Administración de Procesos y del Procesador

Mecanismos de Sincronización Entrada/Salida

Es la parte del código donde acceden procesos concurrentes.

Soluciones:

- Mutex (Exclusión Mutua): Impide que más de un proceso entre a la sección crítica.
- Semáforos: Contador que gestiona los accesos a la sección crítica (wait y signal).
- Monitores: Mecanismos de sincronización de alto nivel para controlar el acceso.

Planificación de Procesos en SO

Se encarga de asignar recursos de CPU a los procesos de forma eficiente.

Tipos de Planificación:

- Planificación a Corto Plazo: Selecciona procesos de la cola de listos y los asigna al procesador.
- Planificación a Mediano Plazo:
 Decide qué procesos deben permanecer en memoria principal.
- Planificación a Largo Plazo: Controla la admisión de nuevos procesos en el sistema.

Algoritmo SJF (Shortest Job First)

Planificación que selecciona el proceso con el menor tiempo de ejecución.

Propiedades:

- Preemptivo o No Preemptivo: Interrumpe o no interrumpe los procesos en ejecución.
- Ventajas: Minimiza el tiempo promedio de espera.
- Desventajas: Dificultad para estimar el tiempo de ejecución de cada proceso.

Parámetros del Algoritmo Multilevel Feedback Queues

Es un algoritmo de planificación que organiza múltiples colas de prioridades.

Parámetros:

- Número de Colas: Cada cola tiene su propia política de planificación.
- Criterios de Ascenso/Descenso: Basado en el comportamiento del proceso (consumo de CPU).
- Quantum de CPU: La cantidad de tiempo asignada a cada cola.

Utilización:

- Evitar la concurrencia no controlada: Se utiliza para asegurar que dos procesos no accedan simultáneamente a una sección crítica, evitando que ocurran errores o condiciones de carrera.
- Controlar el acceso:
 Implementa un mecanismo para controlar el acceso a recursos compartidos entre dos procesos, asegurando que solo uno esté en la sección crítica a la vez.

Algoritmo de Dekker

Fue el primer algoritmo que resolvió el problema de exclusión mutua en sistemas concurrentes, permitiendo que dos procesos no accedan simultáneamente a la sección crítica.

Características:

- Usa variables de estado para indicar la intención de los procesos de acceder a la sección crítica.
- Turno alternado: Si ambos procesos desean entrar a la sección crítica, uno cede el paso, garantizando que sólo uno entre a la vez.

Algoritmo de Peterson

Es una mejora del algoritmo de Dekker, más simple y eficiente, usado para resolver la exclusión mutua entre dos procesos.

Características:

- Utiliza dos variables: una indica si un proceso quiere acceder a la sección crítica, y la otra guarda el turno del proceso.
- Sencillez: Su estructura es más directa y fácil de implementar.
- Eficiencia: Al igual que Dekker, garantiza que un solo proceso acceda a la sección crítica a la vez, pero con menos sobrecarga y mejor rendimiento.

Utilización:

- Sincronización más eficiente:
 Al igual que Dekker, se utiliza
 para resolver el problema de
 exclusión mutua, pero de una
 manera más sencilla y
 eficiente, permitiendo que
 dos procesos se alternen para
 acceder a la sección crítica.
- Evitar conflictos en la
 ejecución concurrente: El
 algoritmo garantiza que dos
 procesos no accedan
 simultáneamente a la sección
 crítica, pero con menos
 sobrecarga en su
 implementación.