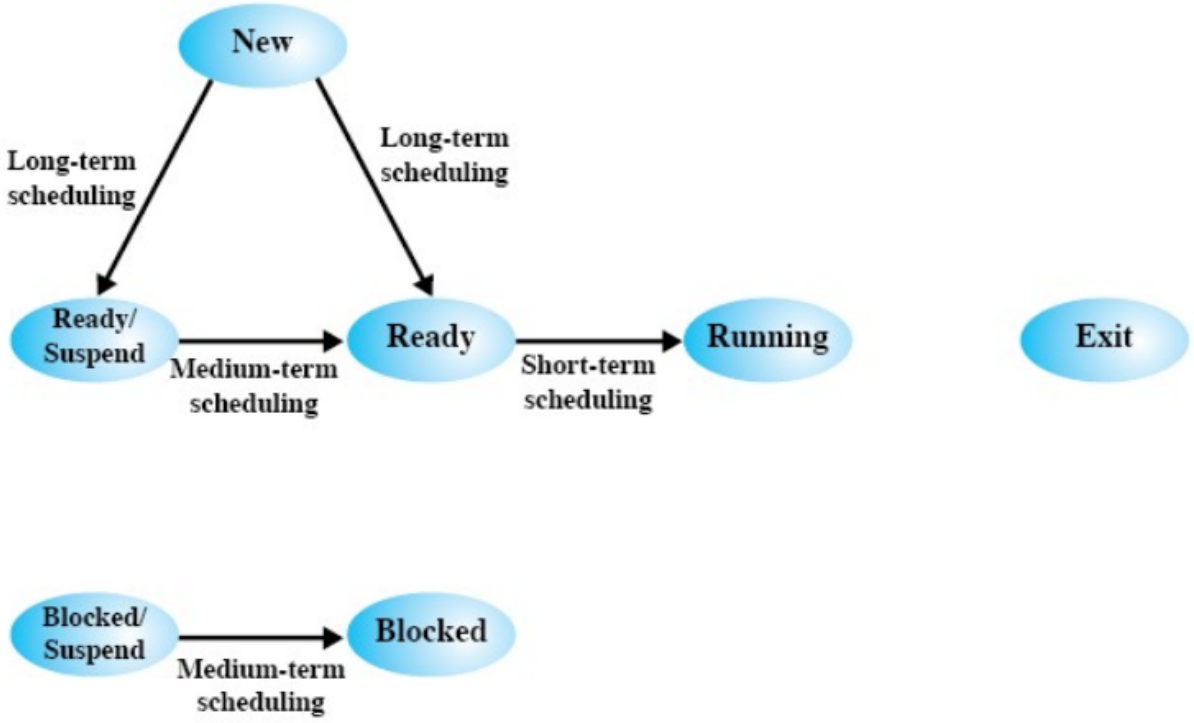
Planificadores

Son fundamentales para la multiprogramación y están diseñados para optimizar el uo del sistema con los siguientes objetivos:

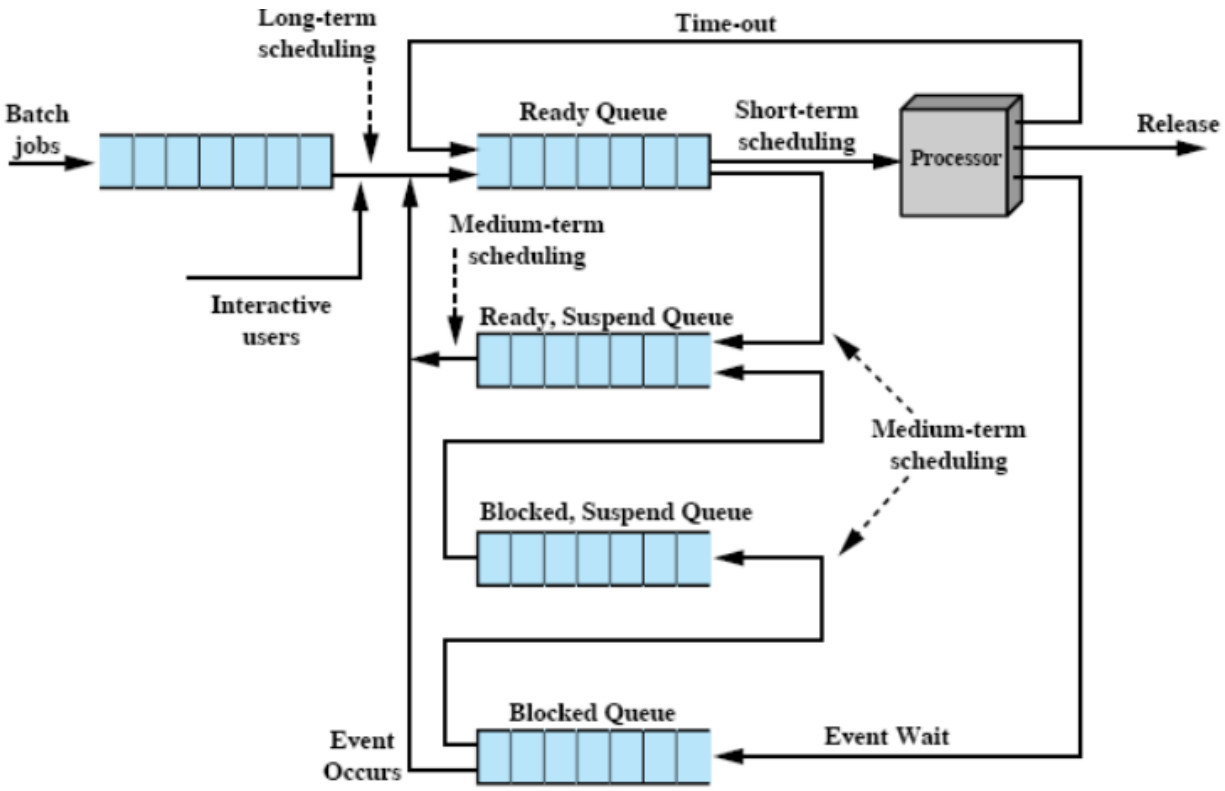
- Menor tiempo de respuesta
- Mayor rendimiento
- Uso eficiente del procesador

Tipos de planificadores

- Long-Term-Scheduler: Controla cuántos procesos se admiten en la memoria, regulando el grado de multiprogramación.
- Medium-Term-Scheduler: Realiza el intercambio entre disco y memoria (swapping) según lo determine el sistema operativo.
- Short-Term-Scheduler: Decide qué proceso pasará a ejecutarse en el procesador.



Planificadores y Colas



Esta imagen representa el modelo de colas en un sistema operativo multiprogramado para la gestión de procesos.

1- Batch jobs e Interactive users

- **Batch jobs:** Procesos que se ejecutan en segundo plano sin interacción directa con el usuario.
- **Interactive Users:** Procesos que requieren interacción directa con el usuario.

Ambos tipos de procesos llegan al sistema y son enviados a la Ready Queue mediante el Long Term Schedulling.

2- Ready Queue

- Posee los procesos que están listos para ejecutarse y esperan su turno en la CPU.

- Los procesos son seleccionados de esta cola por el Short-term schedulling, que asigna a la CPU a los procesos.

3- Processor

- Representa el procesador, que ejecuta los procesos seleccionados de la Ready Queue
- Una vez que un proceso termina su ejecución o es interrumpido, puede ser:
  - Liberado del sistema si ya se terminó.
  - Reinsertado en la Ready Queue

4- Suspensión y Bloqueo

- Si un proceso necesita esperar un evento, para a la Blocked Queue.
- Si el sistema está sobrecargado, los procesos de las colas Ready o Blocked pueden ser suspendidos y movidos a:
  - **Ready, Suspend Queue:** Procesos listos pero suspendidos temporalmente
  - **Blocked, Suspend Queue:** Procesos bloqueados y suspendidos

Esto es gestionado por el Medium-term schedulling, que equilibra el uso de los recursos del sistema.

5. Bloqueo y reanudación de eventos

- Los procesos en la **Blocked Queue** esperan un evento externo (como completar una operación de disco o de red).
- Cuando ocurre el evento, son movidos nuevamente a la **Ready Queue**.

Tiempos de los procesos

Los tiempos de los procesos son métricas clave para evaluar el rendimiento en un sistema operativo:

1. **Tiempo de retorno:** Es el tiempo total desde que el proceso llega al sistema hasta que finaliza su ejecución.
2. **Tiempo de espera:** Es el tiempo que el proceso pasa en el sistema sin ejecutarse (TR - Tcpu).
3. **Promedios:** Son los valores promedio de los tiempos de retorno y espera para todos los procesos, utilizados para medir la eficiencia del sistema.

Apropiación vs No Apropiación

No apropiativa (Nonpreemptive):

- El proceso en ejecución continúa hasta que finaliza o se bloquea.
- Tiene menos overhead (sobrecarga), pero puede causar que un proceso monopolice el procesador

Apropiativa (Preemptive):

- El proceso en ejecución puede ser interrumpido y enviado de vuelta a la cola de listos.
- Genera mayor overhead, pero mejora el servicio al evitar que un proceso monopolice el procesador.

Algoritmo FIFO

El algoritmo **FIFO (First Come, First Served)** selecciona el proceso más antiguo para ejecutarse. No prioriza ningún tipo de procesos, pero tiende a beneficiar a los **CPU Bound**, que completan su primera ráfaga, mientras que los **I/O Bound** suelen esperar más tiempo.

```
#Ejemplo 1
TAREA ''1'' PRIORIDAD=3 INICIO=0
[CPU,9]
```

```
TAREA ''2'' PRIORIDAD=2 INICIO=1
[CPU,5]
TAREA ''3'' PRIORIDAD=1 INICIO=2
[CPU,3]
TAREA ''4'' PRIORIDAD=2 INICIO=3
[CPU,7]
```

Job	Llegada	CPU	Prioridad
1	0	9	3
2	1	5	2
3	2	3	1
4	3	7	2

Algoritmo SJF

El algoritmo **SJF (Shortest Job First)** es una política no expropiativa que selecciona el proceso con la ráfaga más corta, basándose en ejecuciones previas. Favorece procesos cortos, pero los procesos largos pueden sufrir inanición (**starvation**).

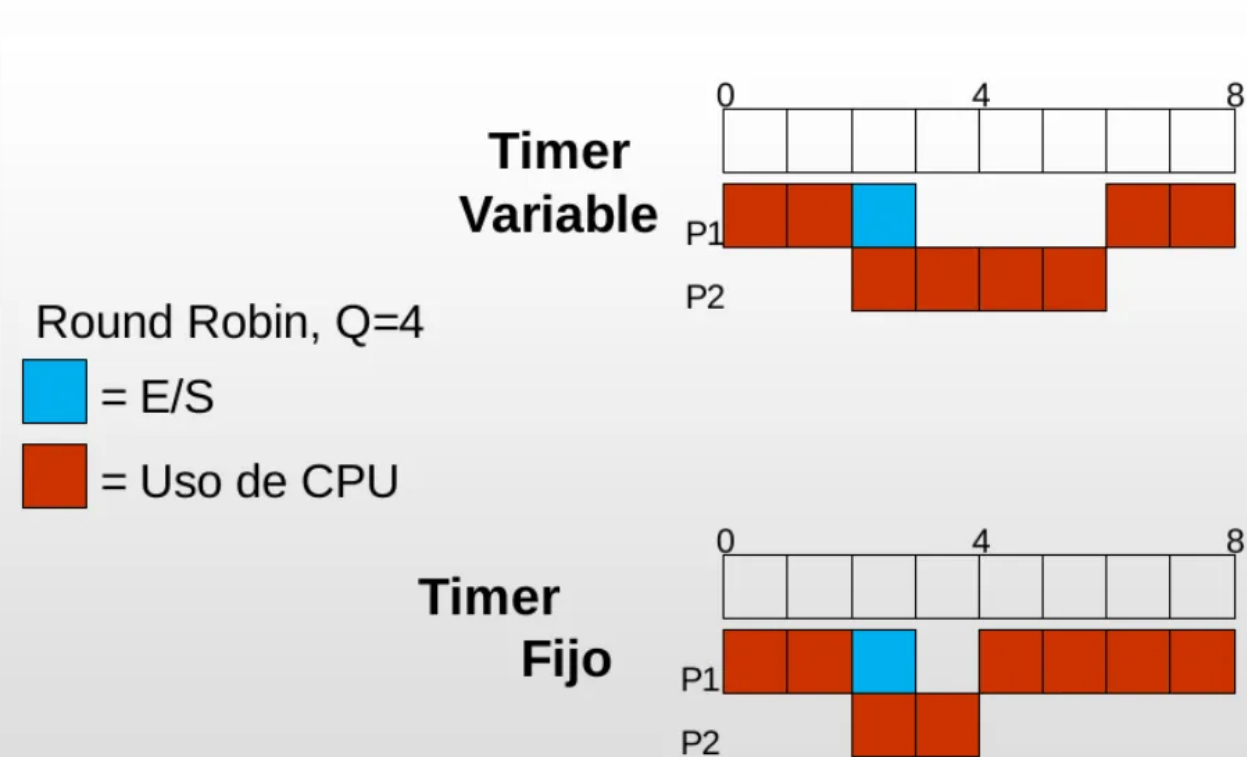
Algoritmo RR

Es una política basada en un reloj con quantum (Q) que determina cuánto tiempo puede usar el procesador cada proceso. Si un proceso agota su quantum sin terminar, es expulsado y colocado al final de la Ready Queue (FIFO circular)

- Quantum (Q):
  - **Pequeño:** Incrementa el overhead por cambios de contexto frecuentes.
  - **Grande:** Puede causar largas esperas para otros procesos.
- Contador:
  - Registra las unidades de CPU que el proceso ha usado
  - Puede ser:
    - Global: Está compartido entre procesos.
    - Local (En el PCB): Propio de cada proceso.

Variables del contador

- **Timer variable:**
  - Se inicializa en Q cada vez que un proceso es asignado a la CPU
  - Es más utilizada y común en simuladores
- **Timer Fijo:**
  - Se inicializa en Q solo cuando su valor llega a 0
  - Funciona como un Q compartido entre procesos\

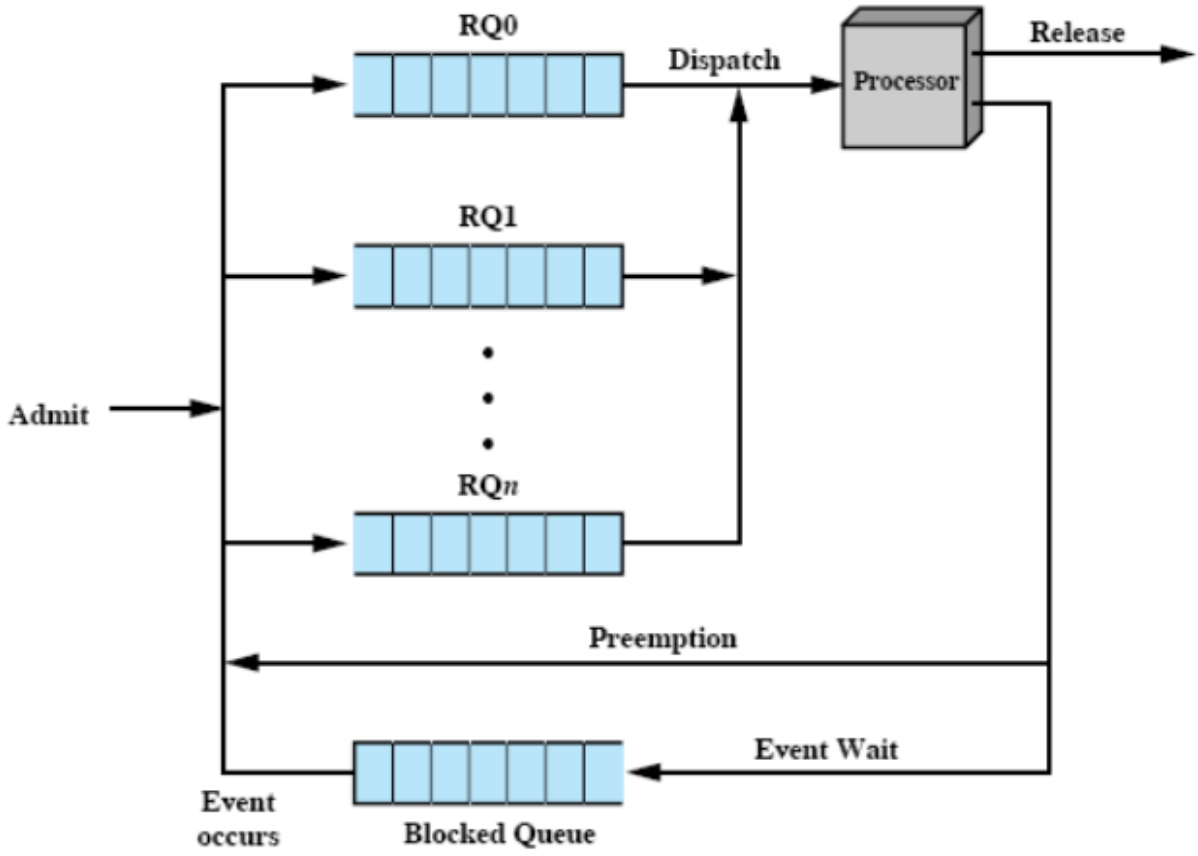


Algoritmo con Prioridades:

Cada proceso tiene un valor de prioridad (menor valor = mayor prioridad). Se selecciona el proceso de mayor prioridad de la **Ready Queue**, que puede estar dividida por niveles de prioridad.

- **Problema:** Procesos de baja prioridad pueden sufrir inanición (starvation).
- **Solución:** Implementar **aging**, que incrementa la prioridad de procesos que esperan mucho tiempo.

Puede ser **preemptive** (interrumpe procesos en ejecución) o **no preemptive** (deja que los procesos terminen antes de cambiar).



Algoritmo SRTF

Es la versión preemptive del algoritmo SJF. Selecciona y ejecuta el proceso que tiene menor tiempo restante para completar su ráfaga de CPU.

- **Ventaja:** Favorece a procesos I/O Bound, optimizando el tiempo de respuesta.

Algoritmos de planificación: CPU + I/O

- El ciclo de vida del proceso alterna entre uso de CPU y operaciones de I/O.

- Cada dispositivo tiene su propia cola de espera, un Scheduler independiente para gestionarla.
- Las operaciones de I/O se consideran independientes de la CPU, permitiendo que CPU e I/O trabajen simultáneamente.

Criterios de desempate

- **Orden de llegada:** Se priorizan los procesos según el momento en que llegaron.
- **PID:** En caso de empate se utiliza el identificador del proceso (PID).
- **Consistencia:** La misma política se aplica siempre para evitar conflictos.

Un recurso por proceso

```
#Ejemplo 2
RECURSO ''R1''
RECURSO ''R2''
RECURSO ''R3''
TAREA ''1'' INICIO=0
[CPU,3] [1,2] [CPU,2]
TAREA ''2'' INICIO=1
[CPU,2] [2,2] [CPU,2]
TAREA ''3'' INICIO=2
[CPU,2] [3,3] [CPU,1]
```

Job	Llegada	CPU	E/S (rec., inst., dur.)
1	0	5	(R1, 3, 2)
2	1	4	(R2, 2, 2)
3	2	3	(R3, 2, 3)

Recurso compartido

```
#Ejemplo 3
RECURSO ''R1''
RECURSO ''R2''
TAREA ''1'' INICIO=0
[CPU,3] [1,3] [CPU,2]
```

```
TAREA ''2'' INICIO=1
[CPU,1] [1,2] [CPU,3]
TAREA ''3'' INICIO=2
[CPU,2] [2,3] [CPU,1]
```

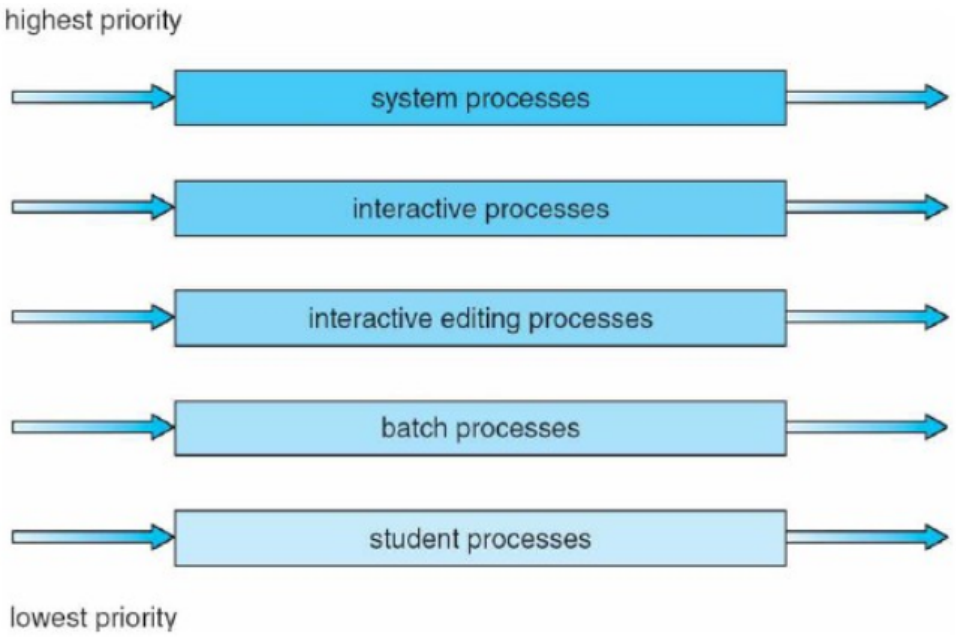
Job	Llegada	CPU	E/S (rec., inst., dur.)
1	0	5	(R1, 3, 3)
2	1	4	(R1, 1, 2)
3	2	3	(R2, 2, 3)

Esquema de Colas Multinivel:

En los **schedulers actuales**, se combinan varios algoritmos. La **Ready Queue** se divide en varias colas según la clasificación de los procesos realizada por el sistema operativo, similar a las prioridades.

- **Planificador horizontal:** Cada cola tiene su propio algoritmo de planificación.
- **Planificador vertical:** Planifica cuál cola se ejecutará a continuación.
- **Retroalimentación:** Los procesos pueden moverse entre colas, adaptándose a su comportamiento.

Ejemplo 1



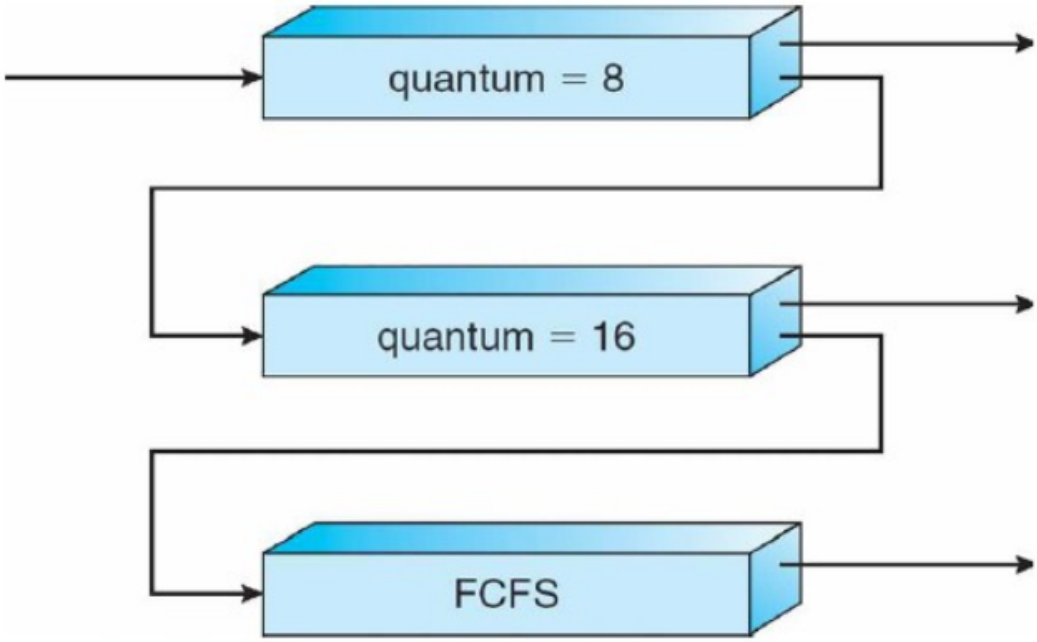
Ejemplo 2

El sistema consta de 3 colas:

- **Q0:** Se planifica con **Round Robin (RR)**, **quantum = 8**.
- **Q1:** Se planifica con **Round Robin (RR)**, **quantum = 16**.
- **Q2:** Se planifica con **FCFS** (First Come First Serve).

Criterios de Planificación:

- Los procesos comienzan en **Q0**. Si no completan su quantum de 8, se mueven a **Q1**.
- En **Q1**, si el proceso no completa su quantum de 16, se mueve a **Q2**.
- **Q2** utiliza **FCFS**, sin preempción.



Planificación con Múltiples Procesadores:

La planificación de CPU es más compleja con **múltiples CPUs**, ya que la carga se distribuye entre ellas, lo que mejora las capacidades de procesamiento. Si un procesador falla, otro toma el control.

Criterios:

- **Planificación temporal:** Determina qué proceso se ejecuta y por cuánto tiempo.
- **Planificación espacial:** Decide en qué procesador se ejecuta el proceso, teniendo en cuenta:
  - **Huella:** El estado que el proceso deja en la caché del procesador.
  - **Afinidad:** Preferencia de un proceso por ejecutar en un procesador específico.

Asignación de Procesos:

- **Estática:** Un proceso tiene afinidad con un procesador específico.
- **Dinámica:** La carga se reparte entre los procesadores (balanceo de carga).

Política:

- **Tiempo compartido:** Puede haber una cola global o local a cada procesador.
- **Espacio compartido:** Se pueden usar grupos (threads) o particiones.

Clasificaciones:

1. **Procesadores homogéneos:** Todas las CPUs son iguales, sin ventajas físicas.
2. **Procesadores heterogéneos:** Cada procesador tiene su propia cola, reloj y algoritmo de planificación.
3. **Procesadores débilmente acoplados:** Cada CPU tiene su propia memoria y canales.
4. **Procesadores fuertemente acoplados:** Comparten memoria y canales.
5. **Procesadores especializados:** Combinan procesadores principales de uso general con otros especializados.