

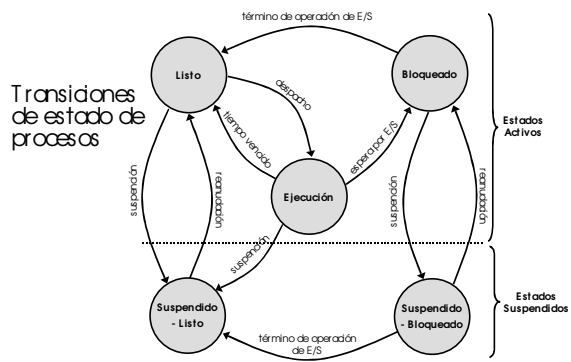
5 ITINERACIÓN DE LA CPU

- 1 CONCEPTOS BÁSICOS
- 2 CRITERIOS DE ITINERACIÓN
- 3 ALGORITMOS DE ITINERACIÓN
- 4 ITINERACIÓN DE MÚLTIPLES PROCESADORES
- 5 ITINERACIÓN EN TIEMPO REAL
- 6 EVALUACIÓN DE ALGORITMOS

5.1 CONCEPTOS BÁSICOS

- 1 Cdo de ráfagas de CPU y E/S
- 2 Itinerador de la CPU (scheduler)
- 3 Itineración Apropiativa
- 4 Despachador (dispatcher)

5.1 CONCEPTOS BÁSICOS

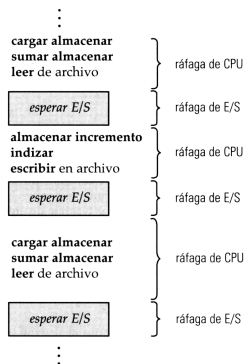


5.1.1 Cdo de ráfagas de CPU y E/S

- Utilización máxima de la CPU, obtenida mediante la multiprogramación.
- Cdo de ráfagas CPU-E/S; La ejecución de procesos consiste en un cdo (alternancia) de ejecución en la CPU y espera por E/S.

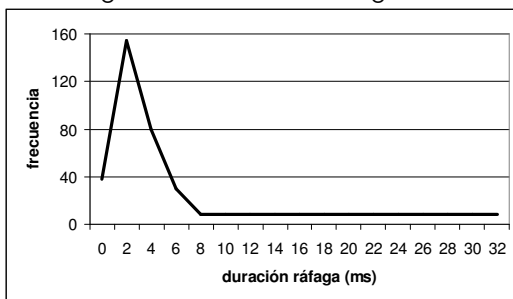
5.1.1 Cdo de ráfagas de CPU y E/S

Secuencia
alternada de
ráfagas de CPU y
E/S



5.1.1 Cdo de ráfagas de CPU y E/S

Histograma de duración de ráfagas de CPU



5.1.2 Itinerador de la CPU

El Itinerador de la CPU (o itinerador a corto plazo) selecciona de entre los procesos en memoria que están listos para ejecutarse (en cola ready) y le asigna la CPU a alguno de ellos.

5.1.3 Itineración Apropiativa

- Las decisiones de itineración de la CPU pueden ocurrir cuando un proceso:
 - 1 Se cambia de estado running a waiting.
 - 2 Se cambia de estado running a ready.
 - 3 Se cambia de estado waiting a ready.
 - 4 Termina.
- Las itineraciones 1 y 4 son no apropiativas.
- Las itineraciones 2 y 3 son apropiativas.

5.1.4 Despachador (dispatcher)

- Este módulo le traspasa el control de la CPU al proceso seleccionado por el itinerador de corto plazo; esto involucra:
 - Cambio de contexto
 - Cambiar a modo de usuario
 - “Saltar” a la ubicación apropiada en el programa de usuario para recomenzar dicho programa
- Latencia de Despacho: es el tiempo que demora el despachador en detener un proceso y recomenzar el otro; es *overhead*.

5.2 CRITERIOS DE ITINERACIÓN

- **Utilización de CPU:** mantenerla lo más ocupada posible.
- **Throughput:** número de procesos que completan su ejecución por unidad de tiempo.
- **Tiempo de Turnaround:** cantidad de tiempo en ejecutar un proceso en particular.
- **Tiempo de Espera:** cantidad de tiempo que un proceso ha estado esperando en la cola ready.
- **Tiempo de Respuesta:** cantidad de tiempo que transurre desde que un requerimiento es submitido hasta que la primera respuesta es producida (sólo para time-sharing).

5.2 CRITERIOS DE ITINERACIÓN

Optimización de los Criterios

- Máxima utilización de la CPU.
- Máximo Throughput.
- Mínimo tiempo de Turnaround.
- Mínimo tiempo de espera.
- Mínimo tiempo de respuesta.

5.3 ALGORITMOS DE ITINERACIÓN

- 1 Por orden de llegada (FCFS)
- 2 Por trabajo más breve (SJF)
- 3 Por prioridad
- 4 Por turno circular (RR)
- 5 Colas multinivel
- 6 Colas multinivel con redimentación

5.3.1 Por orden de llegada (FCFS)

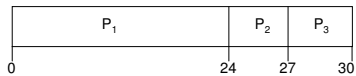
Modelo

5.3.1 Por orden de llegada (FCFS)

Ejemplo: Proceso Duración ráfaga

P ₁	24
P ₂	3
P ₃	3

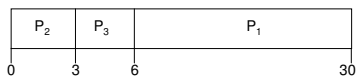
Suponer que los procesos arriban en orden: P₁, P₂, P₃. La carta Gantt para la itineración es:



- Tiempo de espera para P₁ = 0; P₂ = 24; P₃ = 27
- Tiempo de espera Promedio: $(0 + 24 + 27) / 3 = 17$

5.3.1 Por orden de llegada (FCFS)

- Si los procesos arriban en el orden: P₂, P₃, P₁.
- La carta Gantt para la itineración es:



- Tiempo de espera para P₁ = 6; P₂ = 0; P₃ = 3
- Tiempo de espera Promedio : $(6 + 0 + 3) / 3 = 3$
- Es mucho mejor que el caso anterior.
- Efecto Convoy: el proceso más breve detrás del proceso más extenso.

5.3.2 Por Trabajo más Breve (SJF)

- Se asocia con cada proceso la duración de su próxima ráfaga de CPU, la cual es usada para itinerar, seleccionando el proceso con el tiempo más breve.
- Hay dos esquemas:
 - No Apropiativo: luego que la CPU ha sido asignada, se debe esperar que el proceso complete su ráfaga.
 - Apropiativo: si un proceso nuevo arriba con una duración de ráfaga de CPU menor que el tiempo restante del proceso actualmente en ejecución, puede ser interrumpido. Es conocido como Shortest-Remaining-Time-First (SRTF).
- SJF es óptimo: asigna con tiempo de espera promedio mínimo, para un conjunto de procesos.

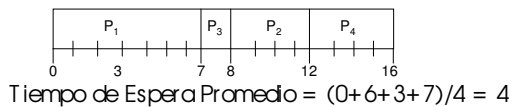
5.3.2 Por Trabajo más Breve (SJF)

Ejemplo de SJF

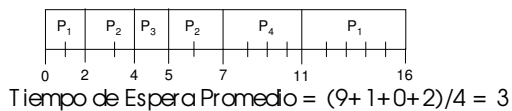
Proceso	Tiempo de Arribo	Duración Ráfaga
P ₁	0	7
P ₂	2	4
P ₃	4	1
P ₄	5	4

5.3.2 Por Trabajo más Breve (SJF)

SJF no apropiativo:



SJF apropiativo:



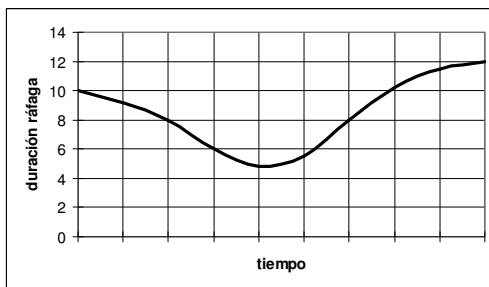
5.3.2 Por Trabajo más Breve (SJF)

Determinación de la duración de la próxima ráfaga de CPU

- Sólo se puede estimar la duración.
- Puede ser hecho utilizando la duración de las ráfagas de CPU previas, usando promedio exponencial.
 - 1 t_n = duración real de la ráfaga n ésima
 - 2 τ_{n+1} = valor estimado siguiente ráfaga
 - 3 α : $0 \leq \alpha \leq 1$.
 - 4 Se define: $\tau_{n+1} = \alpha t_n + (1 - \alpha) t_n$

5.3.2 Por Trabajo más Breve (SJF)

Determinación de la duración de la próxima ráfaga de CPU (cont)



5.3.3 Itineración por Prioridad

- Un número de prioridad (entero) es asociado a cada proceso.
- Se asigna la CPU al proceso con mayor prioridad (número menor = prioridad mayor). Puede ser:
 - Apropiativa
 - No Apropiativa
- SJF es una itineración basada en prioridad, donde la prioridad es el tiempo estimado para la siguiente ráfaga de CPU.

5.3.3 Itineración por Prioridad

- Problema: bloqueo indefinido o hambruna (starvation); los procesos de menor prioridad puede que nunca se ejecuten.
- Solución: envejecimiento (aging); a medida que el tiempo progresa, aumentar la prioridad del proceso.
- La prioridad puede ser Interna o Externa.

5.3.4 Turno Circular (Round Robin)

Modelo

5.3.4 Turno Circular (Round Robin)

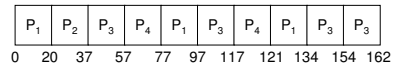
- Cada proceso obtiene un pequeño lapso de CPU (q), usualmente de 10 a 100 ms.
- Luego de transcurrido este lapso, el proceso es interrumpido y puesto al final de la cola ready.
- Si hay n procesos en la cola, cada proceso obtiene $1/n$ del tiempo de CPU de duración máxima q . Luego, ninguno esperará más de $(n-1)q$.
- Rendimiento:
 - Si q es muy grande $\Rightarrow$ FIFO
 - Si q es muy pequeño $\Rightarrow$ debe ser suficientemente grande respecto de la duración del cambio de contexto, de otra manera, el overhead sería muy grande.

5.3.4 Turno Circular (Round Robin)

Ejemplo : RR con $q=20$

Proceso	Duración ráfaga
P_1	$53 = 20 + 20 + 13$
P_2	$17 = 17$
P_3	$68 = 20 + 20 + 20 + 8$
P_4	$24 = 20 + 4$

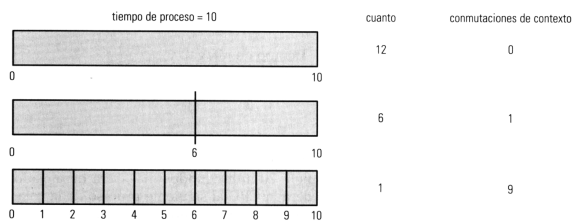
Carta Gantt:



Típicamente, tiene un tiempo promedio de turnaround mayor que SRTF, pero un mejor tiempo de respuesta.

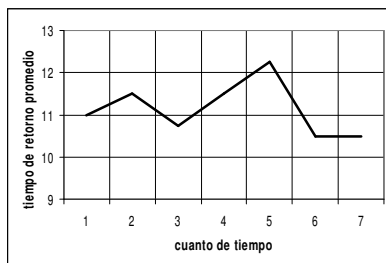
5.3.4 Turno Circular (Round Robin)

Cómo q más pequeño aumenta el Cambio de Contexto



5.3.4 Turno Circular (Round Robin)

Variación del Tiempo de Turnaround respecto de q



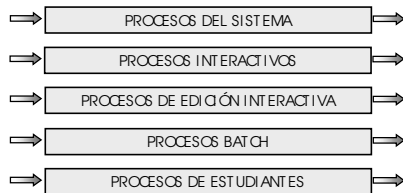
Proceso	Tiempo
P1	6
P2	3
P3	1
P4	7

5.3.5 Colas Multinivel

- La cola ready es particionada en colas separadas:
 - Foreground (interactiva)
 - Background (batch)
- Cada cola tiene su propio algoritmo de itineración
 - Foreground: RR
 - Background: FCFS
- Debe hacerse itineración entre las colas.
 - Itineración de prioridad fija: es decir, servir a todos los procesos de la cola foreground y luego a los de la cola background. Esto puede generar hambre.
 - Tijada de tiempo: cada cola obtiene una cantidad de tiempo de CPU para sus procesos; ej., 80% a foreground en RR y 20% a background en FCFS.

5.3.5 Colas Multinivel

Mayor Prioridad



Menor Prioridad

5.3.6 Colas Multinivel con Redimentación

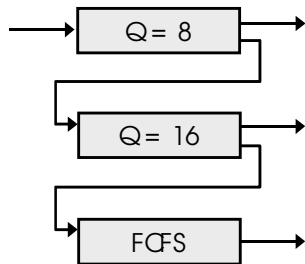
- Un proceso puede moverse entre varias colas; de esta manera, se puede implementar envejecimiento.
- Un itinerador de colas multinivel con redimentación puede ser definido por los siguientes parámetros:
 - Número de colas
 - Algoritmo de itineración para cada cola
 - Método usado para determinar cuándo cambiar un proceso de cola (hacia arriba o hacia abajo)
 - Método usado para determinar a cuál cola puede entrar un proceso cuando necesite algún servicio.

5.3.6 Colas Multinivel con Redimentación

Modelo

5.3.6 Colas Multinivel con Redimentación

Ejemplo



5.3.6 Colas Multinivel con Redimentación

Ejemplo

- Tres colas:
 - Q_0 : RR con time quantum de 8 milisegundos
 - Q_1 : RR con time quantum de 16 milisegundos
 - Q_2 : FCFS
- Itineración:
 - Un nuevo job entra en la cola Q_0 , donde es atendido en modalidad FCFS. Cuando obtiene la CPU, recibe 8 ms. Si no termina en 8 ms, es movido a la cola Q_1 .
 - En Q_1 , donde es nuevamente atendido en modalidad FCFS, obtiene 16 ms adicionales. Si aún no termina, es interrumpido y movido a la cola Q_2 .

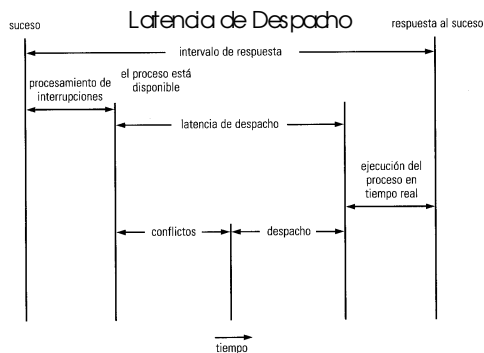
5.4 ITINERACIÓN DE MÚLTIPLES PROC

- La itineración de la CPU es más compleja cuando hay múltiples CPUs disponibles.
- En multiprocesamiento simétrico, con procesadores homogéneos, se puede tener compartimiento de la carga.
- En multiprocesamiento asimétrico, sólo un procesador accede a las estructuras de datos del sistema, dividiendo la necesidad de compartir datos. Es más simple que el simétrico.

5.5 ITINERACIÓN EN TIEMPO REAL

- Sistemas **"hard"**: requeridos para completar una tarea crítica dentro de una cantidad de tiempo restringida.
- Sistemas **"soft"**: requieren que los procesos críticos tengan mayor prioridad, sobre los demás. Son menos restrictivos que los hard.

5.5 ITINERACIÓN EN TIEMPO REAL



5.6 EVALUACIÓN DE ALGORITMOS

- Definir los criterios de evaluación.
- Definir la importancia relativa de los criterios.
Por ejemplo:
 - Maximizar la utilización de CPU, bajo la restricción de que el tiempo de respuesta sea, a lo más, de 1 seg.
 - Maximizar el throughput, de manera que el tiempo promedio de turnaround sea directamente proporcional al tiempo total de ejecución.

5.6 EVALUACIÓN DE ALGORITMOS

- Una vez definidos los criterios, se evalúan los algoritmos.
- Formas :
 - Modelamiento determinístico: forma de evaluación analítica que toma una carga de trabajo particular predeterminada y define el rendimiento de cada algoritmo para aquella carga.
 - Modelos de colas de espera
 - Simulación
- Implementación

5.6 EVALUACIÓN DE ALGORITMOS

Evaluación de itinerarios de CPU mediante Simulación

