Sincronización entre procesos

Fundamentos

Los procesos cooperativos pueden compartir directamente un espacio de direcciones lógico o compartir datos . El acceso a estos datos puede llegar a tener incoherencias, es por eso que existe la llamada SINCRONIZACIÓN DE PROCESOS

Sección Crítica

Consideremos un sistema, el cual conste de n procesos. Cada uno de estos procesos tiene un segmento de su código, denominado, selección crítica, en el cual, ese proceso tiene la posibilidad de modificar variables comunes, actualizar alguna tabla, incluso escribir en algún archivo

Problema de la selección crítica

Ahora bien este problema, consiste en diseñar un protocolo que los procesos puedan utilizar para trabajar de esta forma. Cada uno de ellos deberá solicitar permiso para ingresar en su sección crítica.

Cada solución a este problema deberá satisfacer 3 requisitos:

1. Exclusión mutua: No puede haber más de 1 proceso ejecutándose en su sección crítica, a la vez.
2. Progreso: Si ningún proceso se ejecuta en su sección crítica y otros lo necesitan, serán los siguientes.
3. Espera Limitada: Existe un límite en el número de veces que se permite que otros procesos entren en sus secciones.

Solución de Peterson

- Proporciona una buna descripción algorítmica de la resolución de este problema.
- Se restringe a 2 procesos que van alternando la ejecución de su sección crítica y su sección restante
- Requiere que los dos procesos compartan dos elementos de datos

Hardware de sincronización

Existen algunas instrucciones hardware que están disponibles en múltiples sistemas las cuales se pueden implementar para la resolución del problema de la selección crítica. Por ejemplo:

- **TestAndSet: Sirve para leer y modificar atómicamente una variable.**
- **Swap: Sirve para intercambiar el valor de dos variables .**

Semáforos

Es una variable entera a la que solo se accede mediante dos operaciones atómicas estándar wait() y signal(). Se pueden dividir en dos categorías (Contador y Binario). Actualmente el Binario podría ayudar a resolver el problema mencionado anteriormente. Ya que los n procesos comparten un solo semáforo llamado *"mutex"* inicializado con el valor de 1

Monitores

Se tratan de una colección de procedimientos, variables y estructuras de datos que se agrupan en un módulo. Los procesos pueden invocar a estos procedimientos en el momento que quieran, pero no se les permite acceder a las estructuras de datos internas del monitor.