

2.20. Describir las principales configuraciones en función del número y tipo de hilos soportados por un sistema operativo.

- ❖ *Múltiples hilos de usuario sin soporte de hilos del núcleo.*
- ❖ *Un hilo del núcleo por cada hilo de usuario.*
- ❖ *Menor número de hilos del núcleo que hilos de usuarios.*

Preguntas de autoevaluación tema 3

3.1. Enumerar y describir brevemente los diferentes tipos de colas de procesos que mantiene un sistema operativo.

- ❖ ***Cola de procesos en el estado preparado.*** En ella se encuentran aquellos procesos que desean acceder al procesador para iniciar o continuar su ejecución. No tiene por qué ser única.
- ❖ ***Cola de procesos en el estado preparado en memoria secundaria.*** Contiene procesos que están a la espera de regresar a la memoria principal al estado preparado.
- ❖ ***Cola de procesos en el estado bloqueado.*** Los procesos que pertenecen a esta cola están a la espera de poder volver al estado preparado tan pronto ocurra el evento por el que han entrado en el estado bloqueado. No tiene por qué ser única.
- ❖ ***Cola de procesos en el estado bloqueado en memoria secundaria.*** Contiene procesos que están a la espera de pasar al estado preparado en memoria secundaria, si ocurre el evento por el que entraron en el estado bloqueado.
- ❖ ***Cola de trabajos por lotes o cola de entrada.*** Se almacena en memoria secundaria y contiene los nuevos trabajos que llegan a un sistema operativo por lotes o a la parte por lotes de un sistema operativo híbrido. Cada trabajo se encuentra a la espera de que el sistema operativo cree un nuevo proceso para atenderlo.

3.2. ¿En qué consiste la actividad del sistema operativo conocida como planificación de procesos?

Un proceso durante su existencia puede pasar por varias colas. El sistema operativo debe determinar cuándo un proceso abandona una cola para acceder a un recurso o para ingresar en otra cola.

¿Qué niveles de planificación se distinguen?

- ❖ ***Planificación a corto plazo.*** Decide qué proceso en la cola de preparados será ejecutado a continuación en el procesador.

- ❖ **Planificación a medio plazo.** Decide qué proceso en una cola de memoria principal es intercambiado a una cola de memoria secundaria, o viceversa.
- ❖ **Planificación a largo plazo.** Decide qué trabajo de la cola de trabajo por lotes pasa a ser ejecutado en el sistema mediante la creación de un proceso.

3.3. ¿Qué función realiza el planificador a corto plazo?

La elección del proceso perteneciente a la cola de procesos en el estado preparado que pasará al estado ejecutándose.

¿Cuáles son sus componentes?

- ❖ **Encolador (enqueueer).** Cuando un proceso entra en el estado preparado, el encolador se encarga de incluir al proceso en la cola de procesos preparados mediante la configuración de los punteros necesarios en el bloque de control del proceso.
- ❖ **Conmutador de contexto (context switcher).** Se encarga de guardar el contexto del proceso que va a ser desalojado del procesador y cargar el contexto del proceso que ha sido planificado para ser ejecutado.
- ❖ **Distribuidor o despachador (dispatcher).** Se encarga de seleccionar un proceso de la cola de procesos preparados de acuerdo con un determinado algoritmo de planificación. También se encarga de cederle el control del procesador.

3.4. Enumerar algunas circunstancias que requieran la invocación del planificador a corto plazo.

- ❖ **Cuando se ha terminado de crear un nuevo proceso hijo.** En este caso el planificador debe decidir si continúa ejecutando el proceso padre o comienza la ejecución del proceso hijo.
- ❖ **Cuando un proceso entra en el estado bloqueado en espera de que se produzca algún evento.** En este caso dicho proceso ya no requiere utilizar el procesador por lo que el planificador debe decidir qué proceso ejecutar.
- ❖ **Cuando se termina de atender una interrupción.** Por ejemplo, si llega una interrupción para avisar que se ha completado una operación de E/S con el disco por la que estaba esperando en el estado bloqueado un cierto proceso.
- ❖ **Cuando un proceso finaliza.** Si un proceso ha entrado en el estado terminado obviamente ya no utilizará el procesador por lo que el planificador debe decidir qué proceso será ejecutado a continuación.

3.5. ¿Qué tareas realiza el planificador a medio plazo?

Se encarga de decidir qué procesos en memoria principal de los pertenecientes a la cola de procesos en el estado preparado o la cola de procesos en el estado bloqueado serán intercambiados a memoria secundaria, pasando a formar parte de la cola de procesos preparados memoria secundaria o de la cola de procesos bloqueados en memoria secundaria, respectivamente. También se encarga de realizar la decisión contraria, es decir, qué procesos en las colas de memoria secundaria pasarán a las colas de memoria principal.

3.6. ¿Qué tareas realiza el planificador a largo plazo?

Se encarga de decidir qué trabajo de los existentes en la cola de trabajos por lotes será admitido en el sistema para ser ejecutado. Además, debe decidir cuándo se pueden admitir nuevos trabajos. La admisión de un trabajo consiste en la creación de un proceso para el procesamiento de dicho trabajo, el cual pasa a la cola de procesos en el estado preparado o la cola de procesos en el estado preparado en memoria secundaria.

3.7. ¿Qué es un proceso limitado por CPU?

Un proceso que emplea la mayor parte de su tiempo haciendo cálculos en el procesador.

¿Y un proceso limitado por E/S?

Un proceso que emplea la mayor parte de su tiempo esperando por la realización de operaciones de E/S.

3.8. ¿Cuales son los objetivos principales que debe cumplir la función de planificación de procesos?

- ❖ **Equidad.** En ausencia de un esquema de prioridades preestablecido, los procesos con requisitos similares deben obtener un uso de los recursos similares. Es decir, ningún proceso debe sufrir inanición.
- ❖ **Previsibilidad.** Trabajos con características similares deben ejecutarse en tiempos similares usando recursos similares.
- ❖ **Equilibrado de los recursos.** Se debe intentar mantener ocupados todos los recursos del computador. Para cumplir con este objetivo se debe procurar que los programas cargados en memoria sean una mezcla de programas limitados por CPU y programas limitados por E/ S.
- ❖ **Proporcionalidad.** Las peticiones que son percibidas por el usuario como sencillas deben emplear menos tiempo de

respuesta que las complejas ya que de lo contrario el usuario se irrita o considera que el sistema se ha colgado.

3.9. Definir los siguientes criterios de planificación del procesador: utilización del procesador o eficacia, productividad, tiempo de entrega o tiempo de estancia, tiempo de espera, tiempo de respuesta y plazo de finalización.

- ❖ **Utilización del procesador o eficacia.** Fijado un cierto tiempo de observación, hace referencia al tanto por ciento de dicho tiempo que el procesador ha estado activo.
- ❖ **Productividad (throughput) o rendimiento.** Es el número de trabajos completados por unidad de tiempo (típicamente una hora). Esta medida depende obviamente de la longitud de los procesos.
- ❖ **Tiempo de entrega (turnaround time), también denominado tiempo de retorno o tiempo de estancia.** Es el tiempo medio que transcurre desde que se lanza un proceso hasta que finaliza. Se calcula como la suma del tiempo de ejecución y el tiempo de espera por los recursos, incluyendo el procesador.
- ❖ **Tiempo de espera.** Es la suma de los tiempos que un proceso pasa esperando en las diferentes colas del sistema por la obtención de recursos.
- ❖ **Tiempo de respuesta (response time).** Es el tiempo transcurrido desde que un usuario manda una orden desde su terminal hasta que comienza a recibir la respuesta. Obviamente se encuentra limitado por la velocidad de los dispositivos de E/S.
- ❖ **Plazo de finalización (deadline).** Hace referencia al tiempo máximo preestablecido que un proceso tiene para ser completado en un sistema de tiempo real.

3.10. Señalar los criterios principales considerados en la planificación del procesador dependiendo del tipo de sistema operativo.

- ❖ ***El planificador de un sistema por lotes debe intentar maximizar la utilización de procesador y la productividad. Además debe intentar minimizar el tiempo de entrega.***
- ❖ ***El planificador de un sistema de tiempo compartido debe intentar minimizar el tiempo de respuesta.***
- ❖ ***El planificador de un sistema de tiempo real debe intentar maximizar el número de plazos de finalización cumplidos.***

3.11. Definir planificación expropiativa y no expropiativa. Señalar sus principales diferencias y ventajas.

La planificación del procesador es *no expropiativa* (nonpreemptive) si permite que un proceso pueda estar ejecutándose en el procesador ininterrumpidamente hasta que termine o se bloquee en espera de un evento.

La planificación del procesador es *expropiativa* (preemptive) si el proceso que se está ejecutando en el procesador puede ser interrumpido en cualquier momento y pasado al estado preparado para proceder a ejecutar otro proceso distinto.

Una planificación expropiativa produce una mayor sobrecarga al sistema que una no expropiativa, que se ejecuta con mayor frecuencia el planificador y se realizan más cambios de proceso; pero proporciona un mejor servicio a la población de procesos ya que previene que un proceso monopolice del procesador.

3.12. ¿Qué es un cuanto de ejecución?

El tiempo máximo durante el cual un proceso se ejecuta ininterrumpidamente.

3.13. ¿Qué tipo de planificación, expropiativa o no expropiativa, es más recomendable para un sistema operativo por lotes?

La no expropiativa, o la expropiativa pero con un cuanto largo.

¿Y para un sistema de tiempo compartido?

La expropiativa con objeto de atender las peticiones de todos los usuarios.

3.14. Describir el algoritmo de planificación del primero en llegar primero en ser servido, (FCFS) (First-Come First-Served).

Este algoritmo cede el uso del procesador al primer proceso que lo solicite, éste puede utilizarlo ininterrumpidamente hasta que termine o se bloquee en espera de un evento.

3.15. Describir el algoritmo de planificación de primero el proceso más corto, SJF (Shortest Jop First).

Es un algoritmo de tipo no expropiativo que selecciona para ser ejecutado al proceso con tiempo de procesamiento más corto. Si dos o más procesos tienen el mismo tiempo de procesamiento entonces se les aplica el algoritmo FCFS.

3.16. Describir el algoritmo de planificación del proceso con menor tiempo restante, SRT (Shortest Remaining Time Next).

Siempre selecciona para ser ejecutado al proceso con menor tiempo restante de ejecución. Si mientras se está ejecutando en el

procesador un proceso A llega a la cola de preparados un proceso B con menor tiempo restante de ejecución, el planificador expropia el procesador al proceso A y se lo cede al proceso B.

3.17. Describir el algoritmo de planificación de turno rotatorio (*round robin*).

Asigna el uso ininterrumpido del procesador a o durante una cantidad fija de tiempo denominada *cuanto* (quantum) o *rodaja de tiempo* (time slice). Finalizado el cuanto, se genera una interrupción de reloj que produce que la ejecución del proceso sea interrumpida y se pase a ejecutar el primer proceso de la cola de preparados, como en la planificación. El proceso interrumpido se coloca al final de dicha cola y tendrá que esperar turno volver a utilizar el procesador.

Si un proceso termina de usar el procesador antes de consumir su cuanto, entonces se planifica el primer proceso de la cola preparados sin esperar a que finalice el cuanto. Por otra parte, si finaliza un cuanto y no existen procesos en la cola de preparados, entonces el proceso que se estaba ejecutando puede seguir usando el procesador.

3.18. Describir el algoritmo de planificación basada en prioridades.

Asigna a cada proceso en el momento de su creación una determinada *prioridad de ejecución* en la forma de un número entero positivo. Dependiendo del sistema, la máxima prioridad puede corresponder a un número pequeño, típicamente 0, o a un número grande. Siempre se planifica para ejecución al proceso de mayor prioridad. En caso de que varios procesos posean la misma prioridad éstos se deben planificar mediante otro algoritmo distinto, como por ejemplo, FCFS o SJF.

3.19. Describir el algoritmo de planificación basada en múltiples colas de prioridad.

Implementa varias colas de procesos preparados o *colas de prioridad*, cada una de las cuales solo pueda contener procesos con una determinada prioridad o dentro rango de prioridades.

Siempre se ejecuta un proceso perteneciente a la cola de preparados con mayor prioridad. Sólo cuando dicha cola está vacía se puede planificar un proceso perteneciente a una cola de preparados de menor prioridad. Un proceso a lo largo de su vida solo puede pertenecer a una determinada cola de preparados.

Además para planificar cada cola se puede utilizar un algoritmo distinto: FCFS, SJF, rotatorio, etc.

Puede producir la inanición de los procesos pertenecientes a las colas de menor prioridad si las colas de mayor prioridad nunca se quedan vacías. Para evitar este problema se puede asignar procesos de cada cola un determinado porcentaje de tiempo de uso del procesador.

3.20. Describir el algoritmo de planificación basada en múltiples colas de prioridad y realimentación.

Un proceso solo puede pertenecer a una determinada cola de preparados o cola de prioridad durante su tiempo de vida. En consecuencia, posee una prioridad estática. Esta restricción simplifica la implementación del algoritmo pero también lo hace poco flexible. Si se levanta esta restricción y se permite que un proceso pueda ir cambiando de cola de prioridad durante su existencia, es decir, que su prioridad pueda ser modificada dinámicamente, se dice que el algoritmo posee *realimentación*.

3.21. Describir el algoritmo de planificación por tiempo límite.

En los sistemas de tiempo real estrictos el tiempo de servicio de algunos procesos deben completarse en un determinado *plazo* (deadline) o *tiempo límite*.

Para lograr este objetivo el planificador solo admite un proceso en la cola de procesos preparados si es capaz de garantizarle que su tiempo de servicio se atenderá dentro de su tiempo límite, teniendo en cuenta los tiempos de servicio y plazos de los procesos ya existentes en dicha cola. En consecuencia el planificador de estos sistemas debe conocer por adelantado para cada proceso su tiempo de servicio y su tiempo límite de ejecución.

3.22. Describir la planificación de hilos en función del tipo de hilo soportado por el sistema operativo.

Si entre se soportan hilos a nivel de usuario, el sistema operativo no es consciente de la existencia de estos hilos y por ello realiza una planificación global a nivel de proceso de acuerdo con un determinado algoritmo de planificación.

Una vez que un proceso está siendo ejecutado en modo usuario es el hilo de planificación dentro de dicho proceso el que decide qué hilo va a ser ejecutado. Por lo tanto cada proceso puede planificar sus hilos con el algoritmo de planificación que considere más oportuno.

Puesto que en modo usuario no se pueden tratar las interrupciones, no existen interrupciones de reloj disponibles para establecer el cuanto de un hilo. Éste se ejecuta hasta que voluntariamente decide ceder el uso del procesador o expira el cuanto del proceso. En dicho caso el sistema operativo debe seleccionar otro proceso para ser ejecutado.

Si se soportan hilos del núcleo, el sistema operativo planifica estos hilos usando algún determinado algoritmo de planificación. El planificador puede tener en cuenta consideraciones relativas al rendimiento del sistema a la hora de planificar hilos del núcleo de igual prioridad.

En sistemas con una configuración del tipo menor número de hilos del núcleo que hilos de usuarios, aparte de las dos planificaciones comentadas, es necesario implementar con la biblioteca de planificación local para establecer qué hilo de usuario puede usar un hilo del núcleo.

Preguntas de autoevaluación tema 4

4.1. Explicar cuándo se produce y en qué consiste el problema de la condición de carrera.

Se producen cuando varios procesos acceden a un recurso compartido, con lo cual en vez de cooperar compiten por el ya que el resultado final de la ejecución depende del orden en que se hayan planificado los procesos, es en que se hayan ejecutado las instrucciones de lectura y escritura de cada proceso sobre el recurso.

Surge cuando múltiples procesos independientes se ejecutan concurrentemente y acceden para leer o escribir en un recurso compartido.

4.2. ¿Qué es una sección crítica?

Una instrucción o conjunto de instrucciones secuenciales del código de un proceso que requieren manipular un recurso compartido (variable, dato, fichero, ...) con otros procesos. Dentro del código de un proceso pueden existir varias secciones críticas.

4.3. ¿En qué consiste la exclusión mutua?

Que el uso de un recurso por parte de un proceso excluya su uso para los restantes.

¿Qué problema permite resolver?

Las condiciones de carrera.