

Fundamentos de los Sistemas Operativos

Tema 2. Procesos Planificación de CPU

© 1998-2015 José Miguel Santos - Alexis Quesada - Francisco Santana

Contenido

- Modelo del sistema y criterios de rendimiento
- Algoritmo FCFS
- Algoritmo SJF
- Turno rotatorio (*Round-Robin*)
- Métodos basados en prioridades
- Métodos multicolos
- Planificación en multiprocesadores

Planificación del procesador (*processor scheduling*)

- El sistema operativo decide qué proceso ocupa el procesador cuando éste queda libre.
- También en qué momento el proceso que está en ejecución debe abandonar el procesador.

Política de planificación

- ¿Cuál es la «mejor» manera de decidir qué proceso entra en la CPU?
 - ¿En orden de llegada, primero la tarea más corta, por prioridades...?
- Debemos definir posibles criterios de valoración de las políticas

Criterios de valoración

- Podemos usar algunas magnitudes para medir el *rendimiento* de los algoritmos:
 - **Utilización de CPU:** % de tiempo que la CPU está ocupada.
 - **Tiempo de retorno:** tiempo transcurrido entre la llegada de un proceso y su finalización.
 - **Tiempo de espera:** tiempo que un proceso permanece en la cola de preparados.
 - **Tiempo de respuesta:** tiempo que un proceso bloqueado tarda en entrar en CPU, desde que ocurre el evento por el que estaba esperando.

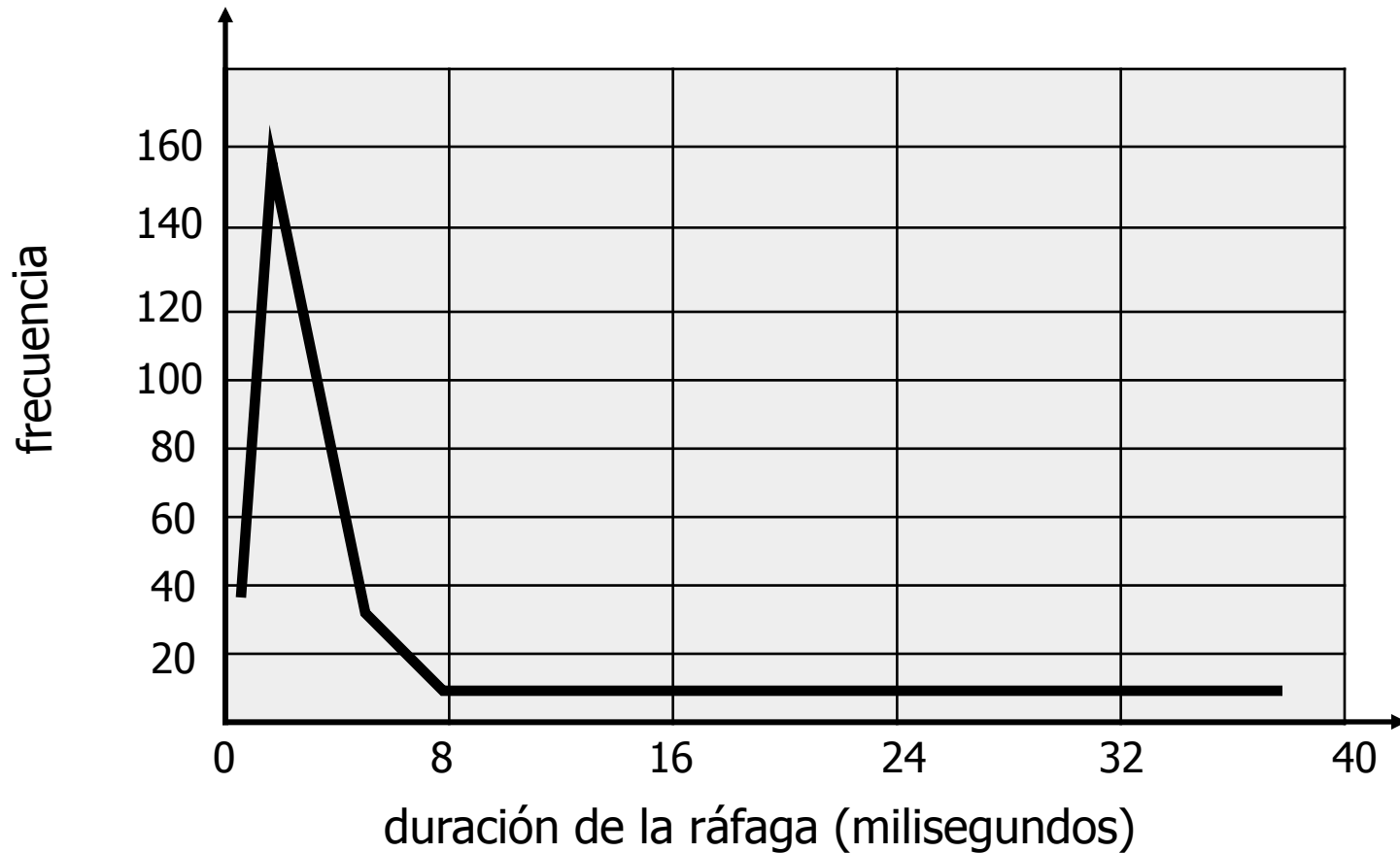
Criterios de valoración

- Posibles objetivos de la planificación:
 - Minimizar el tiempo medio de espera o de retorno
 - Maximizar la utilización de CPU
 - Mantener el tiempo de respuesta por debajo de un valor máximo, etc.
- Se pueden considerar las medias, valores extremos o varianzas de estas magnitudes.
- No existe una política de planificación óptima para todos los criterios.
 - Habrá que llegar a un compromiso.

Modelo del sistema: ráfagas de CPU y E/S

- Podemos considerar que la vida activa de un proceso es una sucesión de:
 - ráfagas de CPU → el proceso ejecuta instrucciones
 - ráfagas de E/S → el proceso utiliza o espera por la E/S
- Según la utilización de los recursos, se observan:
 - procesos **intensivos en CPU** (ej. cálculos numéricos)
 - procesos **intensivos en E/S** (ej. interactivos)

Distribución típica de duraciones de las ráfagas de CPU



Algoritmos expulsivos (*preemptive*)

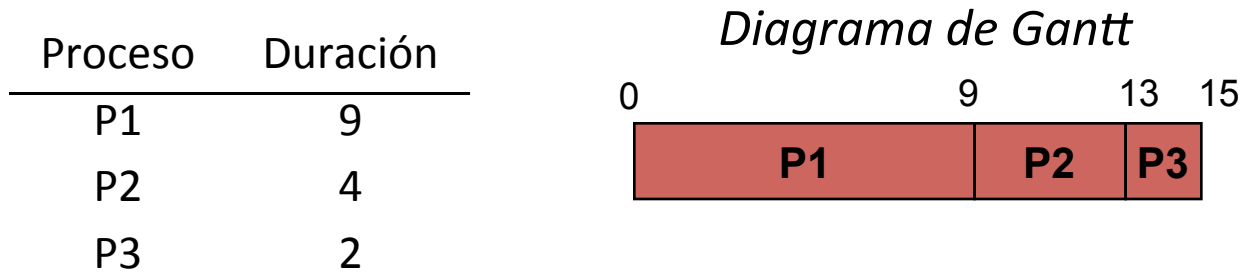
- No expulsivos: el proceso que está en CPU la abandona cuando lo solicita (ej. FCFS)
 - riesgo de acaparamiento injusto de la CPU
 - Windows 3.11, Apple Macintosh original...
- **Expulsivos:** el planificador puede desalojar al proceso que está en CPU
 - para implementar tiempo compartido y tiempo real, es necesaria una planificación expulsiva: Unix, Windows NT, Mac OS X...

FCFS (en orden de llegada)

Proceso	Duración
P1	9
P2	4
P3	2

- Calcular el tiempo de espera, tiempo de retorno y tiempo medio de espera si aplicamos el algoritmo FCFS suponiendo que llegan en el mismo instante en el siguiente orden: P1, P2, P3
- Realizar los mismos cálculos suponiendo que llegan en el siguiente orden: P2, P3 y P1

FCFS: ejemplo de diagrama de Gantt

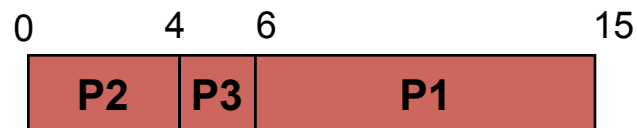


Tiempos de espera: P1=0; P2=9; P3=13

Tiempos de retorno: P1=9; P2=13; P3=15

t. espera medio: $(0+9+13)/3 = 7,3$

Si P1 hubiera llegado el último, los tiempos hubieran mejorado bastante (espera media=3,3):



FCFS: características

- La cola de preparados se gestiona como una FIFO
- Simple de implementar
- Muy sensible al orden de llegada de los procesos
- Perjudica a los procesos intensivos en E/S (*efecto convoy*)

SJF (primero el más corto)

- SJF = *Shortest Job First*
- Entra en CPU el proceso con la ráfaga de CPU más breve.
- Versión expulsiva: *Shortest Remaining Time First* (**SRTF**)
 - El proceso en CPU es desalojado si llega a la cola un proceso con duración más corta.

SJF - ejemplo

Proceso	Llegada	Duración
P1	0	7
P2	2	4
P3	4	1
P4	5	4

- Calcular el tiempo medio de espera que resulta de aplicar un algoritmo SJF no expulsivo
- Calcular el tiempo medio de espera que resulta de aplicar un algoritmo SJF expulsivo (SRTF)

SJF - ejemplo

Proceso	Llegada	Duración	espera SJF	espera SRTF
P1	0	7	0	9
P2	2	4	6	1
P3	4	1	3	0
P4	5	4	7	2

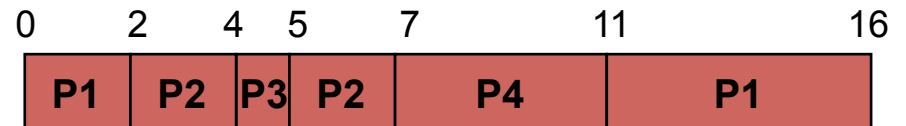
SJF no expulsivo

espera media: $(0+6+3+7)/4=4$



SJF expulsivo (SRTF)

espera media: $(9+1+0+2)/4=3$



SJF: características

- Minimiza el tiempo de espera medio (óptimo).
- Riesgo de **inanición** de los procesos con ráfagas de larga duración.
- No se puede predecir la siguiente ráfaga → una implementación real debe calcular una estimación.
- Posible estimación: media exponencial de las ráfagas anteriores

$$\tau_{n+1} = \alpha t_n + (1 - \alpha) \tau_n.$$

Planificación basada en prioridades

- Cada proceso tiene una prioridad asignada. Entra en CPU aquel con mayor prioridad.
 - la política puede ser expulsiva o no
 - Prioridades definidas de forma *interna* (por el S.O.) o *externa* (por los usuarios)
 - El SJF es un ejemplo (prioridad=duración estimada)
- Riesgo de **inanición** de los procesos con menos prioridad.
- Solución: **envejecimiento (*aging*)**. Aumentar progresivamente la prioridad a los procesos en espera.

Turno rotatorio (*Round-Robin*)

- Adecuado para implementar sistemas de tiempo compartido.
- Como el FCFS, pero cada proceso dispone de un «cuanto de tiempo» de duración Q para estar en la CPU:
 - si transcurre Q y el proceso continúa en CPU, el planificador lo desaloja y lo ingresa al final de la cola de preparados
- La cola de preparados se gestiona con FIFO.
- Típicamente Q está entre 10 y 200 milisegundos.
- Si hay N procesos en cola, el tiempo de respuesta es como mucho: $Q \cdot (N-1)$

Round Robin: ejercicio

Proceso	Duración
P1	15
P2	4
P3	3

- Probar con $Q=4$, $Q=2$, $Q=1$

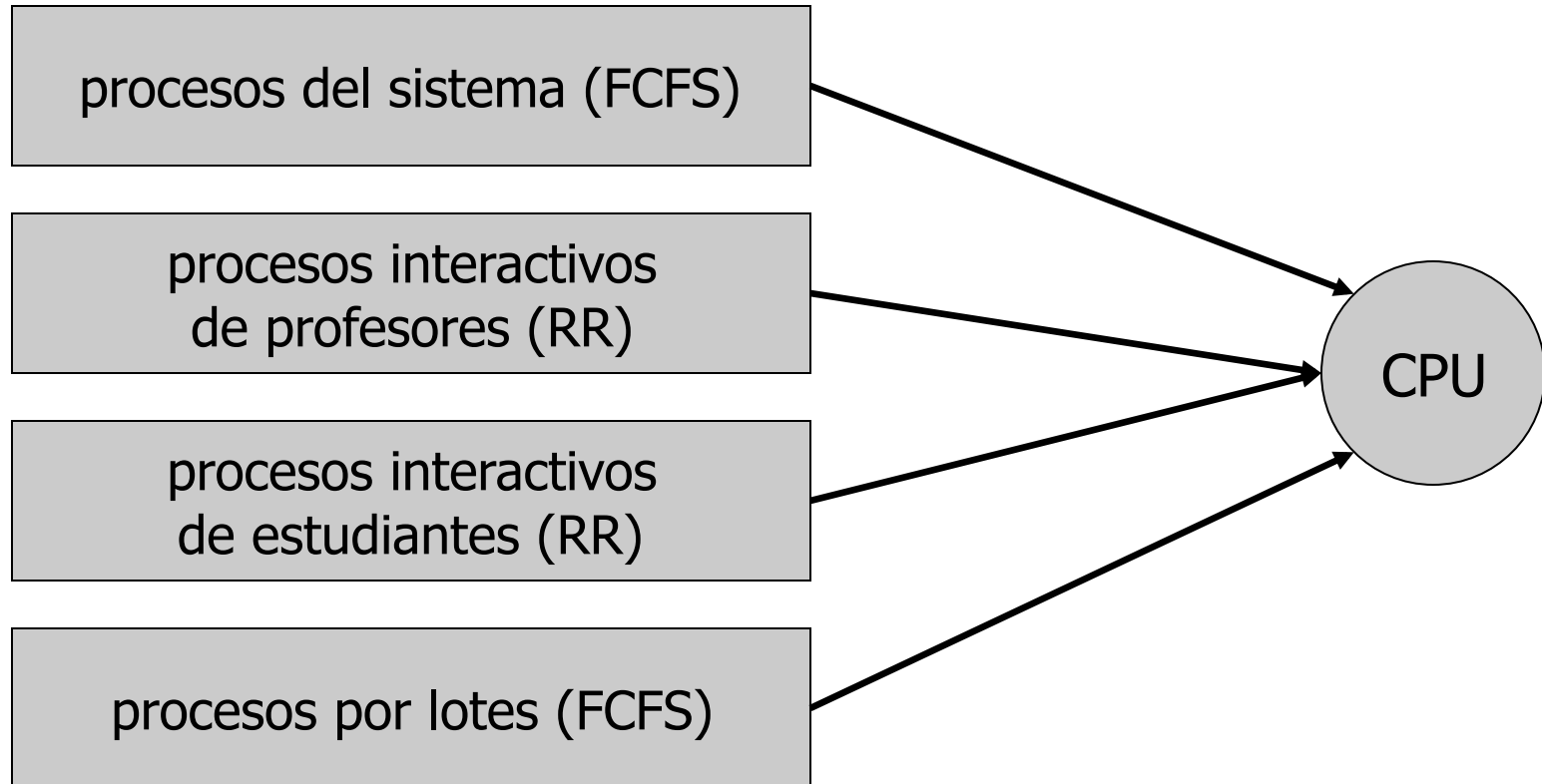
RR: influencia de Q

- Si Q es muy grande, los procesos terminan sus ráfagas de CPU antes de que termine el cuanto: se comporta como un FCFS.
- Si Q es muy pequeño, se tiende a un sistema en el que cada proceso dispone de un procesador a $1/N$ de la velocidad del procesador real (*procesador compartido*).
- Ojo, si Q es muy pequeño, ocurren más cambios de contexto y baja el rendimiento (Q debería ser mucho mayor que el tiempo que dura un cambio de contexto).

Multicolos

- Varias colas de preparados, cada una gestionada con una política diferente. Ejemplo:
 - Cola de procesos interactivos con RR
 - Cola de procesos por lotes con FCFS
- Las colas se reparten la CPU según alguna política:
 - por prioridad absoluta
 - un % de tiempo para cada cola, etc.
- **Multicolos con realimentación**
 - posibilidad de que un proceso se mueva de una cola a otra, p.ej. si cambia su comportamiento
 - Ej. UNIX clásico: un proceso que lleva mucho tiempo en espera se mueve a una cola de más prioridad

Multicolas - ejemplo



Planificación de multiprocesadores

- **Procesamiento asimétrico.** Un procesador se encarga de planificar el trabajo de los restantes (*técnica en desuso*).
- **Procesamiento simétrico.** Cada procesador realiza su planificación.
 - cola de preparados única en mem. compartida
 - una cola por procesador

Políticas para multiprocesadores

- **Afinidad al procesador.** Mantener al proceso siempre en un mismo procesador, para que aproveche los datos que puedan estar en la caché de la CPU.
- **Equilibrado de carga.** Tratar de que no haya procesadores muy ocupados u ociosos.
 - Migración comandada (*push*) → una tarea supervisora comprueba la carga y traslada procesos
 - Migración solicitada (*pull*) → un procesador ocioso puede «robar» procesos a otro más cargado

Ejemplos de políticas

- Ver capítulo 5.6 del Silberschatz
 - Solaris
 - Windows
 - Linux

Fundamentos de los Sistemas Operativos

Tema 2. Procesos FIN

© 1998-2015 José Miguel Santos - Alexis Quesada - Francisco Santana