

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.

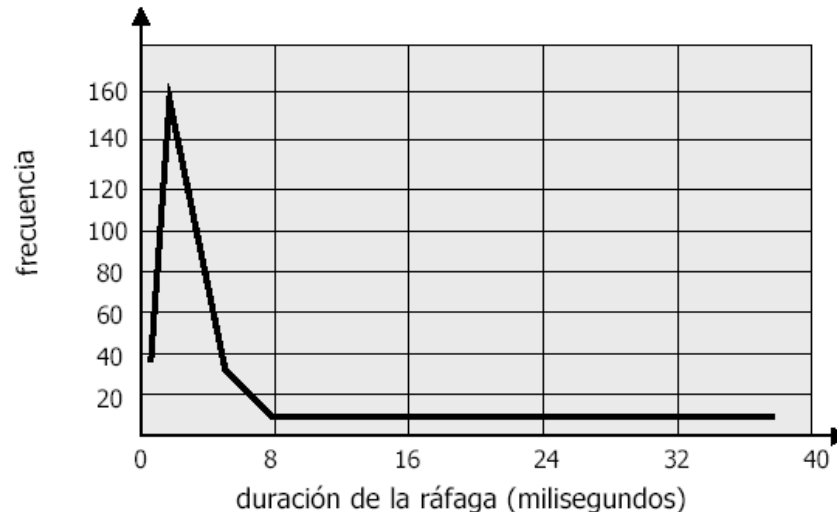
- Introducción:
 - Objetivo de la **multiprogramación**:
 - Tener un **proceso en ejecución en todo momento** para un **mayor aprovechamiento de la CPU**.
 - **En un sistema monoprocesador nunca habrá más de un proceso en ejecución**. Si hay más procesos, tendrán que esperar a que la CPU quede libre para reasignarse.
 - 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.
 - Casi todos **los recursos del computador se planifican 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.
 - Ráfagas de CPU largas: programas limitados por CPU.



5.1 Criterios y tipos de planificación.

- Planificación expropiativa:
 - Las **decisiones de planificación de CPU** se toman según las cuatro situaciones siguientes:
 - 1. Cuando un proceso pasa del estado en ejecución a en espera.
 - 2. Cuando un proceso pasa del estado en ejecución a listo.
 - 3. Cuando un proceso pasa del estado en espera al estado listo.
 - 4. Cuando un proceso termina.
 - Los **casos 1 y 4** corresponden a una **planificación no expropiativa** (a la fuerza hay que escoger un nuevo proceso).
 - Los **casos 2 y 3** corresponden a una **planificación expropiativa**.

5.1 Criterios y tipos de planificación.

- Criterios de planificación:
 - **Utilización de CPU:** % de tiempo que la CPU está ocupada.
 - **Rendimiento:** n^o de procesos, trabajos, que se completan por unidad de tiempo.
 - **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 la CPU desde que ocurre el suceso que lo bloquea.

5.1 Criterios y tipos de planificación.

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

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.

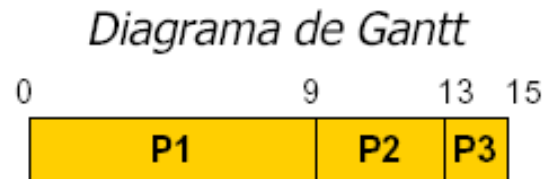
- FCFS (First Come, First Served), ejemplo:
 - **Planificación de servicio por orden de llegada.**
 - Calcular el **tiempo de espera, tiempo de retorno y tiempo medio de espera** si aplicamos el algoritmo FCFS suponiendo que los procesos siguientes llegan en el mismo instante y en el orden: P1, P2, P3.
 - ¿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
P1	9
P2	4
P3	2

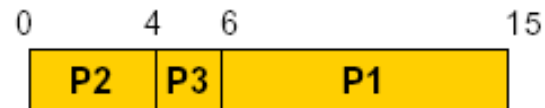


Tiempos de espera: P1=0; P2=9; P3=13

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):



5.2 Algoritmos de planificación.

- FCFS:
 - **Planificación de servicio por orden de llegada.**
 - **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 preparados.**
 - **Tiempo de espera promedio bastante largo.**

5.2 Algoritmos de planificación.

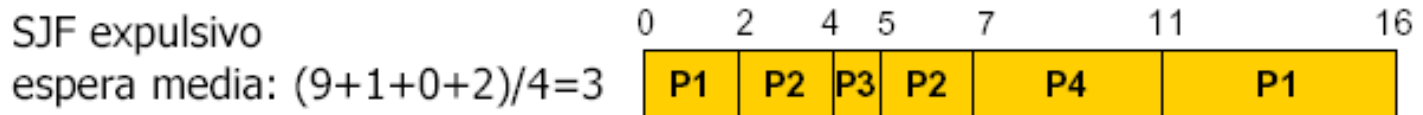
- SJF (Shortest Job First), ejemplo:
 - **Primero el trabajo más corto.**
 - Calcular el tiempo medio de espera que resulta de aplicar:
 - Un algoritmo SJF no expulsivo.
 - Un algoritmo SJF expulsivo (SRTF, Shortest Remaining Time First).

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



5.2 Algoritmos de planificación.

- SJF:
 - Entra en la CPU el proceso con la ráfaga de CPU más corta.
 - Minimiza el tiempo de espera medio.
 - Riesgo de inanición de los procesos de larga duración.
 - No es implementable. Se pueden estimar las duraciones de los procesos, según su historia reciente.
 - 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.

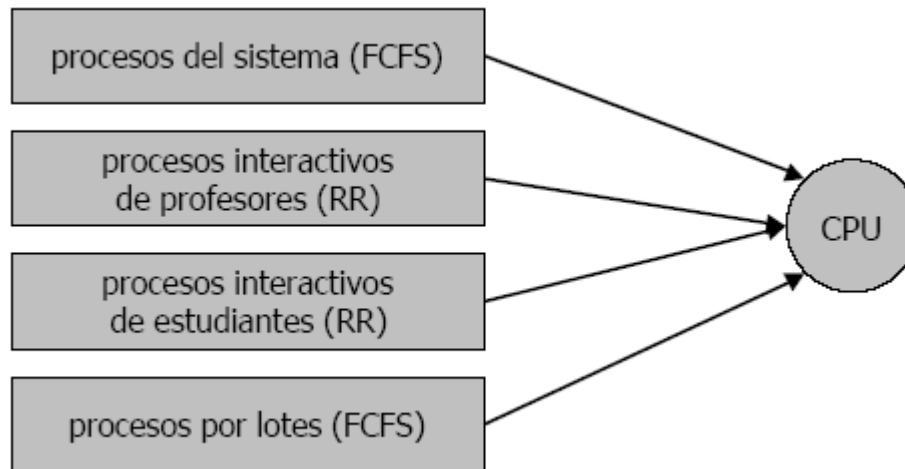
- Planificación por prioridad:
 - Cada proceso tiene una prioridad, **entrará primero en la CPU el que tenga mayor prioridad.**
 - **Política de prioridades expulsiva o no.**
 - La prioridad se puede definir:
 - De forma interna, la define el SO.
 - De forma externa, la definen los usuarios.
 - SJF es un caso de planificación por prioridad.
 - Los **procesos de prioridad más baja** tienen **riesgo de inanición.**
 - **Solución: envejecimiento.** Ir aumentando de forma progresiva la prioridad de los procesos en espera.

5.2 Algoritmos de planificación.

- Planificación por turno circular:
 - **RR (Round Robin):**
 - Adecuado para implementar **tiempo compartido**.
 - **Comportamiento como FCFS**, con la diferencia de que cada proceso dispone de un cuanto de tiempo máximo.
 - Si cuando expira el cuanto de tiempo 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 como FIFO.**
 - 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**).
 - Si **Q es muy pequeño** se suceden **constantemente los cambios de contexto** y del rendimiento disminuye mucho.

5.2 Métodos Multicola y Multiprocesador.

- Planificación con colas de múltiples niveles:
 - Distinguímos entre procesos que pueden tener **distintas necesidades en cuanto al tiempo de respuesta y por tanto distintas necesidades de planificación.**
 - **Procesos** que se ejecutan en primer plano o **interactivos.**
 - **Procesos** que se ejecutan en segundo plano o **por lotes.**



5.2 Métodos Multicola y Multiprocesador.

- Planificación con colas de múltiples niveles:
 - 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.

5.2 Métodos Multicola y Multiprocesador.

- Planificación con colas de múltiples niveles y realimentación:
 - **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)

5.2 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**.

5.2 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.**
 - **Uso de cola común para todos los procesos:**
 - **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.**

5.2 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.

5.2 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.

5.2 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.

5.2 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 ...**
 - **Sistema informático como:**
 - **Red de servidores**, cada servidor con su cola de procesos en espera

5.2 Evaluación de políticas de planificación.

- Simulaciones:
 - **Programación de un modelo** del sistema de computación.
 - **Generación de datos:** generador de números aleatorios modificado para generar:
 - **Procesos.**
 - **Tiempo de ráfagas de CPU.**
 - **Llegadas, partidas, ...**
 - **Problemas:**
 - **Simulaciones costosas**, requieren mucho tiempo y recursos.
 - Cintas de rastreo, registran secuencias de sucesos reales, requieren **mucho espacio de almacenamiento.**
 - **Tarea compleja de diseño, codificación y depuración** del simulador.

5.2 Evaluación de políticas de planificación.

- Implementación:
 - Método más fiable, **forma exacta de evaluar un algoritmo de planificación.**
 - Metodología:
 - Codificarlo.
 - Colocar 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.

5.2 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.**