

Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías

Departamento de ciencias computacionales.

Ingeniería en computación



Seminario de Solución de Problemas de Sistemas Operativos - D01.

Violeta del Rocío Becerra Velázquez

“Filósofos, productor, lectores”

20/03/2023

Olguín Hernández Jair Benjamín

217439707

Índice

Contenido.....	3
Investigue en que consiste y las soluciones de los siguientes problemas:	3
La cena de los filósofos.	3
Productor-Consumidor.....	3
Lectores-Escritores.	3
1. ¿En qué consiste el problema de la concurrencia?.....	4
2. ¿Cuáles son los procesos concurrentes cooperantes?	4
3. ¿En qué consiste la exclusión mutua?	5
4. Defina Interbloqueo.	5
5. Defina Inanición.....	5
6. Defina Excesiva cortesía.....	6
7. ¿Qué son los hilos?	6
8. ¿Qué son los semáforos?	6
9. ¿Qué es lo que mejora el tener mas de un núcleo?	7
Conclusión	7
Bibliografía.....	8

Contenido.

Investigue en que consiste y las soluciones de los siguientes problemas:

La cena de los filósofos.

El problema de la cena de los filósofos es un clásico problema de concurrencia en ciencias de la computación, que se utiliza para ilustrar los problemas que pueden surgir en sistemas concurrentes cuando múltiples procesos intentan acceder a los mismos recursos simultáneamente.

El problema consiste en un grupo de 5 filósofos sentados alrededor de una mesa, donde cada uno de ellos tiene un plato de comida y un tenedor a cada lado. Los filósofos alternan entre dos estados: "pensando" y "comiendo". Para comer, un filósofo debe tener en su poder dos tenedores. Sin embargo, solo hay un tenedor disponible para cada filósofo, es decir, cinco tenedores en total.

Solución: Una de las soluciones podría ser el que se establezca un orden de turnos, donde por cada turno un filósofo pueda comer, y cada filósofo tenga que esperar. Ya que así se asegura que cada filósofo va a tener dos o ningún tenedor, donde solo uno de ellos puede tener dos.

Productor-Consumidor.

El problema del Productor-Consumidor es un problema de sincronización en sistemas operativos que surge cuando un conjunto de procesos productores genera datos y los depositan en un búfer, y otro conjunto de procesos consumidores retiran los datos del búfer para su uso. El objetivo es garantizar que los procesos productores y consumidores trabajen en armonía para evitar condiciones de carrera y otros problemas de concurrencia.

El problema consiste en que dos procesos, productor y consumidor, ambos comparten un búfer de tamaño finito. La tarea del productor es generar un producto, almacenarlo y comenzar nuevamente; mientras que el consumidor toma (simultáneamente) productos uno a uno. El problema consiste en que el productor no añada más productos que la capacidad del buffer y que el consumidor no intente tomar un producto si el buffer está vacío.

Solución: Una de las soluciones podría ser el hacer uso de variables de condición que se utilicen se produzca una determinada condición antes de continuar con la ejecución del proceso. Se pueden utilizar variables de condición para permitir que los procesos productores agreguen datos al búfer solo si hay espacio disponible y evitar que los procesos consumidores lean datos del búfer cuando esté vacío.

Lectores-Escritores.

El problema de los Lectores-Escritores es un problema de sincronización en sistemas operativos que surge cuando un conjunto de procesos lectores accede a un recurso compartido para leer datos, mientras que otro conjunto de procesos escritores

accede al mismo recurso para escribir datos. El objetivo es garantizar que los procesos lectores y escritores trabajen en armonía para evitar condiciones de carrera y otros problemas de concurrencia.

El problema consiste en que teniendo una base de datos que es utilizada para varios procesos, unos leen y otros escriben, pero 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).

Solución: Una de las soluciones podría utilizar algún sistema de prioridades para poder determinar cual de los procesos de lectura o escritura tiene acceso prioritario. Donde pueda que los escritores tengan una prioridad sobre los lectores y así evitar que se produzca alguna inanición para los procesos.

1. ¿En qué consiste el problema de la concurrencia?

El problema de la concurrencia se refiere a la situación en la que varios procesos o hilos de ejecución intentan acceder y manipular los mismos recursos compartidos de manera simultánea. Esto puede conducir a errores y comportamientos impredecibles, como la corrupción de datos, bloqueos y condiciones de carrera, en los que dos o más procesos intentan acceder al mismo recurso al mismo tiempo, lo que lleva a un estado inconsistente o incorrecto. El problema de la concurrencia se vuelve más complejo a medida que aumenta el número de procesos y recursos compartidos. La programación concurrente es la rama de la informática que se encarga de resolver estos problemas y desarrollar soluciones para garantizar la correcta coordinación y sincronización de los procesos concurrentes.

2. ¿Cuáles son los procesos concurrentes cooperantes?

Los procesos concurrentes cooperantes son aquellos que interactúan y colaboran entre sí para lograr una tarea común. Estos procesos suelen compartir recursos, comunicarse entre sí y coordinar sus acciones para lograr un objetivo común. Algunos ejemplos de procesos concurrentes cooperantes incluyen:

- Un sistema de gestión de bases de datos en el que varios procesos pueden leer y escribir datos en la misma base de datos.
- Un servidor web que maneja múltiples solicitudes simultáneamente, donde cada solicitud se maneja en un hilo de ejecución separado.
- Un sistema de edición de documentos colaborativo, en el que varios usuarios pueden editar el mismo documento al mismo tiempo y sus cambios se sincronizan en tiempo real.
- Un sistema de control de tráfico aéreo, en el que varios controladores aéreos trabajan juntos para coordinar el movimiento de aviones en el espacio aéreo compartido.

Los procesos concurrentes cooperantes trabajan juntos para lograr una tarea compartida de manera eficiente y efectiva, a través de la comunicación, la sincronización y la cooperación. La programación concurrente y la implementación de algoritmos y estructuras de datos adecuados son fundamentales para garantizar el correcto funcionamiento de estos sistemas.

3. ¿En qué consiste la exclusión mutua?

La exclusión mutua es un problema común en la programación concurrente que se refiere a la necesidad de garantizar que un recurso compartido solo pueda ser accedido y manipulado por un proceso a la vez. En otras palabras, se trata de asegurar que dos o más procesos no intenten acceder al mismo recurso compartido al mismo tiempo, lo que podría causar condiciones de carrera y errores en la aplicación.

La exclusión mutua se puede lograr mediante la implementación de mecanismos de sincronización, como los semáforos, los mutex o los monitores, que permiten a los procesos acceder a los recursos compartidos de manera ordenada y controlada. Estos mecanismos permiten que solo un proceso a la vez pueda bloquear y acceder al recurso compartido, mientras que los demás procesos esperan en una cola hasta que el recurso esté disponible. Por ejemplo, si varios procesos necesitan acceder a una base de datos, se puede implementar la exclusión mutua utilizando un mutex que se bloquea cuando un proceso está accediendo a la base de datos y se desbloquea cuando el proceso termina de trabajar con la base de datos. De esta manera, solo un proceso puede acceder a la base de datos a la vez, evitando posibles conflictos y errores.

4. Defina Interbloqueo.

El interbloqueo, también conocido como "deadlock" en inglés, se refiere a una situación en la que dos o más procesos o hilos de ejecución están bloqueados permanentemente, ya que cada uno está esperando que otro proceso libere un recurso que necesita para continuar su ejecución. En otras palabras, se produce una situación de interbloqueo cuando dos o más procesos se bloquean entre sí porque cada uno necesita un recurso que está siendo utilizado por otro proceso, y ninguno de ellos puede continuar sin ese recurso. El interbloqueo puede ser causado por una variedad de factores, como la asignación inadecuada de recursos, la falta de coordinación entre los procesos, la competencia por los recursos compartidos y la programación inadecuada de los algoritmos de sincronización. El interbloqueo es un problema grave en la programación concurrente, ya que puede causar la paralización de la aplicación, lo que puede llevar a la pérdida de datos y a la interrupción de los servicios.

5. Defina Inanición.

La inanición, también conocida como "starvation" en inglés, es un problema que ocurre en la programación concurrente cuando un proceso o hilo de ejecución se ve privado de los recursos necesarios para continuar su ejecución de manera indefinida. En otras palabras, la inanición se produce cuando un proceso no puede avanzar y quedará bloqueado o en espera indefinidamente debido a la falta de recursos que necesita. La

inanición puede ser causada por una variedad de factores, como la asignación inadecuada de recursos, la falta de priorización de los procesos, la programación inadecuada de los algoritmos de sincronización y la competencia por los recursos compartidos. La inanición puede ser un problema grave en la programación concurrente, ya que puede causar la paralización de la aplicación, lo que puede llevar a la pérdida de datos y a la interrupción de los servicios.

6. Defina Excesiva cortesía.

La excesiva cortesía, también conocida como "over-politeness" en inglés, se refiere a una actitud exagerada y demasiado formal en las interacciones sociales. En otras palabras, la excesiva cortesía se produce cuando una persona muestra una cortesía o amabilidad extrema que puede parecer exagerada o incluso falsa. La excesiva cortesía puede ser causada por una variedad de factores, como la ansiedad social, la necesidad de agradar a los demás, la falta de confianza en sí mismo y el deseo de evitar conflictos. Aunque la cortesía es una forma importante de comunicación social y puede ser muy útil en muchas situaciones, la excesiva cortesía puede tener efectos negativos, como la creación de barreras en las relaciones sociales, la falta de autenticidad en las interacciones y la pérdida de confianza en uno mismo.

7. ¿Qué son los hilos?

Un hilo, también conocido como "thread" en inglés, se refiere a la unidad más pequeña de procesamiento dentro de un proceso. Un proceso es un programa en ejecución en el sistema operativo, mientras que un hilo es una secuencia de instrucciones que pueden ser ejecutadas concurrentemente con otras secuencias de instrucciones dentro del mismo proceso. Los hilos comparten el mismo espacio de memoria y los mismos recursos del sistema que el proceso que los creó, lo que los hace más eficientes en términos de uso de recursos que la creación de múltiples procesos separados. Además, los hilos pueden ser programados para trabajar de manera independiente y al mismo tiempo, lo que puede mejorar la eficiencia de la aplicación.

La capacidad de los sistemas operativos para admitir múltiples hilos en un solo proceso se llama "multihilo" o "multithreading". Los sistemas operativos modernos suelen tener soporte integrado para hilos y ofrecen bibliotecas de programación que permiten a los desarrolladores crear aplicaciones multihilo de manera sencilla.

8. ¿Qué son los semáforos?

Un semáforo es una variable de sincronización que se utiliza para controlar el acceso a recursos compartidos por múltiples procesos o hilos. El semáforo se utiliza para evitar condiciones de carrera y garantizar la exclusión mutua en el acceso a recursos compartidos.

El semáforo se puede considerar como una especie de señalización que indica si un recurso está disponible o no. Los semáforos tienen dos operaciones básicas: "esperar" (wait) y "señalar" (signal). La operación de espera reduce el valor del semáforo y bloquea el proceso o hilo si el valor del semáforo es cero. La operación de señalar aumenta el

valor del semáforo y desbloquea cualquier proceso o hilo que esté esperando. Por lo tanto, los semáforos se utilizan comúnmente en los sistemas operativos como mecanismo de sincronización para controlar el acceso a los recursos compartidos, como la memoria, los archivos, los dispositivos de entrada/salida, entre otros. Los semáforos son especialmente útiles en situaciones en las que múltiples procesos o hilos necesitan acceder a un recurso compartido de manera concurrente, ya que pueden garantizar que solo un proceso o hilo acceda al recurso compartido en un momento dado.

9. ¿Qué es lo que mejora el tener mas de un núcleo?

El tener más de un núcleo en un procesador permite que una computadora pueda realizar múltiples tareas al mismo tiempo. Cada núcleo es como un "cerebro" separado en el procesador, que puede ejecutar instrucciones independientemente del otro núcleo.

Cuando una computadora tiene múltiples núcleos, cada núcleo puede manejar una tarea diferente simultáneamente, lo que permite una mejor multitarea y un rendimiento más rápido en general. También puede mejorar la capacidad de respuesta del sistema en general. Por ejemplo, mientras hacemos una tarea de fondo en computadora, como descargar un archivo grande, la computadora puede seguir siendo rápida y ágil para otras tareas, como navegar por Internet o trabajar en un procesador de texto.

Conclusión.

En esta actividad pude tener mas conocimiento de algunos conceptos que pueden llegar a ser más técnicos en el tema de sistemas operativos, pero así mismo te ayudan a entender un poco mas del cómo es que funcionan los sistemas operativos y todo la organizan y situaciones que tienen que llegar a contemplar. Y con esto también hago mención de los problemas presentados, ya que yo trate de primero resolver el problema con los conocimientos que tengo, y después leí algunas respuestas y unas son bastante eficientes, pero necesitan de conocimientos, y vi que no estaba tan alejado de las soluciones presentadas. Aun así creo que tengo que profundizar un poco en los temas vistos ya que muchos necesitas verlos a fondo con ejemplos para poder entender el por que suceden y como pueden solucionarse, como con los problemas creo que es una buena manera de entender el planteamiento de los problemas en los sistemas operativos.

Bibliografía

Cabala PC. (n.d.). Sistemas Operativos. [www.dc.fi.udc.es](http://www.dc.fi.udc.es/~so-grado/SO-Procesos-planif.pdf). Retrieved Marzo 20, 2023, from <https://www.dc.fi.udc.es/~so-grado/SO-Procesos-planif.pdf>

Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts. Wiley.

ESTRADA SOTO, J. (s. f.). *PROBLEMAS CLÁSICOS DE LA COMUNICACION ENTRE LOS PROCESOS*. [academicos.azc.uam.mx](https://academicos.azc.uam.mx/jestrada/archivos/so/tema4/p_clasicos.pdf). https://academicos.azc.uam.mx/jestrada/archivos/so/tema4/p_clasicos.pdf

PROBLEMA DE LOS LECTORES-ESCRITORES. (s. f.). <https://www.infor.uva.es/%7Ecllamas/concurr/pract98/sisos30/index.html>

Problema del productor y consumidor. (s. f.). <http://formella.webs.uvigo.es/doc/cd04/node72.html>

Sistemas Operativos - Interbloqueo. (s. f.). <https://sistemasoperativos07.es.tl/Interbloqueo.htm>

Steynberg, M. (2019, 16 diciembre). *Definición y descripción general interbloqueo de SQL Server*. SQL Shack - articles about database auditing, server performance, data recovery, and more. <https://www.sqlshack.com/es/definicion-y-descripcion-general-interbloqueo-de-sql-server/>