

Equipo: Equipo

Integrantes: Barajas Ruíz Leslie, Benites Onofre Fernando, Chaparro Sicardo Tanibeth, Lerín Hernández Natalia.

MONITORES

Glosario de temas revisados

Concurrencia

La concurrencia se refiere a más de un proceso ocurriendo en orden no *determinista* y cuya ejecución está vinculada de alguna manera, por ejemplo al compartir cierta información o tener alguna dependencia uno de otro. Algunos procesos requieren esta interacción, ya que parte del procesamiento puede depender de datos obtenidos en fuentes externas y la cooperación con hilos o procesos externos es fundamental [1].

La concurrencia comprende un gran número de cuestiones de diseño, incluyendo la comunicación entre procesos, compartición y competencia por los recursos, sincronización de la ejecución de varios procesos y asignación del tiempo de procesador a los procesos [3].

Operación atómica

De “indivisible”, se le llama así a la manipulación de datos que requiere la garantía de que se ejecutará como una sola unidad de ejecución, o fallará por completo, sin resultados o estados parciales observables por otros procesos o el entorno [1]. Se denomina así a una unidad de trabajo ininterrumpible [2]. Ejemplos son las primitivas u operaciones estándar *wait* y *signal* [2,3].

Condición de carrera

(Race Condition)

Categoría de errores de programación que involucra a dos procesos que fallan al comunicarse su estado mutuo, llevando a resultados inconsistentes. Es uno de los problemas más frecuentes y difíciles de depurar y ocurre típicamente por no considerar la *no atomicidad* de una operación [1]. Se le denomina así a una situación en la que varios procesos manipulan y acceden a los mismos datos

concurrentemente y el resultado de la ejecución depende del orden concreto en que se produzcan los accesos [2].

Competencia entre procesos por los recursos

Los procesos concurrentes entran en conflicto cuando compiten por el uso del mismo recurso; estos no son conscientes de la existencia de los otros y cada uno debe dejar tal y como esté el estado de cualquier recurso que utilice. La ejecución de un proceso puede influir en el comportamiento de los procesos que compiten; si dos procesos desean acceder a un único recurso, el sistema operativo le asignará el recurso a uno de ellos mientras el otro espera. Se deben solucionar tres problemas de control cuando hay procesos en competencia. Se requiere exclusión mutua, que lleva a problemas de control adicionales: interbloqueo e inanición.

Sección crítica

Es el segmento de código en el que un proceso (en un sistema que consta de n procesos) puede modificar variables comunes, actualizar una tabla, escribir en un archivo, etc [2]. Esta área requiere ser protegida de accesos simultáneos donde se realiza la modificación de datos compartidos [1]. Dos procesos no pueden ejecutar su sección crítica al mismo tiempo [2].

Recurso compartido

Es aquel al que se puede tener acceso desde más de un proceso. Suele tratarse de una variable en memoria, pero podrían ser de otro tipo de datos compartidos como archivos o periféricos [1].

-Exclusión mutua

Se refiere a que dos procesos no pueden ejecutar su sección crítica simultáneamente. La exclusión mutua de sección crítica asegura que éstas se ejecuten atómicamente [2].

Es importante que solo un programa pueda acceder a su sección crítica en un momento dado. No se puede confiar solamente en el sistema operativo para aceptar y hacer cumplir esta restricción; los requisitos específicos pueden no ser tan obvios y las líneas de los procesos en competencia podrían intercalarse [3].

Los algoritmos de exclusión mutua fueron tratados por primera vez en los clásicos documentos de Dijkstra [2].

Interbloqueo

(deadlock) Se trata de la situación en la cual dos o más procesos están esperando indefinidamente a que se produzca un suceso que sólo puede producirse como consecuencia de las operaciones efectuadas por otro de los procesos en espera [2]. Como ejemplo se consideran dos procesos, P1 y P2, y dos recursos R1, y R2, tales que cada proceso necesita acceder a ambos recursos para llevar a cabo una parte de su función. Podría suceder que el sistema operativo asigne R1 a P1 y R2 a P2. Cada proceso estará esperando a uno de los dos recursos pero ninguno liberará el que ya posee hasta que adquiera el otro y ejecute su sección crítica, por lo que se encuentran interbloqueados [3]. En esta situación el sistema puede seguir operando normalmente, pero ninguno de los procesos involucrados podrán avanzar [1].

Inanición

Una condición en la que un proceso se retarda indefinidamente porque otros procesos siempre tienen la preferencia [1]. Es la situación en que un proceso no puede avanzar en su ejecución dado que necesita recursos que están asignados a otros procesos [1]; es un bloqueo en el cual algunos procesos esperan de forma indefinida [2].

-Progreso

Si ningún proceso está ejecutando su sección crítica y algunos procesos desean entrar en sus correspondientes secciones críticas, sólo aquellos procesos que no estén ejecutando sus secciones restantes pueden participar en la decisión de cuál será el siguiente que entre en su sección crítica, y esta selección no se puede posponer indefinidamente [2].

-Espera limitada

Existe un límite en el número de veces que se permite que otros procesos entren en sus secciones críticas después de que un proceso haya hecho una solicitud para entrar en su sección crítica y antes de que la misma haya sido concedida [2].

Espera activa

Ejecución repetida de un bucle de código mientras se espera a que se produzca un suceso.

(Spinlocks) Un proceso puede consumir mucho tiempo de procesador sólo para esperar que otro proceso cambie una bandera, lo cual puede resultar dañino para el desempeño global de un sistema con manejo de prioridades [1].

Mecanismos de sincronización

Son los que pueden emplearse para programar considerando a la concurrencia: candados, semáforos, variables de condición, monitores, memoria transaccional, otros [1].

Candado

En general, se puede afirmar que cualquier solución al problema de la sección crítica requiere una herramienta simple llamada candado o *cerrojo*. Las condiciones de carrera se evitan requiriendo que las secciones críticas se protejan mediante candados: adquirir un candado antes de entrar en una y liberarlo cuando salga de ella.

Bibliografía

[1] [FUNDAMENTOS DE SISTEMAS OPERATIVOS](#), G. Wolff, et. al., UNAM.

[2] [Fundamentos de Sistemas Operativos, 7ma Edición](#), Silberschatz, et. al. McGraw Hill.

[3] [SISTEMAS OPERATIVOS](#), Stallings. Pearson Educación.