



Hoja de Trabajo – CPU Scheduling

1. Explique cuál es la diferencia entre Scheduling Permisivo y No Permisivo.
2. ¿Cuál de los siguientes algoritmos de Scheduling podría provocar un bloqueo indefinido? Explique su respuesta.
 - a. First-come, first-served
 - b. Shortest job first
 - c. Round robin
 - d. Priority
3. De estos dos tipos de programas:
 - a. I/O-bound (un programa que tiene más I/Os que uso de CPU)
 - b. CPU-bound (un programa que tiene más uso de CPU que I/Os)¿Cuál tiene más probabilidades de tener cambios de contexto voluntarios y cuál tiene más probabilidades de tener cambios de contexto no voluntarios? Explica tu respuesta.
4. Utilizando un sistema Linux, escriba un programa en C que cree un proceso hijo (fork) que finalmente se convierta en un proceso zombie. Este proceso zombie debe permanecer en el sistema durante al menos 10 segundos.
Los estados del proceso se pueden obtener del comando: `ps -l`

1. Diferencia entre Scheduling Permisivo y No Permisivo:

- **Scheduling Permisivo:** En este enfoque, un proceso en ejecución puede ser interrumpido por otro proceso de mayor prioridad. Esto significa que el sistema operativo permite que un proceso en ejecución sea desalojado antes de que haya completado su tiempo de CPU asignado. Esto puede conducir a cambios de contexto más frecuentes.
- **Scheduling No Permisivo:** Aquí, un proceso en ejecución no puede ser interrumpido por otro proceso de prioridad mayor. El proceso en ejecución debe ceder la CPU voluntariamente, por ejemplo, al realizar una operación de E/S o al finalizar su ejecución. Esto puede llevar a una planificación más predecible, pero también podría resultar en situaciones donde procesos de alta prioridad tengan que esperar más tiempo para obtener acceso a la CPU.

2. Algoritmo de Scheduling que podría provocar un bloqueo indefinido:

- **Respuesta correcta:** d. Priority
- **Explicación:** En el algoritmo de prioridad, si un proceso con alta prioridad nunca se bloquea o finaliza, los procesos con prioridades más bajas nunca tendrán la oportunidad de ejecutarse. Esto podría llevar a un bloqueo indefinido de los procesos con prioridades más bajas.

3. Probabilidades de cambios de contexto voluntarios y no voluntarios para programas I/O-bound y CPU-bound:

- **I/O-bound:** Tiene más probabilidades de tener cambios de contexto voluntarios. Esto se debe a que los programas I/O-bound realizan muchas operaciones de E/S y pueden ceder la CPU voluntariamente mientras esperan que se completen estas operaciones.
- **CPU-bound:** Tiene más probabilidades de tener cambios de contexto no voluntarios. Estos programas utilizan la CPU intensivamente y generalmente no tienen muchas oportunidades de cederla voluntariamente. Por lo tanto, es más probable que ocurran cambios de contexto cuando el sistema operativo decide desalojar el proceso para permitir que otros procesos se ejecuten.

