PLANIFICADOR DE PROCESOS

Karim Guevara Puente de la Vega

2018

Objetivos de la Planificación

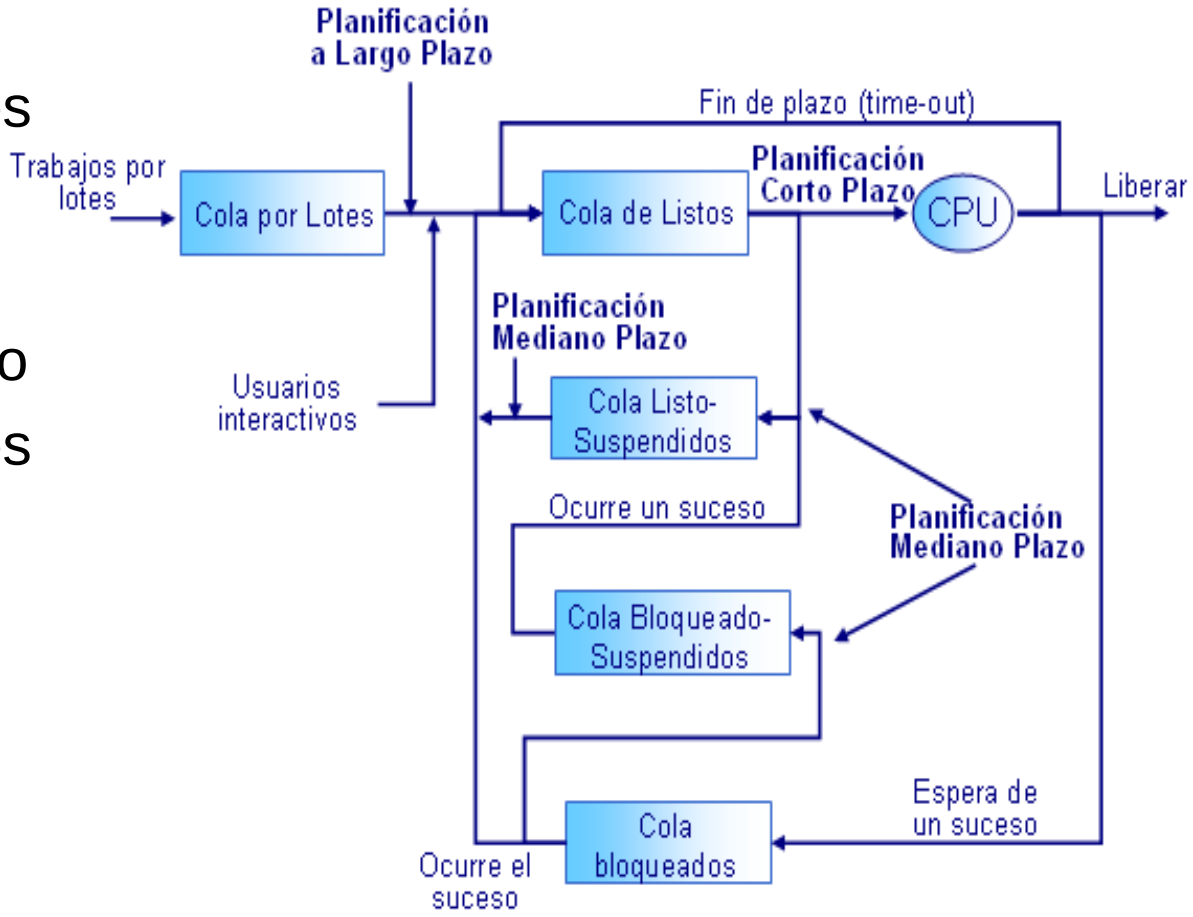
Los procesos obtengan sus turnos de ejecución apropiadamente, conjuntamente con un buen rendimiento y minimización de la sobrecarga (overhead) del planificador mismo.

Criterios:

- **Justicia o Imparcialidad**:
  - Todos los procesos son tratados de la misma forma, y en algún momento obtienen su turno de ejecución o intervalos de tiempo de ejecución hasta su terminación exitosa.
- **Maximizar la Producción**
  - El sistema debe de finalizar el mayor número de procesos por unidad de tiempo.
- **Maximizar el Tiempo de Respuesta**
  - Cada usuario/proceso debe observar que el sistema les responde consistentemente a sus requerimientos.
- **Evitar el aplazamiento indefinido**
  - Los procesos deben terminar en un plazo finito de tiempo.
- **El sistema debe ser predecible**
  - Ante cargas de trabajo ligeras el sistema debe responder rápido y con cargas pesadas debe ir degradándose paulatinamente.

Planificadores

- Planificador a largo plazo o planificadores de trabajo
- Planificadores a corto plazo o planificadores de CPU
- Planificadores a mediano plazo o swapping



Planificadores

- Largo plazo:
  - Sucede cuando se crea un proceso (API)
  - Está en función de los recursos o políticas del sistema.
  - Determina el grado de multiprogramación
  - Se ejecuta en intervalos de tiempo grandes (varios minutos)
- Corto plazo:
  - Sucede cuando un proceso sale del estado en ejecución
  - Se ejecuta con mucha frecuencia (cada 10 mseg.)
  - Debe ser lo más rápido (overhead)
- Mediano plazo:
  - Sucede cuando no hay disponibilidad de memoria principal o para mejorar el rendimiento.
  - Se ejecuta cada pocos segundos o minutos.

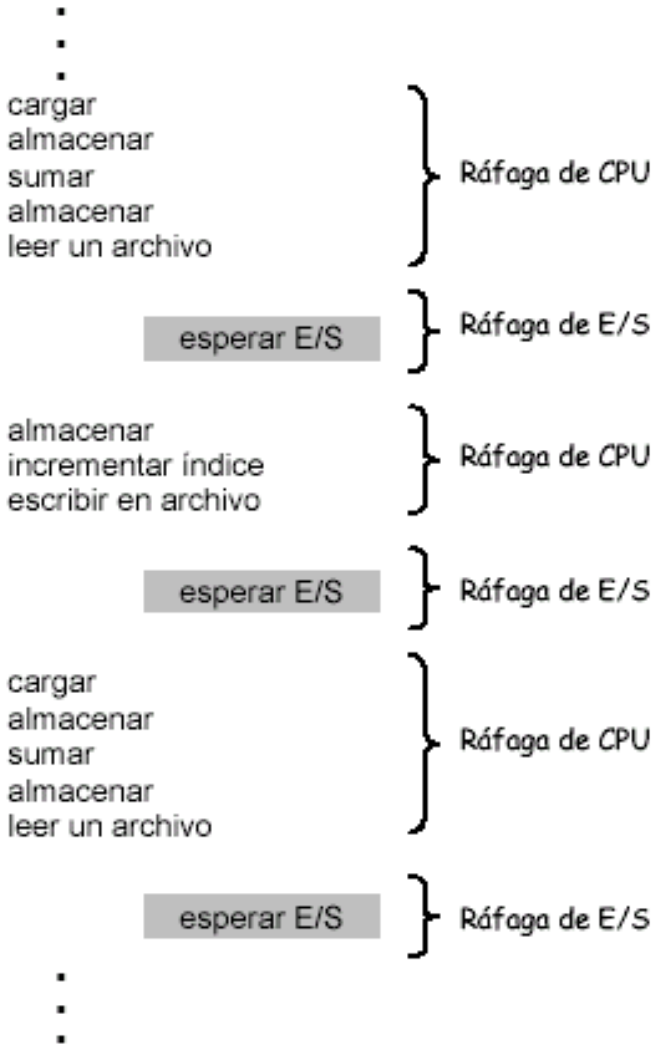
Algoritmos de planificación a corto plazo

- El algoritmo de planificación es el encargado de elegir, en un momento dado, el proceso que va ha ser asignado al CPU de entre los que están preparados para ejecución.

- El algoritmo de planificación de los diferentes sistemas operativos, utilizan diversas políticas de planificación, en función del **tipo de proceso**.

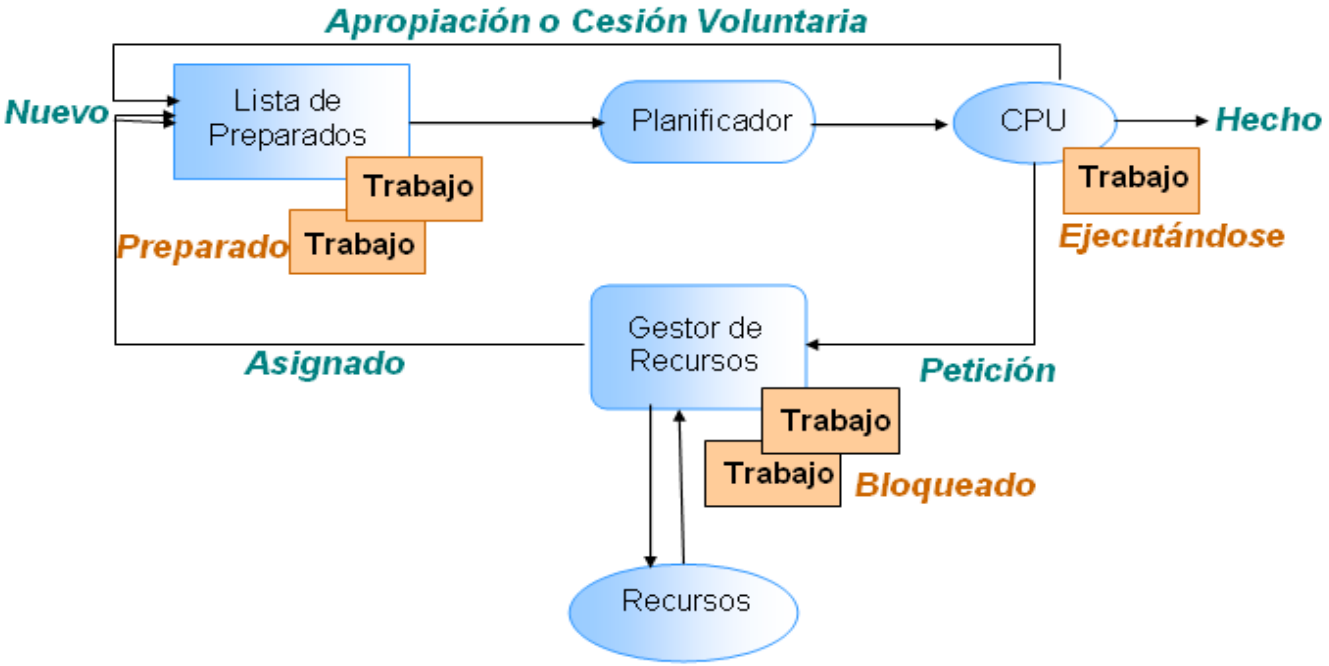
Tipos de procesos

- Por tipo de transacciones
  - Tiempo real
  - Segundo plano: (procesos de cálculo servidores)
  - Interactivos
- Por tipo de operaciones
  - Procesos con uso intensivo (limitado) de CPU: gran cantidad de complejos cálculos matemáticos (CPU-bound)
  - Procesos con uso intensivo (limitado) de E/S: mucha interacción con dispositivos de E/S (I/O-bound)



Planificación de la CPU (corto plazo)

- Razones para que el hilo/proceso deje de utilizar la CPU:
  - Completa su función (**en ejecución → terminado**)
  - Solicita un recurso, abandona voluntariamente la CPU (**en ejecución→bloqueado**)
  - Abandona involuntariamente la CPU (**en ejecución→listo**)
  - Proceso termina su E/S (**bloqueado→listo**)

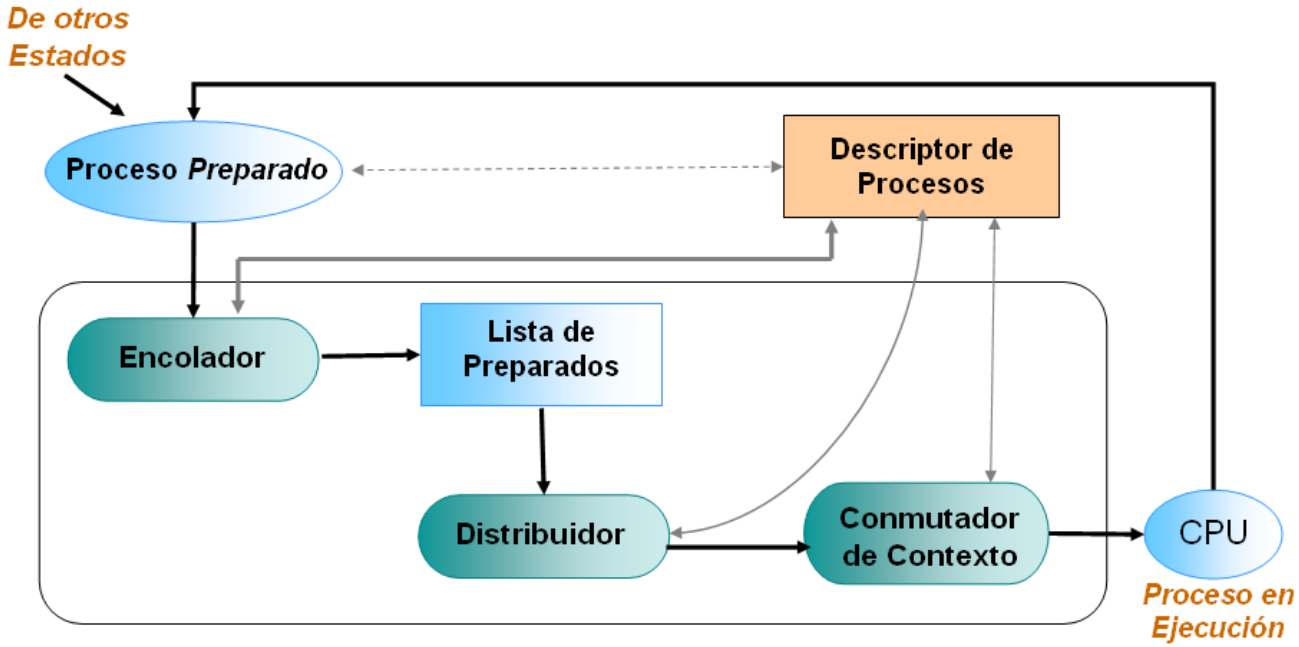


Planificación de la CPU

- Planificador del SO:
  - Un **MECANISMO** para la conmutación de contexto
    - Cómo se asignará la CPU
    - Cómo se desalojará al hilo
  - Una **POLITICA** determina el orden de servicio.
    - Cuándo es el momento de desalojar el hilo
    - Cuál es el hilo elegible

Mecanismo de Planificación

- Depende de las características del Hw:
  - Tiene temporizador?... Lo demás está implementado por medio de Sw del SO
  - Organización del planificador: **Encolador** (*enqueuer*), **Conmutador de contexto** (*context switcher*) y **Distribuidor** (*dispatcher*)



Política de Planificación

- **No preemtive**: *(no expropiativo / no apropiativo)*
  - Proceso asignado a la CPU, se mantiene en ejecución hasta que termine o hasta que emite una solicitud de E/S, no puede ser suspendido.
  - Puede ser peligroso
  - **Efecto Convoy:**
    - Un trabajo muy grande aplaza mucho a uno pequeño
    - Un proceso de alta prioridad debe esperar a que termine el proceso actual en ejecución
- **Preemptive**: (*expropiativo / apropiativo*)
  - Da un *tiempo limite* (quantum, existe un reloj/temporizador)
  - El proceso es suspendido al termino del *tiempo límite*

Algoritmos de Planificación

- Criterios útiles para planificar procesos
  - Tiempo ocioso o en espera
  - Previsibilidad, plazos o tiempos de respuesta requerido
  - Equidad, prioridades y privilegios
  - Trabajo pendiente o tareas por realizar
  - Exigencias de E/S
  - Productividad, utilización y carga del CPU y E/S

Algoritmos de Planificación

- FIFO / FCFS.
- Primero el más corto (SJF, Shortest job-first). SPN(Shortest-Process-Next)
- Prioridad.
- Por Tiempo Límite
- Round-Robin.
- Colas de múltiples niveles.
- Colas de múltiples niveles con retroalimentación

FIFO o FCFS

- Algoritmo no expropiativo:
  - Sólo se activa cuando proceso en ejecución se bloquea o termina.
- Fácil de implementar:
  - Cola de listos gestionada en modo FIFO.
- No válido para procesos interactivos.
- Mal tiempo medio de espera en cola de listos:
  - Proceso con ráfagas de CPU más pequeñas pueden tener que esperar.
  - Efector convoy
  - Mejor realizar antes “servicios” más cortos:
    - Algoritmo “primero el más corto” (SJF, Shortest Job First).

FIFO o FCFS

Proceso	T. Servicio
0	350
1	125
2	475
3	250
4	75

SJF (Shortest Job First)

- Asocia a cada proceso la longitud de la última ráfaga de CPU.
  - La CPU es asignada al proceso que tiene la próxima ráfaga de CPU más corta.
- Si se da el caso que dos procesos tienen la misma longitud de ráfaga → FCFS.
- Da el mínimo tiempo de espera promedio para un conjunto de procesos.
- Su dificultad es conocer la longitud de la próxima ráfaga de CPU de un proceso.
- Puede producir **inanición** (los procesos largos pueden esperar si existe flujo continuo de procesos cortos)

SJF

- Un enfoque es tratar de aproximar la planificación mediante el cálculo de una aproximación de la longitud de la próxima ráfaga de CPU.
- Puede ser apropiativo o no apropiativo
  - La planificación Preemptive suele llamarse “primero el de tiempo restante más corto” SRT(Shortest-Remaining-Time)

Proceso	T. Servicio
0	350
1	125
2	475
3	250
4	75

Por prioridades

- El proceso de mayor prioridad se ejecuta primero
- Procesos de igual prioridad se usa FCFS
- Casos especiales:
  - SJF → prioridad es la inversa del tiempo requerido de CPU.
  - FCFS → prioridad es entregado por orden de llegada de los procesos
- Puede ser: (si prioridad del proceso recién llegado es mayor al proceso en ejecución)
  - **No Apropiativo:** se coloca al recién llegado en la cabeza de la cola Listo.
  - **Apropiativo:** se le expropia la CPU al que esta en ejecución en favor del recién llegado.

Criterios para definir las prioridades

- Internos/Externos:
  - Rango o categoría del usuario (multiusuario).
  - Tipo de proceso: sistema, interactivo, o por lotes.
  - Factores políticos
  - Razones físicas (sistemas de tiempo real)
  - Cuánto hayan ocupado la CPU
- Estáticas/Dinámicas:
  - Asignación de prioridades a priori. (P.e.: costos de procesos)
  - A fin de lograr ciertos objetivos del sistema, (P.e.: procesos limitados a E/S, se les otorga de inmediato la CPU).
    - Quantum / f → f fracción del último quantum que utilizó un proceso .

Por prioridades

Proceso	T. Servicio	Prioridad
0	350	5
1	125	2
2	475	3
3	250	1
4	75	4

Por Prioridades

- Inanición o Hambruna
  - Dejar a un proceso de baja prioridad esperando indefinidamente por el procesador → hay algún proceso de mayor prioridad.

  - Solución → Envejecimiento
    - La prioridad de los procesos listos se incrementa gradualmente (reloj).
    - La prioridad de los procesos que se ejecutan se va disminuyendo.

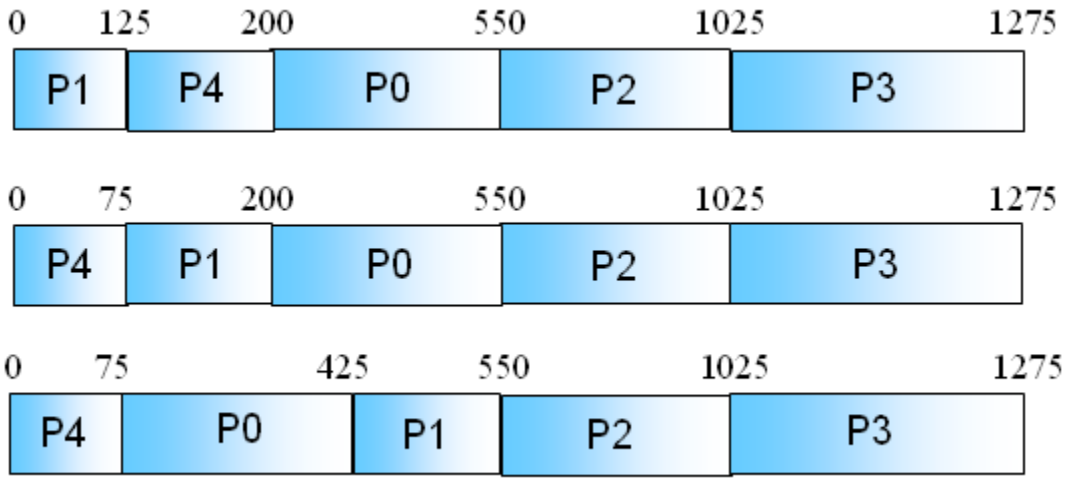
Por tiempo límite

- Sistemas de tiempo real (estricto)
  - ¿El sistema será capaz de cumplir con la planificación de los tiempos límites de este tipo de procesos/hilo?
  - Precisan conocer el máximo tiempo de servicio de cada proceso, para cada periodo de vida
  - Debe ser admitido en la lista de preparados si el planificador garantiza que es capaz de proveerle el tiempo necesario antes del tiempo límite impuesto

Por tiempo límite

Proceso	T. Servicio	T. Limite
0	350	575
1	125	550
2	475	1050
3	250	(none)
4	75	200

Plan. Primero el tiempo más cercano



Algortimo Round Robin - RR

- Algoritmo FIFO + quantum de tiempo.
  - Utiliza un reloj/temporizador/timer
- Cuando se asigna CPU a proceso se le da un quantum de tiempo.
  - Si lo gasta (rafaga>quantum) es expulsado al final de cola de listos.
- Algoritmo expulsivo pero sólo con fin del quantum
- Tiempo de respuesta acotado:
  - Si quantum igual a C y N procesos listos:
    - Proceso no esperará más de (N-1)*C unidades de tiempo.
- Útil en sistemas multiusuario de tiempo compartido

Algoritmo Round Robin - RR

- Performance del sistema depende del tamaño del quantum:
  - Grande: tiende a FIFO. Mal tiempo de respuesta.
    - Degenera en FCFS:
    - “Que el sistema tarda mucho en atenderlos”
  - Pequeño: Usuarios son atendidos de inmediato, por poco tiempo
    - Demasiada sobrecarga (overhead)
    - Degrada el aprovechamiento de la CPU
  - Debe ser grande con respecto al tiempo de cambio de contexto.
    - Típico: 10-100 milisegundos

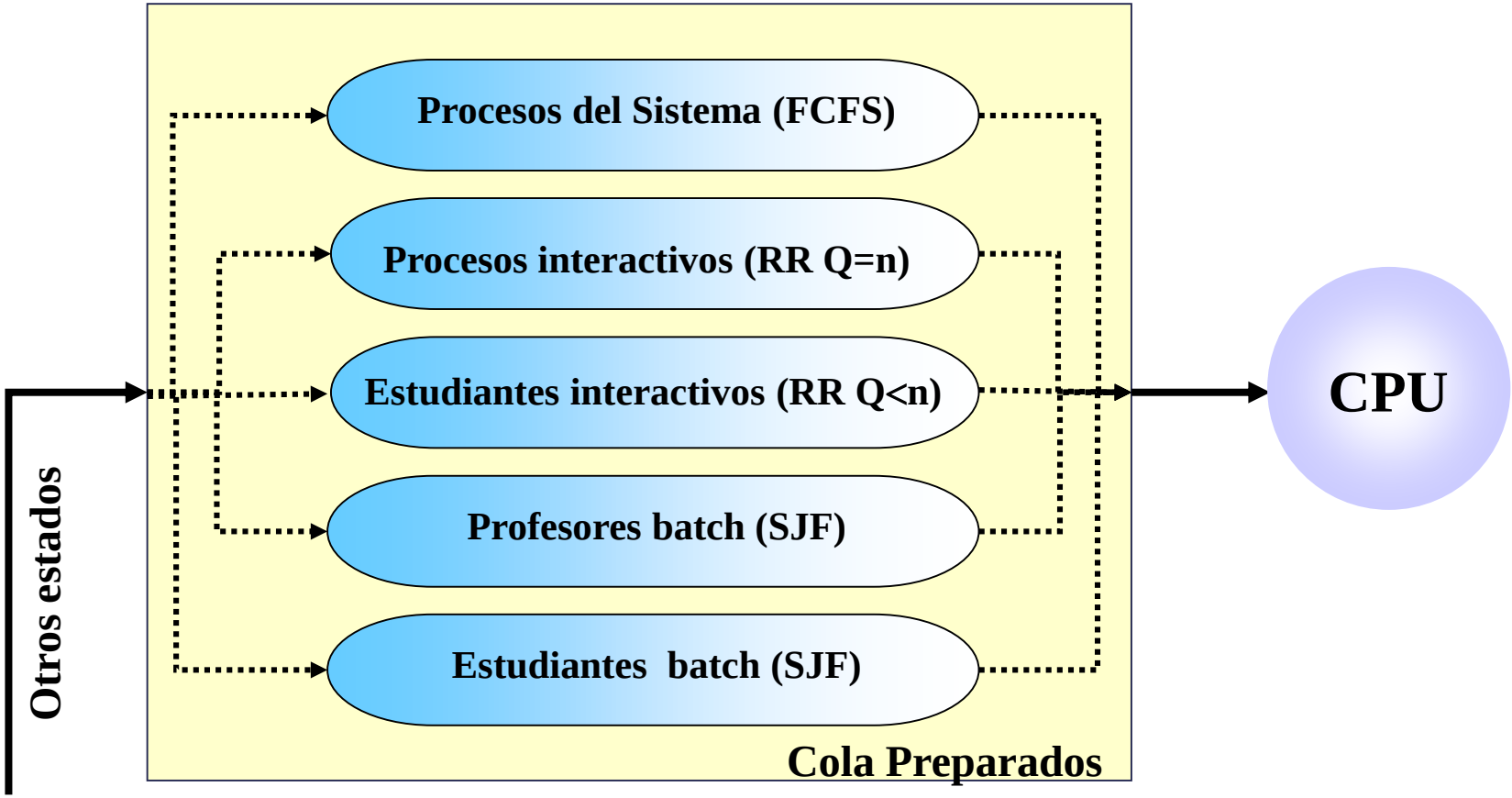
Round-Robin (RR)

- \+ Con respecto a FCFS: mejora tiempo de respuesta
- \+ Colas ***Listo y Bloquedado*** se mantienen balanceadas
- – No asegura que los tiempo de espera sean los mínimos posibles

Proceso	T. Servicio
0	350
1	125
2	475
3	250
4	75

Quantum=50

Colas de múltiples niveles



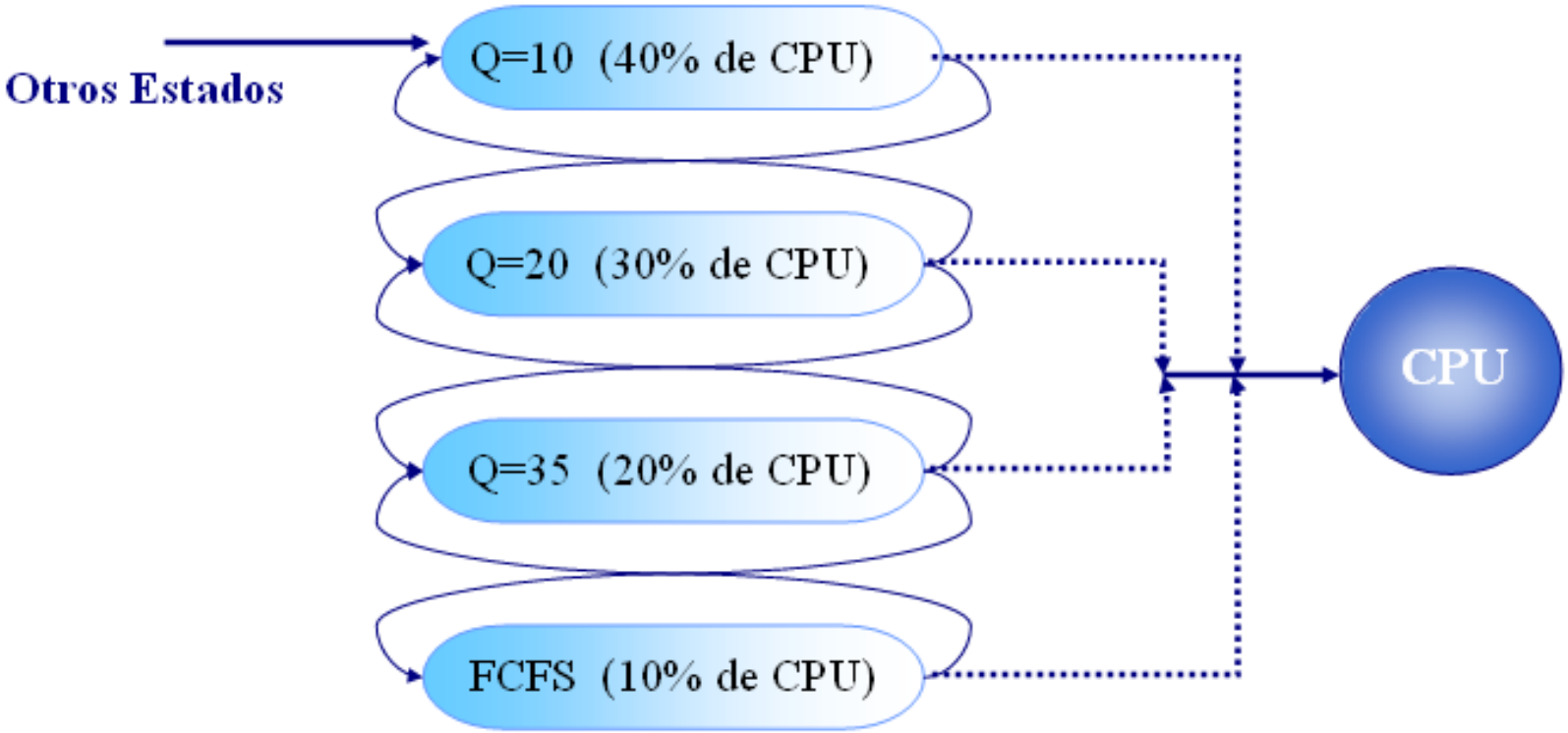
Colas de múltiples niveles

- Modelo más general:
  - Cola de listos organizada en múltiples niveles.
    - P.e. dependiendo de la interactividad de los procesos, tipo de proceso
  - Cada nivel puede tener su propio algoritmo:
    - P.e. RR para niveles más interactivos y FIFO para los menos.
  - Existe un algoritmo para seleccionar entre niveles:
    - P.e. por prioridades, mayores prioridades los niveles más interactivos.
  - Dos algoritmos:
    - Uno para elegir la cola de la cual se sacará el proceso y otro para elegir el proceso de la cola.

Colas de múltiples niveles

- Puede ocurrir que en dos colas distintas se utilice la misma política de planificación:
  - RR, pero con quantum diferentes.
- Ordenar las colas por “prioridad decreciente”:
  - Cola de alta prioridad se vacía antes de moverse a la siguiente cola.
  - Distribuir tajadas de tiempo de CPU en las diferentes colas:
    - Cola del sistema → 60% : RR
    - Procesos por lotes → 5% : FCFS
    - Otras : resto.

Colas de múlt. niveles retroalimentadas - MLF



Colas de múlt. niveles retroalimentadas - MLF

- Realimentación:
  - Posibilidad de que los procesos cambien de nivel.
- Deben existir criterios de cambio de nivel:
  - P.e. Mover de nivel un proceso si tiende a aumentar o disminuir su interactividad.
- Un proceso se puede mover de una cola a otra, si:
  - lleva mucho tiempo en cola de poca prioridad, se le pasa a otra de mayor prioridad (envejecimiento)
  - usa demasiado tiempo de CPU se pasa a una cola de menor prioridad.
- Es el algoritmo de planificación más general y el más complejo.

Planificación en multiprocesadores

- Sistema Homogéneo:
  - Cualquier procesador disponible → ejecuta un proceso listo.
- Sistema Heterogéneo:
  - Programas compilados para un procesador, se ejecutan en ese procesador.
  - Cada procesador tiene su propia cola Listo, y su propia planificación.

Planificación en multiprocesadores

- Sistema Homogéneo:
  - Planificación por si mismo (autoploanificación)
    - Cada procesador es responsable de su planificación y de sacar de la cola LISTO.
    - Requiere sincronización entre procesadores
      - Cola Listo → cuello de botella.

Planificación en multiprocesadores

- Sistema Homogéneo:
  - Maestro/Esclavo
    - Uno de los procesadores planifica.
    - Ejecuta el SO
    - Despacha los trabajos a los esclavos.
    - Hace todo el procesamiento de las interrupciones.
    - Esclavo sólo ejecuta los programas.
    - Puede ser el cuello de botella.

Evaluación de los Algoritmos

- El primer problema es definir el criterio a ser usado en la selección .
- El criterio es definido frecuentemente en términos de:
  - Utilización de la CPU.
  - Tiempo de respuesta.
  - Rendimiento.
- Para seleccionar un algoritmo se debe primero definir la importancia relativa de estas mediciones.
- El criterio elegido puede incluir varias mediciones como:
  - Maximizar la utilización de la CPU bajo la restricción: el máximo tiempo de respuesta es 1 segundo

Criterios de planificación

- Diferentes criterios:
  - Maximizar grado de utilización de UCP.
  - Maximizar rendimiento: nº de procesos terminados/u. de tiempo
  - Minimizar tiempo de espera de cada proceso:
    - tiempo gastado esperando en cola de listos.
  - Minimizar tiempo de respuesta de procesos interactivos:
    - tiempo desde que usuario realiza petición hasta que el proceso empieza a responder.
  - Minimizar tiempo de retorno:
    - tiempo desde que se empieza a ejecutar la aplicación hasta que termina totalmente.
- Tendencia general: Favorecer trabajos más interactivos.

Evaluación de los algoritmos

- Modelo determinista
  - Considerar una carga de trabajo determinada
  - Necesita datos precisos como entrada. Las respuestas sólo son válidas para esos casos
  - Define el rendimiento de cada algoritmo
  - Muy sencillo
  - Válido cuando siempre se están ejecutando los mismos programas y en el mismo orden

Evaluación de los algoritmos

- Simulaciones
  - Método más preciso para medir algoritmos de planificación
    - Construir un simulador
    - Generara en forma aleatoria procesos, tiempos de ráfagas, tiempos de llegada, etc.
  - Problema: el costo

- Implantaciones
  - Implantar el algoritmo en el SO y ver su funcionamiento
    - Más exacto
    - Caro
    - No adecuado para sistemas reales (experimentando)

Evaluación de los algoritmos

- Modelodeterminista

Proceso	T. Servicio
1	10
2	29
3	3
4	7
5	12

Minimizar T_{espera}?

- FCFS
- SJF
- RR (Q=10)