

Unidad 2:

Gestión de Procesos

Tema 5: Planificación de procesos.

- 5.1 Criterios y tipos de planificación.
- 5.2 Algoritmos de planificación.
- 5.3 Métodos multicolos y multiprocesadores.
- 5.4 Evaluación de políticas de planificación.

5.1 Criterios y tipos de planificación.

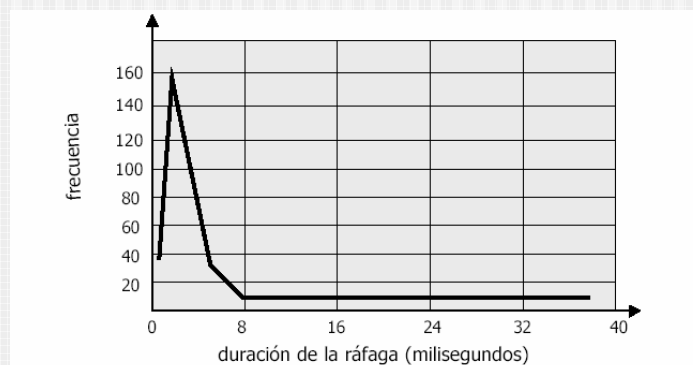
- Introducción:
- Objetivo de la **multiprogramación**:
 - **Proceso en ejecución en todo momento** $\Rightarrow$ **mayor aprovechamiento de la CPU.**
- Recordemos que:
 - **En un sistema en un instante determinado no habrá más de un proceso en ejecución.** Si hay más procesos, tendrán que esperar a que la CPU quede libre para asignarse a ella.
 - Para un mayor aprovechamiento, se mantienen **varios procesos en memoria a la vez**. Cuando un proceso necesita esperar, el SO le quita la CPU y se la da a otro proceso.
 - **Los recursos deben de planificarse antes de usarse**; la planificación es fundamental en el diseño de un SO.

5.1 Criterios y tipos de planificación.

- Ciclo de ráfagas de CPU y E/S:
 - La ejecución de un proceso consiste en un **ciclo de ejecución alternando de manera sucesiva**:
 - **Ráfagas de CPU** (que inician el proceso), durante las cuales el proceso ejecuta instrucciones.
 - **Ráfagas de E/S**, durante las cuales el proceso utiliza o espera por la E/S.
 - Se ha medido la **duración de las ráfagas de CPU** y resulta que **varían considerablemente de un proceso a otro y de un procesador a otro**, sin embargo tienden a tener una curva de frecuencia bien tipificada.

5.1 Criterios y tipos de planificación.

- Ciclo de ráfagas de CPU y E/S:
 - **Histograma de tiempos de ráfaga de CPU**:
 - Se observa:
 - Gran nº de ráfagas de CPU cortas y pocas ráfagas de CPU largas.
 - Ráfagas de CPU cortas: programas limitados por E/S (*I/O bound*)
 - Ráfagas de CPU largas: programas limitados por CPU (*CPU bound*)



5.1 Criterios y tipos de planificación.

- Decisiones de planificación:
 - Las **decisiones de planificación de CPU** se toman según las cuatro situaciones siguientes:
 - 1. Cuando un proceso pasa del estado en ejecución al estado bloqueado (bien por una operación E/S, bien por realizar una llamada al sistema)
 - 2. Cuando un proceso pasa del estado en ejecución a listo.
 - 3. Cuando un proceso pasa del estado bloqueado al estado listo.
 - 4. Cuando un proceso termina.
 - Los **casos 1 y 4** corresponden a una **planificación no expropiativa (*nonpreemptive scheduling*)**
 - Los **casos 2 y 3** corresponden a una **planificación expropiativa (*preemptive scheduling*)**

5.1 Criterios y tipos de planificación.

- Planificación **no expropiativa** (apropiativa):
 - Cuando un proceso pasa al estado de ejecución, **continúa ejecutando hasta que termina** o se bloquea en espera de una E/S.
- Planificación **expropiativa** (no apropiativa):
 - El proceso que se está ejecutando actualmente puede ser **interrumpido y pasado al estado de listo** por el sistema operativo.
 - Permite dar un mejor servicio ya que **evita que un proceso pueda monopolizar** durante mucho tiempo el procesador.

5.1 Criterios y tipos de planificación.

■ Tipos de planificación (resumen capítulo anterior):

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. (Dispatcher)
Planificación de E/S	Decisión sobre qué solicitud de E/S pendiente será tratada por un dispositivo de E/S disponible

5.1 Criterios y tipos de planificación.

■ Política *versus* Mecanismo

- Separar el mecanismo de planificación de la política de planificación, es decir, el algoritmo de planificación queda parametrizado de alguna manera, pero los parámetros pueden ser determinados por medio de procesos del usuario
- Ejemplo:
 - Un proceso puede tener muchos hijos ejecutándose bajo su control y el proceso padre conoce la relevancia de cada uno de ellos. Si un planificador no acepta datos de los procesos del usuario, relativos a decisiones de planificación, pocas veces hará la mejor elección.
 - Si se separa la política del mecanismo, Supongamos que el kernel utiliza un algoritmo de planificación, pero que proporciona una llamada al sistema por la que un proceso puede establecer (y modificar) la prioridad de sus hijos.
 - El padre puede controlar en detalle la forma de planificar sus hijos, incluso aunque él mismo no realice la planificación
 - El mecanismo está en el kernel, pero la política queda establecida por el proceso del usuario

5.1 Criterios y tipos de planificación.

■ Criterios de la planificación (a corto plazo):

■ Criterios de rendimiento

• Orientados al sistema

- **Utilización de CPU:** % de tiempo que la CPU está ocupada.
- **Productividad:** nº de procesos, trabajos, que se completan por unidad de tiempo.

• Orientados al usuario

- **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 listos (no se consideran las esperas por E/S).
- **Tiempo de respuesta:** tiempo que un proceso bloqueado tarda en entrar en la CPU desde que ocurre el suceso que lo bloquea (en sistemas interactivos).

5.1 Criterios y tipos de planificación.

■ Criterios de la planificación (a corto plazo):

■ Otros criterios

• Orientados al sistema

- **Equidad:** Los procesos deben ser tratados de igual forma y ningún proceso debe sufrir inanición (*starvation*)
- **Prioridades:** Cuando se asignan prioridades a los procesos, la política de planificación debe favorecer a los de mayor prioridad
- **Equilibrio de recursos:** La política de planificación debe mantener ocupados los recursos del sistema. Se debe favorecer a los procesos que no utilicen recursos sobrecargados

• Orientados al usuario

- **Previsibilidad:** un determinado trabajo debe ejecutarse, aproximadamente, en el mismo tiempo y con el mismo coste, sin importar la carga del sistema

5.1 Criterios y tipos de planificación.

- Criterios de planificación:
 - **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.
 - En sistemas interactivos, como los sistemas de tiempo compartido, se considera también la **estadística** (medias) de esas magnitudes.
 - **No existe ninguna política de planificación óptima.** La bondad de un método depende de:
 - Tipo de procesos.
 - Criterio de planificación.

5.2 Algoritmos de planificación.

- FCFS (First Come, First Served).
- SJF (Shortest Job First) o SPN (Shortest Process Next).
- SRTF (Shortest Remaining Time First).
- Planificación por prioridad.
- Planificación por turno circular, RR (Round Robin).

- Realimentación.
- HRRN (Highest Response Ratio Next).

5.2 Algoritmos de planificación.

■ FCFS (*First Come, First Served*), ejemplo:

■ Planificación de servicio por orden de llegada:

- Cuando el proceso actual cesa su ejecución se selecciona el proceso más antiguo de la cola.

■ Problema propuesto:

- Calcular el **tiempo de espera, tiempo de retorno y tiempo medio de espera** si aplicamos el algoritmo FCFS suponiendo que los procesos siguientes llegan en el mismo instante y en el orden: P1, P2, P3.
- ¿Y si el orden de llegada es: P2, P3, P1?

Proceso	Duración
P1	9
P2	4
P3	2

5.2 Algoritmos de planificación.

■ FCFS, ejemplo:

Proceso	Duración	<i>Diagrama de Gantt</i>			
P1	9	0	9	13	15
P2	4				
P3	2				

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):

0	4	6	15
P2	P3	P1	

5.2 Algoritmos de planificación.

■ FCFS (Conclusiones):

- Planificación de **servicio por orden de llegada**
- Algoritmo **no expropiativo (apropiativo)**
- **Ventajas:**
 - Es el algoritmo más sencillo, **el primer proceso que solicita la CPU es el primero en recibirla.**
 - Fácil de implementar con una **política FIFO** para la cola de listos.
- **Inconvenientes:**
 - **Tiempo de espera promedio bastante largo.**
 - Bajo nivel de utilización de la CPU
 - Pobre tiempo de respuesta en procesos cortos en esquemas con mucha carga
 - Tiende a favorecer a los procesos con carga de CPU frente a los que tienen carga de E/S
 - Uso ineficiente de los dispositivos de E/S

5.2 Algoritmos de planificación.

- SJF (*Shortest Job First*), ejemplo:
 - **Primero el trabajo más corto:**
 - Se selecciona el proceso con menor tiempo de ejecución.
 - **Problema propuesto:**
 - Calcular el tiempo medio de espera que resulta de aplicar:
 - Un algoritmo SJF no expulsivo.
 - Un algoritmo SJF expulsivo (SRTF, Shortest Remaining Time First).

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

5.2 Algoritmos de planificación.

■ 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	0	7	8	12	16
espera media: $(0+6+3+7)/4=4$	P1 P3 P2 P4				

SJF expulsivo	0	2	4	5	7	11	16
espera media: $(9+1+0+2)/4=3$	P1 P2 P3 P2 P4 P1						

5.2 Algoritmos de planificación.

■ SJF (Conclusiones):

- Entra en CPU el **proceso con ráfaga de CPU más corta**.
- Un **proceso corto saltará a la cabeza de la cola**, “adelantando” a trabajos largos.
- **Ventaja:**
 - **Minimiza el tiempo de espera medio.**
- **Inconvenientes:**
 - **Riesgo de inanición** de los procesos de larga duración.
 - No es implementable, necesitamos un método para predecir cuánto durará una ráfaga del proceso (basándonos en la ráfaga anterior).
 - Se puede estimar la duración, pero se pierde mucho tiempo en hacer la estimación
- **Versión expulsiva (SRTF):** el proceso en CPU es desalojado si llega a la cola un proceso con duración más corta.

5.2 Algoritmos de planificación.

- Planificación por prioridades:
 - Cada proceso tiene una prioridad, **entrará primero en la CPU el que tenga mayor prioridad.**
 - Habrá **múltiples colas de procesos listos**, para representar cada nivel de prioridad.
 - **Política de prioridades expulsiva o no.**
 - La prioridad se puede definir:
 - De forma interna, la define el SO.
 - De forma externa, la definen los usuarios.
 - SJF es un caso de planificación por prioridad.
 - Problema para **procesos de muy baja prioridad:**
 - **Riesgo de inanición**
 - **Solución: envejecimiento** (permite que un proceso cambie su prioridad en función de su edad o su historial de ejecución).

5.2 Algoritmos de planificación.

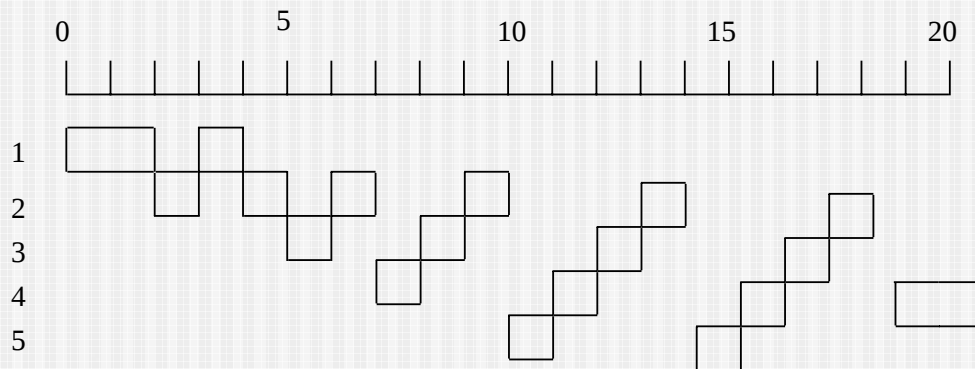
- Planificación por turno circular, RR (Round Robin):
 - **Comportamiento como FCFS**, con la diferencia de que cada proceso dispone de un cuanto de tiempo máximo.
 - **Problema propuesto:**
 - Dibujar el diagrama de Gantt para la carga de trabajos:

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

5.2 Algoritmos de planificación.

■ Planificación por turno circular, RR (Round Robin):

- Si cuando expira el cuanto de tiempo el proceso continúa en la CPU, el planificador lo desaloja y lo ingresa al final de la cola de listos.
- Diagrama de Gantt:



5.2 Algoritmos de planificación.

■ Planificación por turno circular, RR (Round Robin):

■ Conclusiones:

- **Periódicamente, se genera una interrupción de reloj.**
- Cuando se genera la interrupción, el proceso que está en ejecución se sitúa en la cola de listos (FIFO).
- Se determina una cantidad de tiempo (10-100 ms) que permite al proceso utilizar el procesador durante este periodo de tiempo.
 - Según sea el **cuanto de tiempo, 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 $\rightarrow$ 0**, 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**).
 - Pero, si **Q es muy pequeño** se suceden **constantemente los cambios de contexto** y del rendimiento disminuye mucho.
- Adecuado para implementar **tiempo compartido**.

5.2 Algoritmos de planificación.

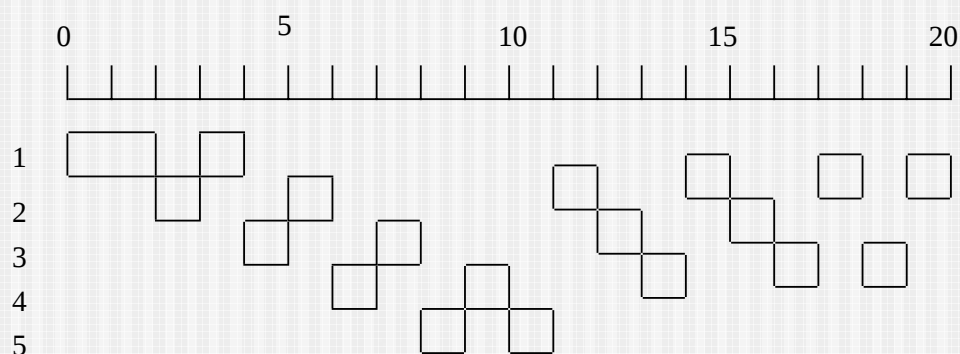
■ Primero el Trabajo con Mayor Tasa de Respuesta (HRRN, *Highest response ratio next*)

- Cuando el proceso actual termina o se bloquea, se elige el proceso con un mayor valor para la tasa de respuesta $R = (w + s)/s$, donde w es el tiempo del proceso en el sistema y s es el tiempo total requerido por el proceso
- Mejora la planificación SJF
- Es una política apropiativa
- Tiene en cuenta la edad del proceso
- Previene la inanición de los procesos

5.2 Algoritmos de planificación.

■ Realimentación:

- **Penaliza a los trabajos que han estado ejecutándose más tiempo.**
- No se conoce el tiempo de ejecución restante del proceso.
- **Problema propuesto:**
 - Dibujar el diagrama de Gantt para la misma carga de trabajos que en RR.



5.3 Métodos Multicola y Multiprocesador.

- Planificación con colas de múltiples niveles:
 - Procesos diferentes tienen necesidades diferentes
 - Procesos interactivos cortos y limitados en E/S, deberían ejecutarse, generalmente, antes que procesos batch limitados en CPU
 - El patrón de comportamiento no es inmediatamente obvio para el planificador
 - Un algoritmo de planificación con colas de múltiples niveles **divide la cola de procesos listos en varias colas.**
 - **Cada cola tiene su propio algoritmo de planificación.**
 - Existirá, por tanto, una **planificación entre colas.**
 - Las colas se pueden planificar según:
 - **Planificación expropiativa** de prioridades fijas.
 - **Dividiendo el tiempo de CPU** entre las colas.

5.3 Métodos Multicola y Multiprocesador.

- Planificación con colas de múltiples niveles y realimentación:
 - A diferencia del caso “sin realimentación”, ahora **se permite que un proceso pase de una cola a otra.**
 - Si un proceso **gasta demasiado tiempo de CPU**, se le pasa a una **cola de menor prioridad.**
 - Si un proceso **espera mucho tiempo** en una cola de baja prioridad **se le puede pasar a una de mayor prioridad**, evitando la inanición. (Caso del sistema UNIX)

5.3 Métodos Multicola y Multiprocesador.

- Planificación con colas de múltiples niveles y realimentación:
 - Un **planificador de colas multinivel con realimentación** está definido por los siguientes parámetros:
 - El **número de colas**.
 - El **algoritmo de planificación** para cada cola.
 - Los **métodos para determinar cuando mover un proceso a una cola de distinta prioridad a la que ocupa**.
 - El **método para determinar en qué cola ingresará un proceso**.

5.3 Métodos Multicola y Multiprocesador.

	FIFO	RR	SJF (*)	HRRN (*)	SRTF (*)	MFQ
Función Selección	$\text{Max}\{w\}$	Constante, quantum	$\text{Min}\{s\}$	$\text{Max}\{(w+s)/s\}$	$\text{Min}\{s-e\}$	$\text{Min}\{w\}$, utilizando colas
Modo Decisión	Apropiativo	Expropiativo	Apropiativo	Apropiativo	Expropiativo	Expropiativo
Rendimiento	No resalta	Bajo, si el quantum es muy bajo	Alto	Alto	Alto	No resalta
Tiempo Respuesta	Elevado, si varianza grande	Bueno, para procesos cortos	Bueno, para procesos cortos	Bueno	Bueno	No resalta
Efecto	Penaliza procesos cortos y limitados en E/S	Tratamiento justo	Penaliza procesos largos	Buen balance	Penaliza procesos largos	Puede favorecer procesos limitados E/S
Inanición	No	No	Posible	No	Posible	No
(*) Requiere conocer el tiempo requerido por el proceso w = tiempo en el sistema; e = tiempo de ejecución; s = tiempo total requerido						

5.3 Métodos Multicola y Multiprocesador.

- Planificación en multiprocesadores:
 - **Varias CPU => planificación más compleja.**
 - Supondremos procesadores de igual funcionamiento:
 - **Cualquier procesador disponible podrá ejecutar cualquier proceso de la cola.**
 - Generalmente: una **cola común para todos los procesos.**
 - **Todos los procesos ingresan en la misma cola** y se les asigna cualquier procesador que esté disponible.
 - **No habrá procesadores ociosos** con su cola vacía.
 - **No habrá procesadores con muy alta ocupación.**
 - **Dos posibles estrategias:**
 - **Cada procesador se autoplanifica**, asegurándonos de que:
 - Dos procesadores no escojan el mismo proceso.
 - No se quede ningún proceso sin escoger.
 - **Un procesador actúa como planificador de los demás procesadores.**

5.4 Evaluación de políticas de planificación.

- ¿Cómo escoger el algoritmo de planificación a utilizar?
 - **Criterios de evaluación de un algoritmo:**
 - Grado de utilización de la CPU.
 - Tiempo de respuesta.
 - Rendimiento.
 - **Evaluación de políticas:**
 - Modelado determinista.
 - Modelos de colas (estadísticos).
 - Simulaciones.
 - Implementación.

5.4 Evaluación de políticas de planificación.

■ Modelado determinista:

■ Evaluación analítica de algoritmos: **se calcula el desempeño de un algoritmo teniendo en cuenta la carga de trabajo del sistema:**

- 1. Se definen los criterios de rendimiento.
- 2. Se buscan los algoritmos candidatos.
- 3. Se establece una carga de trabajo representativa del sistema.
- 4. Para cada algoritmo:
 - Sometemos la carga de trabajo a su planificación.
 - Evaluamos su rendimiento en función de los criterios de 1.
- 5. Seleccionamos el que mejor se comporte.

5.4 Evaluación de políticas de planificación.

■ Modelado determinista:

■ Características:

- **Cómodo** de realizar.
- **Proporciona magnitudes exactas** con las que comparar los algoritmos.
- **Limitación de su validez** en cuanto a que se le somete a una carga concreta de trabajo.

5.4 Evaluación de políticas de planificación.

■ Modelos de colas:

■ Metodología:

- Determinación de la **distribución (estadística) de ráfagas de CPU y de E/S.**
- **Distribución de los tiempos de llegada al sistema.**

■ Resultado: probabilidad de una ráfaga de CPU dada.

- **Distribución exponencial** que se describe en términos de su media.
- Cálculos de:
 - **Rendimiento promedio.**
 - **Tiempo de espera.**
 - **Aprovechamiento.**

■ Sistema informático como:

- **Red de servidores**, cada servidor con su cola de procesos en espera.

5.4 Evaluación de políticas de planificación.

■ Simulaciones:

■ Consisten en programar un modelo del sistema de computación:

- Conforme se ejecuta la simulación se recopilan e imprimen datos estadísticos que indican en desempeño del algoritmo.

■ Método más común de generación de datos: generador de números aleatorios programado para generar (según distribuciones de probabilidad):

- **Procesos.**
- **Tiempo de ráfagas de CPU.**
- **Llegadas, partidas, ...**

■ Problemas:

- **Simulaciones pueden ser costosas**, requieren mucho tiempo de computación y recursos.
- Cintas de rastreo, registran secuencias de sucesos reales, requieren **mucho espacio de almacenamiento.**
- **Tarea compleja de diseño, codificación y depuración** del simulador.

5.4 Evaluación de políticas de planificación.

■ Implementación:

- Es la **única forma exacta de evaluar un algoritmo de planificación.**
- Metodología:
 - Codificarlo.
 - Colocarlo en el SO.
 - Probar su funcionamiento.
- Problema:
 - Coste elevado:
 - Modificación del SO.
 - Dificultar el trabajo de los usuarios, puesto que el SO está en continuo cambio.

5.4 Evaluación de políticas de planificación.

■ Solución:

- Plantear un **esquema híbrido** del tipo:
 - 1. **Análisis preliminar de las políticas candidatas** mediante modelos deterministas.
 - 2. **Simulación de la opción u opciones más ventajosas.**
 - 3. **Implementación de la opción óptima:**
 - Primero en un sistema de desarrollo (pruebas).
 - Finalmente en un sistema de producción.
- Problema:
 - Entorno **dinámico y variable** (nuevos programas, nuevos problemas).
 - Característica deseada: Planificación flexible, **separación clara entre mecanismos y políticas.**