



Seminario de Sistemas Operativos

Departamento de Ciencias Computacionales
Centro Universitario de Ciencias Exactas e Ingenierías

Universidad de Guadalajara

Violeta del Rocío Becerra Velázquez

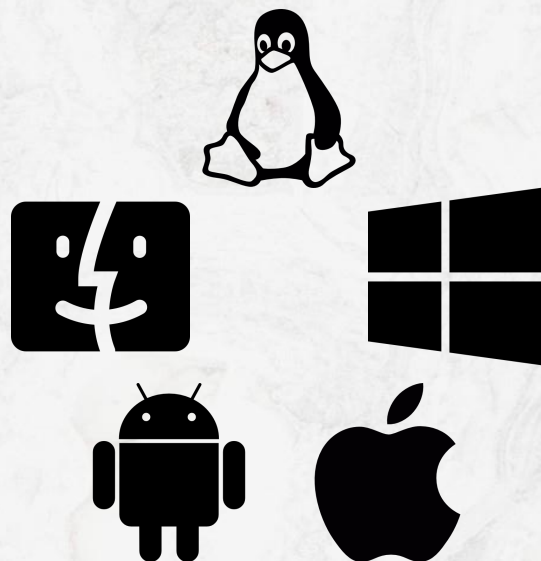
Saul Alejandro Castañeda Pérez

Actividad de Aprendizaje 9

13:00 - 14:55

217564323

D02



Índice

| | |
|--------------------------------------|----------|
| La cena de los filósofos | 3 |
| Productor-Consumidor | 4 |
| Lectores-Escritores | 4 |
| Preguntas | 5 |
| 1. Problema de la concurrencia | 5 |
| 2. Procesos concurrentes cooperantes | 5 |
| 3. Exclusión mutua | 5 |
| 4. Interbloqueo. | 5 |
| 5. Inanición. | 5 |
| 6. Defina Excesiva Cortesía. | 5 |
| 7. Hilos | 5 |
| 8. Semáforos | 5 |
| 9. Mejora al tener más de un núcleo | 5 |
| Conclusión | 6 |
| Referencias | 6 |

La cena de los filósofos

Cinco filósofos están sentados alrededor de una mesa y pasan su vida cenando y pensando, cada filósofo tiene un plato de fideos y un tenedor a la izquierda de su plato, para comer los fideos son necesarios dos tenedores y cada filósofo sólo puede tomar el tenedor que está a su izquierda y el de su derecha. Si cualquier filósofo toma un tenedor y el otro está ocupado, se quedará esperando, con el tenedor en la mano, hasta que pueda tomar el otro tenedor, para luego empezar a comer. El resto de filósofos que no está ni comiendo ni con un tenedor en la mano está pensando, el problema consiste en inventar un algoritmo que permita comer a los filósofos.

Se trata de lanzar cinco procesos (filósofos) y ponerlos a competir por obtener unos recursos, la solución consiste en configurar para que estos procesos (filósofos) puedan acceder a los recursos (dos tenedores) y desarrollar el trabajo (puedan comer), el problema es que el algoritmo de solución debe ser justo y no debe permitir que uno de ellos ocupe todo el sistema y no deje comer a los demás o que entre ellos se bloqueen y ninguno pueda tener acceso paralizando todo el trabajo.

Lógicamente si son cinco filósofos y hay cinco tenedores el algoritmo debe permitir que la mayor parte del tiempo haya dos filósofos comiendo y uno pensando. Pero además se deben tomar en cuenta diferentes factores:

- Si dos filósofos adyacentes intentan tomar el mismo tenedor a una vez, se produce una condición de carrera: ambos compiten por tomar el mismo tenedor, y uno de ellos se queda sin comer.
- Si todos los filósofos toman el tenedor que está a su derecha al mismo tiempo, entonces todos se quedarán esperando eternamente, porque alguien debe liberar el tenedor que les falta, nadie lo hará porque todos se encuentran en la misma situación (esperando que alguno deje sus tenedores). Entonces los filósofos se morirán de hambre.

El problema consiste en encontrar un algoritmo que permita que los filósofos nunca se mueran de hambre y que el comedor esté dando de comer constantemente y de una manera lo más eficiente posible.

Productor-Consumidor

El problema Productor/Consumidor consiste en el acceso concurrente por parte de procesos productores y procesos consumidores sobre un recurso común que resulta ser un buffer de elementos. Los productores tratan de introducir elementos en el buffer de uno en uno, y los consumidores tratan de extraer elementos de uno en uno.

Para asegurar la consistencia de la información almacenada en el buffer, el acceso de los productores y consumidores debe hacerse en exclusión mutua. Adicionalmente, el buffer es de capacidad limitada, de modo que el acceso por parte de un productor para introducir un elemento en el buffer lleno debe provocar la detención del proceso productor. Lo mismo sucede para un consumidor que intente extraer un elemento del buffer vacío.

La idea para la solución es la siguiente, ambos procesos (productor y consumidor) se ejecutan simultáneamente y se “despiertan” o “duermen” según el estado del buffer, el productor agrega productos mientras quede espacio y en el momento en que se llene el buffer se pone a “dormir” y cuando el consumidor toma un producto notifica al productor que puede comenzar a trabajar nuevamente. En caso contrario, si el buffer se vacía, el consumidor se pone a dormir y en el momento en que el productor agrega un producto crea una señal para despertarlo.

Lectores-Escritores

Hay un objeto de datos(fichero de texto) que es utilizado por varios procesos, unos leen y otro que escribe.

Solo puede utilizar el recurso un proceso y solo uno, es decir, o bien un proceso estará escribiendo o bien leyendo, pero nunca ocurrirá simultáneamente (teniendo en cuenta que si no lo está utilizando nadie, tendrá preferencia el escritor ante el lector).

Se considera a cada usuario(lector y escritor) como dos procesos y al fichero en cuestión como un recurso, para que un proceso acceda al recurso que necesita, tenemos que considerar a cada usuario (lector y escritor) como dos semáforos, estos semáforos son binarios y valen 0 si el recurso (fichero) está siendo utilizado por otro proceso y 1 si dicho recurso está disponible.

La solución de este problema se basa en implementar un algoritmo eficiente en el manejo de semáforos y memoria compartida, para que el problema esté bien resuelto se tiene que cumplir:

- No se puede hacer esperar a ningún proceso lector si ningún proceso escritor accede al recurso.
- Cuando un escritor debe realizar su tarea, debe ocurrir cuanto antes, sin que deban interferir los procesos lectores.(Para ello se establecen prioridades)

Preguntas

1. Problema de la concurrencia

Los problemas de concurrencia ocurren cuando los procesos compiten por los recursos del sistema y el SO debe elegir por medio de un algoritmo a quien se le asignan esos recursos, ya que si varios procesos modifican la misma información a la vez, cada uno podría destruir parte del trabajo de los demás, pero si lo hacen uno tras otro, en serie, se obtendrá el resultado correcto.

2. Procesos concurrentes cooperantes

Los procesos son cooperantes cuando están diseñados para trabajar conjuntamente en alguna actividad, por lo que deben ser capaces de comunicarse e interactuar entre ellos.

3. Exclusión mutua

Si un proceso está ejecutando código de la sección crítica, ningún otro proceso lo podrá hacer.

4. Interbloqueo.

Es una situación en la cual uno o más procesos están bloqueados como consecuencia de que cada proceso tiene retenido un subconjunto de los recursos necesarios para su finalización y tengan que esperar la liberación de los recursos restantes retenidos por otros procesos del mismo grupo.

5. Inanición.

Es cuando algún hilo es incapaz de disfrutar de los recursos que necesita porque otros hilos no los liberan.

6. Excesiva cortesía.

Es la tendencia a tratar todos los procesos de manera igualitaria y a ser demasiado justos en la asignación de recursos del sistema.

7. Hilos

Un hilo es una secuencia de código en ejecución dentro del contexto de un proceso, proporcionan la capacidad de tener más de un camino de ejecución en un mismo programa.

8. Semáforos

Son componentes pasivos de bajo nivel de abstracción que sirven para arbitrar el acceso a un recurso compartido.

9. Mejora al tener más de un núcleo

La mayor ventaja de tener más núcleos en una CPU es que pueden trabajar paralelamente para ofrecer un mejor rendimiento.

Conclusión

Si se mantiene un buen control de concurrencia se puede mantener la información consistente en los SO así como también evitar la aparición de errores, ya que los problemas siempre aparecen cuando la concurrencia no se controla, por lo que existen algunos mecanismos de bloqueo que nos permiten controlar la concurrencia en las transacciones, de esta manera, los sistemas de control de concurrencia deben garantizar la consistencia de la información.

Referencias

colaboradores de Wikipedia. (2021, 12 octubre). Problema de la cena de los filósofos. Wikipedia, la enciclopedia libre.
https://es.wikipedia.org/wiki/Problema_de_la_cena_de_los_fil%C3%B3sofos

colaboradores de Wikipedia. (2022, 13 diciembre). Problema productor-consumidor. Wikipedia, la enciclopedia libre.
https://es.wikipedia.org/wiki/Problema_productor-consumidor

Mateus, A. (s. f.). EL PROBLEMA DE LOS LECTORES Y ESCRITORES. prezi.com.
<https://prezi.com/llus9gxf4rsx/el-problema-de-los-lectores-y-escritores/>

Problemas de Concurrencia - Sistemas Operativos. (s. f.).
<https://sites.google.com/site/osupaep2010/administracion-de-procesos/problemas-de-concurrencia>

<https://pdfcoffee.com/pdfviewer/web/viewer.html?file=https%3A%2F%2Fpdfcoffee.com%2Fdownload%2Fsistemas-operativos-18-pdf-free.html%3Freader%3D1>