

Asignatura: PSP

Fecha: 29-09-24

1. ¿Qué estados puede tener un proceso? ¿Qué se debe cumplir para que pase de un estado a otro? Puedes ayudarte de un dibujo.

En la imagen, se muestran los siguientes estados de un proceso en un sistema operativo:

Nuevo: El proceso ha sido creado.

Listo: El proceso está listo para ser ejecutado si se le asigna un procesador.

Ejecutado: El proceso está siendo ejecutado.

Bloqueado: El proceso necesita recursos o realiza solicitudes de entrada/salida.

Finalizado: El proceso ha completado su ejecución.

Para que un proceso pase de un estado a otro, se deben cumplir las siguientes condiciones:

Nuevo a Listo: El proceso está creado y listo para ejecutarse.

Listo a Ejecutado: Se asigna un procesador al proceso.

Ejecutado a Listo: El proceso completa un ciclo de ejecución pero no ha terminado.

Ejecutado a Bloqueado: El proceso necesita recursos o realiza solicitudes de E/S.

Bloqueado a Listo: Se completan las solicitudes de recursos o E/S.

Ejecutado a Finalizado: El proceso completa toda su ejecución.



2. Define los 3 tipos de planificadores que hemos visto en clase

Planificador de Alto Nivel: También conocido como planificador de largo plazo, decide qué procesos se admiten en el sistema desde la cola de nuevos procesos. Controla el grado de multiprogramación, es decir, el número de procesos en la memoria principal.
Ejemplo: Decidir si un nuevo proceso debe ser admitido en el sistema.

Planificador de Medio Nivel: Gestiona la memoria y puede suspender o reanudar procesos. Es responsable de la paginación y el intercambio de procesos entre la memoria principal y el almacenamiento secundario. Su objetivo es optimizar el uso de la memoria y mejorar el rendimiento del sistema.
Ejemplo: Mover un proceso de la memoria principal al disco (swap out) y viceversa (swap in).

Planificador de Bajo Nivel: También conocido como planificador de corto plazo, decide qué proceso en la cola de listos debe ejecutarse a continuación. Asigna el procesador a uno de los procesos listos para ejecutar.
Ejemplo: Seleccionar el siguiente proceso que debe ejecutarse en la CPU.

3. ¿Comenta al menos 5 algoritmos de planificación? Elige un algoritmo y comenta como funcionaría el sistema si le llegan 3 procesos con un tiempo de ejecución aproximado de 3,15 y 9 ms

Round Robin: Este algoritmo asigna un tiempo fijo (quantum) a cada proceso en la cola de listos. Cuando un proceso agota su quantum, es enviado al final de la cola y el procesador se asigna al siguiente proceso.

Ventajas: Es justo y sencillo de implementar. Garantiza que todos los procesos reciban tiempo de CPU.

Desventajas: Puede tener un alto overhead debido a los frecuentes cambios de contexto, especialmente si el quantum es muy pequeño.

First In, First Out: También conocido como First-Come, First-Served este algoritmo asigna la CPU al proceso que llega primero a la cola de listos.

Ventajas: Es simple y fácil de entender.

Desventajas: Puede causar el problema de convoy effect, donde procesos largos retrasan a los procesos más cortos.

Shortest Job First: Este algoritmo selecciona el proceso con el menor tiempo de ejecución estimado.

Ventajas: Minimiza el tiempo de espera promedio.

Desventajas: Requiere conocimiento previo del tiempo de ejecución de los procesos, lo cual no siempre es posible.

Shortest Remaining Time First: Variante preventiva del SJF. El proceso con el menor tiempo de ejecución restante es el que se ejecuta.

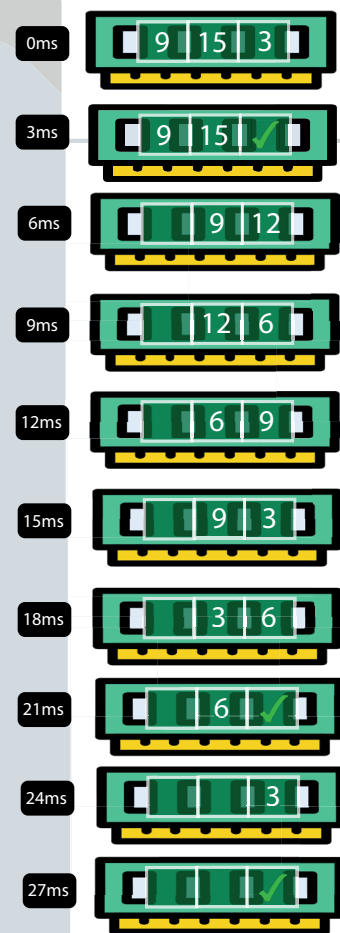
Ventajas: También minimiza el tiempo de espera promedio y es más eficiente en sistemas donde los tiempos de ejecución pueden ser estimados con precisión.

Desventajas: Similar a SJF, requiere conocimiento previo del tiempo de ejecución.

Múltiples Colas: Este algoritmo divide los procesos en diferentes colas basadas en su prioridad o tipo (por ejemplo, procesos interactivos vs. procesos en segundo plano). Cada cola puede tener su propio algoritmo de planificación.

Ventajas: Permite una mayor flexibilidad y puede ser ajustado para diferentes tipos de procesos.

Desventajas: Puede ser complejo de implementar y ajustar correctamente.



4. ¿Qué ventajas tiene cambiar entre hilos de un mismo proceso frente a cambiar entre procesos diferentes?

Cambiar entre hilo de un mismo proceso tiene varias ventajas:

Menor Overhead: Cambiar entre hilos es más rápido y eficiente porque comparten el mismo espacio de direcciones y recursos del proceso.

Compartición de Recursos: Los hilos de un mismo proceso pueden compartir datos y recursos fácilmente.

Facilidad de Programación: La programación con hilos es más sencilla en términos de compartir datos y recursos, ya que todos los hilos dentro de un proceso pueden acceder a las mismas variables y estructuras de datos.

5. Investiga en los apuntes para dar una descripción de “servicio”.

Un servicio es un tipo especial de proceso que generalmente se carga durante el arranque del sistema operativo. Su principal característica es que permanece en espera, listo para realizar tareas cuando otro proceso o usuario lo solicita. Los servicios son fundamentales para el funcionamiento continuo y eficiente del sistema, ya que manejan tareas de fondo sin intervención directa del usuario. Un ejemplo común de servicio es un servidor web, que espera solicitudes de los navegadores para entregar páginas web. Otro ejemplo es un servicio de impresión, que gestiona las solicitudes de impresión enviadas por diferentes aplicaciones.
