Sistemas Operativos I

Módulo III

PLANIFICACIÓN DE LA CPU

1

Introducción

Sistemas multitarea

Característica principal: múltiples procesos de usuarios y del sistema operativo comparten dinámicamente los recursos físicos y lógicos del sistema.

Recursos del sistema: procesador, memoria, archivos, dispositivos de entrada-salida, etc.

Módulos siguientes de Sistemas Operativos I

Se estudiarán temas relacionados con la administración de los recursos más importantes del sistema:

Módulo III: Planificación del procesador.

Módulo IV: Administración de la memoria.

Módulo V: Administración de archivos y dispositivos de E/S.

2

Introducción

Objetivo de los SO multitarea: maximizar la utilización de los recursos del sistema por parte de los procesos.

Utilización del Procesador: se deben planificar los procesos para optimizar su uso.

¿Cuál es la importancia de planificar el uso del procesador?

Planificación del procesador:

- Maximizar su aprovechamiento tiene gran incidencia en el rendimiento global del sistema: p. ej., se debe alcanzar un alto throughput de procesos.
- Throughput: número de procesos ejecutados totalmente por unidad de tiempo. Este es uno de los parámetros indicadores de la eficiencia del sistema en la computación de datos.

3

Introducción

Planificación del procesador *(cont.)*

Planificando eficientemente los procesos que están en la cola Ready ¿es suficiente para obtener un alto throughput?

Es **sólo una** de las tareas para alcanzar el objetivo anterior. Hay otras que también son importantes.

4

Introducción

Consideraciones generales de la planificación de procesos

- Durante su vida en el sistema, un proceso transiciona por distintos estados, existiendo colas asociadas a cada estado.
- Una de las tareas del núcleo es la de seleccionar y mover los procesos entre colas, para lo cual utiliza diferentes algoritmos que inciden en el rendimiento del sistema.
- Punto de vista general: seleccionar y mover procesos entre colas o ponerlos a ejecutar se denomina **planificación o "scheduling"**.
- Dentro del SO existen varios programas planificadores o schedulers. Uno de ellos (el de mayor importancia) es el **planificador de la CPU** o **"dispatcher"**.
- Otro planificador importante puede ser el planificador de disco.

5

Introducción

Planificación de procesos para uso de la CPU

- Los algoritmos usados para planificar la CPU son diferentes y de mayor complejidad que los algoritmos que planifican la utilización de otros dispositivos de E/S.
- Esta diferencia es por la incidencia que tienen en la performance global del sistema y por ser la CPU el recurso más escaso y más rápido del mismo.

6

Introducción

Parámetros indicadores de la eficiencia del sistema

- **Tiempo de respuesta:** Tiempo de primera salida de datos desde que comenzó a ejecutar un programa.
- **Throughput de procesos:**
  (número de procesos terminados) / (unidad de tiempo)
- **Eficiencia del procesador:**
  (tiempo útil) / (tiempo total de uso)
  - Tiempo útil: tiempo neto de cómputo (ejecución de programas de usuario).
  - Tiempo total de uso: tiempo útil + tiempo de no cómputo.
  - Tiempo de no cómputo: tiempo usado en cambios de procesos, procesamiento de IRQ, compactación de memoria, de disco, etc.

7

Tipos de Planificadores

Planificadores existentes en el sistema

- El objetivo de los planificadores de CPU es asignar procesos al procesador para optimizar el tiempo de respuesta, el throughput y el uso eficiente del procesador.
- Para ello, se requiere de la tarea conjunta de **3 planificadores diferentes:**
- Planificador de largo plazo o long-term scheduler.
- Planificador de mediano plazo o medium-term scheduler.
- Planificador de corto plazo o short-term scheduler.

8

Actuación de los Planificadores

Los distintos planificadores actúan sobre colas diferentes.

Long Term:
Actúa cuando se crea un proceso nuevo. (No todos los SO lo tienen). Algunos SO no limitan a priori el ingreso de procesos al sistema, sino que la limitación se da cuando rebalsan las tablas internas (de procesos, de memoria, etc.) o por razones de performance.

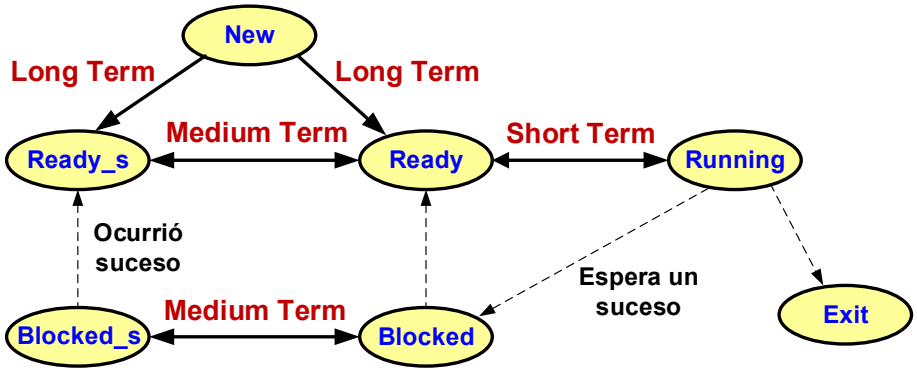
Medium Term:
Participa en la operación de swapping (en la cola Ready o en Blocked). Analiza la disponibilidad de memoria principal en relación a las prioridades de los procesos.

Short Term:
Realiza la selección de un nuevo proceso para usar la CPU (actúa sobre procesos en la cola de Ready).

9

Tipos de Planificadores

Planificadores en un diagrama de estados de procesos



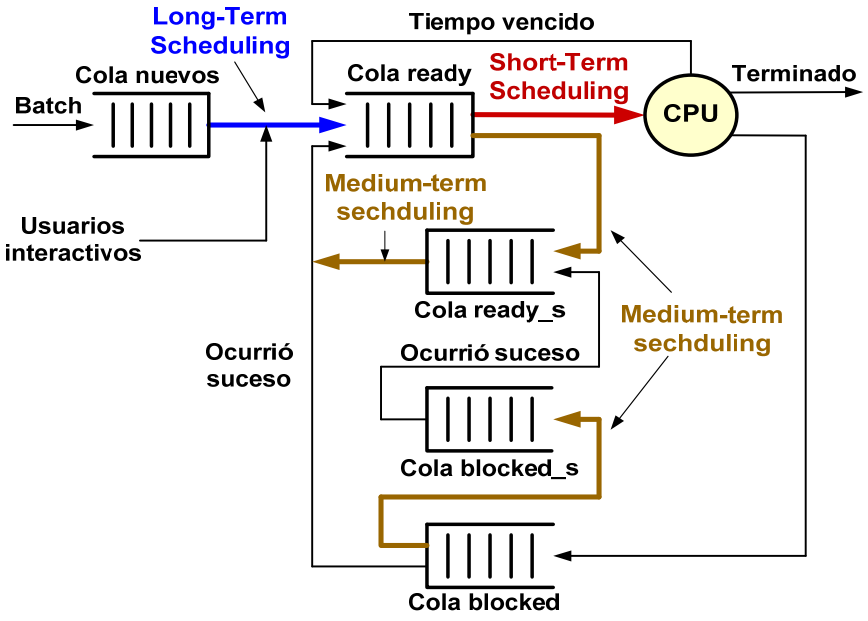
10

Actuación de los Planificadores

- Los planificadores actúan sobre colas diferentes asociadas a los estados de procesos, con particulares restricciones de tiempo por lo que pueden usar distintas directivas (algoritmos) para manipular los procesos.
- En el siguiente diagrama se representan estas colas y lo eventos que disparan las transiciones.

11

Diagrama de Colas y Planificadores



12

Long-Term Scheduler

- Es el encargado de controlar el grado de multiprogramación que admite el SO a fin de evitar congestión y bajo rendimiento.
- Actúa en la admisión de los procesos nuevos a la cola Ready.
- **Sistemas que admiten trabajos por lotes (batch):** existe un planificador de largo plazo. Las tareas nuevas que ingresan al sistema son almacenadas en disco y mantenidos en cola de procesos batch. El planificador llevará procesos de esta cola a lo cola Ready, si el SO los admite y de acuerdo a prioridades.
- **Sistemas de tiempo compartido:** cuando se crea un nuevo proceso interactivo, el SO **no lo pone en una cola**, sino que directamente son admitidos en el sistema hasta que éste llega a un estado de saturación. En cuyo caso se rechaza el proceso y se notifica al usuario de tal situación, invitándolo a intentar en otra ocasión.
- Normalmente, Long-Term Scheduler utiliza el algoritmo FIFO.

13

Medium-Term Scheduler

Planificación de mediano plazo: forma parte de la operación de swapping en los computadores.

- Decisión de desalojar un proceso de memoria principal. Depende de:
  1. Por un lado, tratar de mantener un **grado máximo de multiprogramación**.
  2. Por otro lado, el grado de multiprogramación está limitado por el espacio disponible en memoria.
- Tarea del Medium-Term Scheduler: debe conciliar lo mejor posible entre 1 y 2 (Memoria virtual).

14

Short-Term Scheduler

También es llamado **dispatcher**, o directamente **scheduler**.

Es el encargado de sacar los procesos en estado de Ready y ponerlos en estado de Running.

- **El dispatcher planifica (elige un proceso) cuando:**

1. Un proceso abandona la CPU porque:
   - Ha terminado su ejecución.
   - Pasa de Running a Blocked.
   - Pasa de Running a Ready.
2. Un proceso pasa de Blocked a Ready.

15

Short-Term Scheduler

¿Cuándo entra en acción el dispatcher?

- En general, entra en acción cada vez que ocurre una interrupción del proceso que está en uso de la CPU. (IRQ del timer, system call, trap, etc.).
- El hecho de que el scheduler entre en acción no quiere decir que vaya a planificar (elegir un proceso):

  Primero analiza si corresponde planificar. Si corresponde, planifica; si no, devuelve el procesador al proceso anterior.

  Esta es otra forma de diferenciar entre cambio de proceso y cambio de modo.

  ¿Cuándo decide planificar? Cuando el proceso sí o sí será desalojado de la CPU (cambio de proceso).

16

Tipos de Algoritmos de Planificación de la CPU

Los algoritmos pueden ser:

No apropiativos o no expulsivos (non-preemptives).

Apropiativos o expulsivos (preemptives).

- Algoritmos **no apropiativos:** cuando el proceso abandona la CPU sólo en forma voluntaria (cuando finaliza o se bloquea).

  Ejemplo: MS Windows v3.x.

- Algoritmos **apropiativos:** cuando el uso de la CPU (por parte de un proceso) está siendo controlado por el núcleo.
- Motivos de retiro de un proceso del procesador:
  - Expiración de la cuota (quantum) de tiempo de uso del CPU.
  - Ingreso a la cola Ready de un proceso de mayor prioridad.

  Ejemplo de SO: UNIX, MS Windows 20xx/Win10, IBM OS/2.

17

Tipos de Algoritmos de Planificación de la CPU

- Algoritmos **apropiativos** (*cont.*)**:** significan una mayor sobrecarga al sistema debido al gran número de cambios de procesos que generan respecto de los no apropiativos.

  Sin embargo, los apropiativos proveen un mejor servicio global.

  SO que usan estos algoritmos: UNIX, MS Windows 20xx/Win10, IBM OS/2.

18

Algoritmos de Planificación de la CPU

Algoritmos de planificación

- **First-Come, First-Served (FCFS):** es el más simple. Es no apropiativo por definición. No se utiliza en sistemas de tiempo compartido. El proceso elegido ejecuta hasta que finaliza.
- **Shortest-Job-First (SJF):** otorga la CPU al proceso que la utilizará por menor tiempo en el futuro. En caso de haber más de un proceso en estas condiciones, se usa FCFS. Puede implementarse en forma apropiativa, con una versión no apropiativa, llamada **Shortest Remaining Time (SRT)**.

  Desventaja: es complicado determinar a futuro la duración de una tarea.

19

Algoritmos de Planificación de la CPU

Algoritmos de planificación *(cont.)*

- **Prioritario:** cada proceso tiene asociada una prioridad. La CPU se otorga al proceso que en la cola tenga la mayor prioridad. ¿Varios procesos de igual prioridad? Se usa FCFS.

  Para poder usar este algoritmo se debe asignar una prioridad a los procesos.
- Prioridad: es un número que debe calcularse y, normalmente, es función de diversos enfoques que son representados por parámetros. Por ejemplo:
  a) Desde el punto de vista interno o externo al sistema; o una combinación de ambos.
  b) La prioridad puede ser estática o dinámica en el tiempo.

20

Algoritmos de Planificación de la CPU

Algoritmos de planificación *(cont.)*

- **Prioritario** *(cont.)*

  Definición interna de la prioridad: depende de los requerimientos de computación del proceso en sí: tiene en cuenta los recursos de hardware y software que utiliza el proceso en cuestión (tiempo que utilizó la CPU, tiempo total que lleva en el sistema, memoria utilizada, número de archivos abiertos, etc.).

  Definición externa de la prioridad: requerimientos independientes del SO: prioridades de determinados usuarios; procesos del sistema vs. procesos de usuarios.

  Las prioridades interna y externa se combinan para calcular una única prioridad, la que será usada a los fines de la planificación.

21

Algoritmos de Planificación de la CPU

Algoritmos de planificación *(cont.)*

- **Prioritario** *(cont.)*
  - Además, puede ser apropiativo o no apropiativo.
  - Puede presentar un grave problema: inanición. Una solución posible es aumentar la prioridad cada determinado tiempo para que pueda ejecutar; así, hasta finalizar.
- **Round Robin (RR):** fue especialmente diseñado para sistemas de tiempo compartido. Se define una pequeña unidad de tiempo denominada quantum o time slice (entre 10ms y 1s). La cola Ready se implementa como una "cola circular" donde el scheduler otorga un quantum a cada proceso en dicha cola. De esta forma todos los procesos tienen garantizado el uso de la CPU. Este algoritmo es apropiativo por naturaleza.

22

Algoritmos de Planificación de la CPU

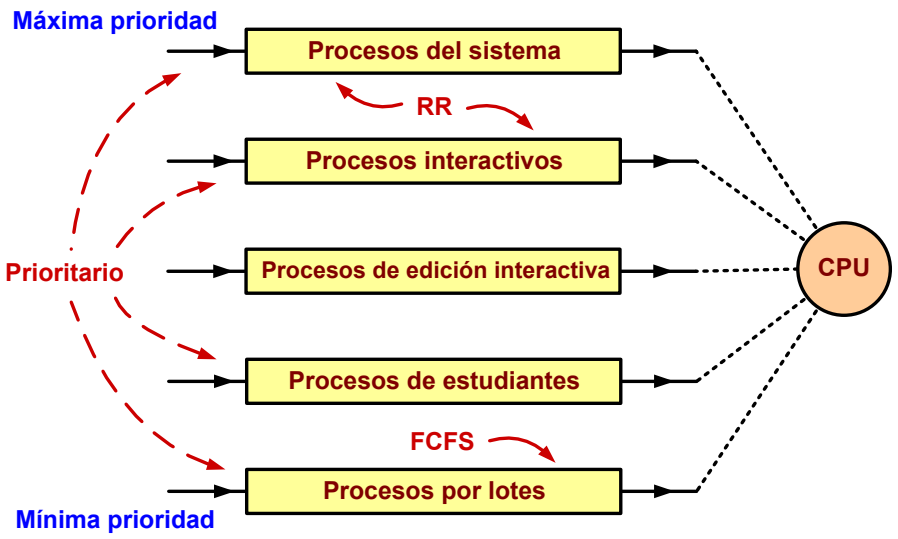
Algoritmos de planificación *(cont.)*

- **Colas Multinivel:**
- Hasta ahora tratamos sistemas con una única cola Ready.
  - Sin embargo, esta no es la solución más apropiada puesto que en los sistemas existen diferentes tipos de procesos (batch, interactivos, de usuario, del sistema) y cada tipo de proceso requiere un tratamiento particular; por lo tanto, debe ser planificado de forma diferente.
  - Una solución adecuada a este requerimiento es contar con varias colas Ready.

Ejemplo: En la figura siguiente se muestra un esquema de planificación con varias colas Ready de acuerdo al tipo de proceso que ingresa al sistema.

23

Diagrama de Colas Ready Multinivel



24

Colas Ready Multinivel

Planificación basada en colas multinivel

Consiste de varias colas sobre las que operan, de manera organizada, un conjunto de algoritmos.

Ejemplo1: un modo de funcionamiento de la planificación en el esquema de la figura anterior podría ser:

1. Un algoritmo ubica los procesos en la cola correspondiente. (p. ej. : los ubica de acuerdo a la prioridad asignada a cada proceso).
2. A su vez, cada cola posee su propio algoritmo de planificación. (p. ej.: la cola con procesos interactivos utiliza RR, mientras que la de procesos por lotes usa FCFS).
3. Los procesos de la cola con una dada prioridad podrán hacer uso de la CPU **siempre que no haya un proceso en una cola de mayor prioridad.**

25

Colas Ready Multinivel

Planificación basada en colas multinivel *(cont.)*

Ejemplo1: *(cont.)*

4. El algoritmo que planifica el proceso que va a hacer uso de la CPU puede ser preemptive:

   Si está haciendo uso del procesador un proceso de una cola con una dada prioridad y llega otro proceso a una cola de mayor prioridad, el primer proceso es retirado del procesador y puesto el proceso recién llegado.

Ventaja del esquema basado en colas multinivel sobre SJF:

Es muy difícil implementar el SJF puro debido a que es imposible, en muchos casos, conocer a priori el tiempo que llevará la ejecución de un proceso.

26

Colas Ready Multinivel con Realimentación

Un esquema basado en colas de Ready multinivel que mejora aún más la performance, es el denominado Colas Multinivel con Realimentación:

Colas Multinivel con Realimentación:

1. Está basado en el esquema de colas multinivel (varias colas Ready sobre las que actúan organizadamente un conjunto de algoritmos).
2. Además, utiliza un **mecanismo dinámico de prioridades** que se provee mediante un mecanismo de realimentación. (Se describe a continuación).

27

Colas Ready Multinivel con Realimentación

Ejemplo2: uso de mecanismo dinámico de prioridades.

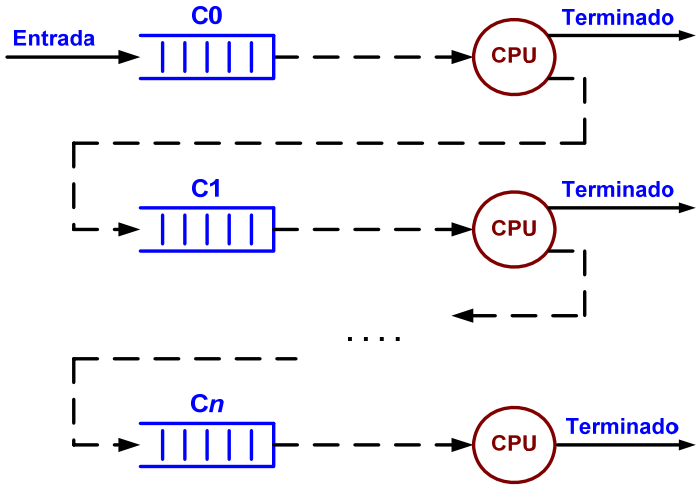
- Cuando un proceso ingresa por primera vez al sistema es puesto en la cola de prioridad C0 (máxima).
- Después de su primera ejecución, si no ha finalizado, el proceso vuelve al estado Ready y es puesto en la cola de prioridad C1 (inferior a C0).
- En resumen: después de cada ejecución, si no ha finalizado, el proceso es puesto en una cola de prioridad inferior a la que estuvo anteriormente.

Por ejemplo, en cada cola puede usarse el algoritmo RR que otorga un quantum mayor a medida que la cola tiene menor prioridad, hasta llegar a la última cola en la que se usa FCFS.

28

Ejemplo de Planificación con Realimentación

Diagrama dinámico en un sistema monoprocesador



El esquema muestra las distintas colas por las que puede pasar un proceso que se ingresa inicialmente en **C0** y necesita, a lo largo de su ejecución, usar la CPU varias veces antes de finalizar.

29

Colas Ready Multinivel con Realimentación

- Este esquema de planificación así concebido, favorece los procesos **nuevos** y **cortos** porque finalizarán rápidamente antes que los viejos y largos. (es una implementación similar a SJF).

  La mayoría de los SO actuales utiliza un esquema similar al presentado, pero cada uno con su propia variante.

  El ejemplo presentado tiene un problema: los procesos largos pueden llegar a padecer inanición.

  Los SO lo salvan implementando un mecanismo que, por ejemplo, cada tanto se indaga en la última cola si hay procesos que no han hecho uso del procesador por un tiempo dado. Si hay, los promocionan a colas de mayor prioridad, aumentándoles, incluso, el quantum.

30

Comparación de Métodos de Planificación

- **Criterios orientados al usuario:**
  Percepción del usuario respecto del comportamiento del sistema.
  Ejemplo: el objetivo del planificador de CPU en un sistema interactivo puede ser **maximizar** el número de procesos cuyo tiempo de respuesta no sea mayor que 2s.
- **Criterios orientados al sistema:**
  El más importante: uso eficiente del procesador.
  Ejemplo: maximizar el throughput de procesos terminados.
  (número de procesos terminados/unidad de tiempo).

31

Parámetros de Evaluación de Algoritmos

- **Utilización del procesador.** (orientado al sistema)
  Lo ideal es mantener la CPU ocupada la mayor cant. de tiempo
- **"Throughput".** (orientado al sistema)
  Mayor cantidad de procesos finalizados por unidad de tiempo.
- **Tiempo de "Turnaround" (retorno).** (orientado al usuario)
  Intervalo de tiempo desde que un proceso ingresa al sistema hasta que lo abandona. Se aplica especialmente a los procesos batch.
  (Suma de T de uso de: CPU + dispos. E/S + espera en colas).
- **Tiempo de Espera.** (orientado al usuario)
  Tiempo de espera del proceso en la cola Ready.
- **Tiempo de Respuesta.** (orientado al usuario)
  Es similar al Tiempo de "Turnaround" pero referido a procesos interactivos: intervalo de tiempo desde que el proceso ingresa al sistema hasta que produce la primera salida de datos.

32

Comportamiento de Algoritmos

Evaluación de algoritmos planificadores según parámetros

- **FCFS:**
  Tiempo de "Turnaround" y Espera. Tiempo de Espera: puede ser bastante alta, puesto que el proceso deberá permanecer en la cola Ready todo el tiempo que demande la ejecución de los procesos que están antes que él.
- **SJF:**
  Tiempo de espera promedio: SJF tiene un comportamiento óptimo porque privilegia los procesos cortos.
- **Prioritario:**
  No se puede decir nada a priori, pues depende de la función utilizada en calcular la prioridad de los procesos.

33

Comportamiento de Algoritmos

Evaluación de algoritmos planificadores según parámetros (*cont.*)

- **Round Robin:**
  Performance íntimamente vinculada con el quantum de tiempo.
  - Si el quantum es muy grande: RR→FCFS.
  - Si es muy pequeño: baja la performance global del sistema por la gran cantidad de cambios de procesos.
- **Colas Multinivel:**
  La evaluación de performance del conjunto de algoritmos usado es bastante compleja de hacer pero, en general, presenta un comportamiento mucho mejor que los algoritmos anteriores.

34

Comportamiento de Algoritmos

	FCFS	SJF	Prioridad	Round Robin	Cola Multinivel
Modo de seleccionar el proceso	Por orden de llegada a la cola de Ready.	Mínimo tiempo para procesar.	Máxima prioridad.	Constante.	Depende de los algoritmos usados.
Modo de decidir el uso de la CPU	Nonpreemptive	Nonpreemptive y preemptive.	Nonpreemptive y preemptive.	Preemptive (cumplido el quantum).	Preemptive.
Throughput (productividad)	No relevante.	Alto.	No relevante. Depende de cómo se asigne	Puede ser baja si el quantum es pequeño.	No relevante.
Tiempo de respuesta (tiempo de 1ª salida de datos)	Puede ser alto. Especialmente si hay procesos CPU bound.	Bueno para procesos cortos.	Bueno para procesos de alta prioridad.	Bueno para procesos cortos.	No relevante.
Overhead (sobrecarga)	Mínimo.	Puede ser alto, sobre todo si es preemptive.	Puede ser alto, sobre todo si es preemptive.	Puede ser alto si el quantum es pequeño.	Puede ser alto.
Efecto en los procesos	Penaliza los procesos cortos y los I/O bound.	Penaliza los procesos largos.	Penaliza los procesos de baja prioridad.	Tratamiento igualitario.	Puede favorecer algunos procesos (según tipo de usuario).

35

Fin del Módulo III

36