



Centro Universitario De Ciencias Exactas e Ingenierías

Sistemas Operativos

Becerra Velázquez Violeta del Rocío

*De Santiago Rodríguez Armando*

Código: 222362658

*Ruiz Arana Adrian*

*Código: 222362119*

*Solorzano Razo Javier*

*Código: 218432544*

Ingeniería en Computación (ICOM)

Sección: D04

Investigación. Soluciones para problemas de exclusión, por software, por hw, del sistema

09/03/2024

Investigación. Soluciones para problemas de exclusión, por software, por hw, del sistema

## Índice

### Tabla de contenido

Índice .....	2
Exclusión Mutua .....	2
Problemas de Exclusión Mutua.....	2
Algoritmos de Exclusión Mutua .....	2
Bloqueo de Recursos Compartidos.....	3
Tipos de Bloqueos.....	3
Implementación en Sistemas Operativos .....	3
Conflictos de Acceso en Sistemas Multiusuario .....	3
Políticas de Control de Acceso.....	3
Soluciones Avanzadas.....	3
Estudios de Caso .....	4
Conclusion .....	4
Referencias .....	4

## Introducción

La exclusión en sistemas operativos es esencial tanto a nivel de hardware como de software para garantizar el acceso seguro y eficiente a los recursos compartidos. Tanto en el nivel de software como de hardware, la exclusión es fundamental para evitar problemas como la corrupción de datos, las condiciones de carrera y el bloqueo del sistema.

### Exclusión Mutua

La exclusión mutua es fundamental en sistemas operativos para evitar que varios procesos accedan simultáneamente a un recurso compartido, lo que podría provocar corrupción de datos o condiciones de carrera. A nivel de software, se implementa utilizando algoritmos como el de Peterson o el de Dekker. El algoritmo de Peterson, por ejemplo, es simple y eficiente para dos procesos, pero puede volverse complicado con más procesos debido a la necesidad de mantener el orden de turno. El algoritmo de Dekker, por otro lado, garantiza la exclusión mutua sin necesidad de un turno explícito, pero puede sufrir del problema de la inanición.

### Problemas de Exclusión Mutua

Los problemas incluyen condiciones de carrera, deadlock y corrupción de datos.

### Algoritmos de Exclusión Mutua

*Software:* Algoritmos como Peterson, Dekker, entre otros.

*Hardware:* Mecanismos como las instrucciones de bloqueo de bus o las instrucciones atómicas.

### Bloqueo de Recursos Compartidos

En sistemas operativos, el bloqueo de recursos compartidos se utiliza para garantizar que solo un proceso tenga acceso exclusivo a un recurso en un momento dado. A nivel de software, se emplean estructuras como semáforos y mutex. Los semáforos son variables especiales que se utilizan para controlar el acceso a recursos compartidos mediante señales de "espera" y "liberación". Los mutex (mutual exclusion locks) son otro mecanismo de exclusión mutua que permite que solo un hilo pueda acceder al recurso compartido a la vez. A nivel de hardware, se pueden usar instrucciones específicas de bloqueo de bus o instrucciones atómicas para implementar el bloqueo de recursos compartidos de manera eficiente.

### Tipos de Bloqueos

*Software:* Semáforos, mutex, entre otros.

*Hardware:* Instrucciones de bloqueo de bus, instrucciones atómicas, entre otros.

### Implementación en Sistemas Operativos

Los sistemas operativos proporcionan mecanismos tanto de software como de hardware para crear, adquirir y liberar bloqueos. A nivel de software, los sistemas operativos ofrecen API y bibliotecas para utilizar semáforos, mutex y otras estructuras de exclusión mutua. A nivel de hardware, los sistemas operativos interactúan con el hardware para utilizar instrucciones específicas de bloqueo de bus o instrucciones atómicas cuando sea necesario.

### Conflictos de Acceso en Sistemas Multiusuario

En entornos multiusuario, los sistemas operativos deben abordar conflictos de acceso a recursos compartidos entre múltiples usuarios y procesos. Esto implica implementar políticas de control de acceso tanto a nivel de software como de hardware. A nivel de software, se utilizan listas de control de acceso (ACL), permisos de archivo y directorio, y control de acceso basado en roles (RBAC). A nivel de hardware, se pueden emplear mecanismos de virtualización y asignación de recursos para controlar el acceso a recursos compartidos en sistemas

### Políticas de Control de Acceso

*Software:* Listas de control de acceso (ACL), permisos de archivo y directorio, control de acceso basado en roles (RBAC).

*Hardware:* Mecanismos de virtualización, asignación de recursos, entre otros.

### Soluciones Avanzadas

Además de las soluciones tradicionales de exclusión mutua, existen soluciones más avanzadas para problemas de exclusión en sistemas operativos. Por ejemplo, en sistemas distribuidos, se utilizan técnicas de sincronización para coordinar la ejecución

Investigación. Soluciones para problemas de exclusión, por software, por hw, del sistema de procesos en diferentes nodos de la red. En bases de datos, se implementan mecanismos de control de concurrencia para garantizar la consistencia de los datos en entornos con múltiples transacciones concurrentes.

### Estudios de Caso

Los estudios de caso son fundamentales para comprender cómo se abordan los problemas de exclusión en sistemas operativos en situaciones reales. Por ejemplo, se pueden estudiar casos en los que se han enfrentado problemas de exclusión mutua en sistemas de tiempo real, sistemas distribuidos o bases de datos de alto rendimiento. Estos casos proporcionan ejemplos concretos de cómo se han aplicado diferentes técnicas y algoritmos para resolver problemas de exclusión en sistemas operativos.

### Conclusion

La exclusión en sistemas operativos, tanto a nivel de hardware como de software, es esencial para garantizar el acceso seguro y eficiente a los recursos compartidos. La exclusión mutua, fundamental para evitar problemas como la corrupción de datos y las condiciones de carrera, se implementa mediante algoritmos como Peterson o Dekker a nivel de software, y mecanismos como instrucciones de bloqueo de bus o instrucciones atómicas a nivel de hardware. El bloqueo de recursos compartidos se logra mediante estructuras como semáforos y mutex a nivel de software, y mediante instrucciones específicas a nivel de hardware.

### Referencias

- Tanenbaum, A. S., & Bos, H. (2015). Modern Operating Systems (4th ed.). Pearson.
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts (10th ed.). Wiley.
- Stallings, W. (2018). Operating Systems: Internals and Design Principles (9th ed.). Pearson.
- Tanenbaum, A. S., & Van Steen, M. (2007). Distributed Systems: Principles and Paradigms (2nd ed.). Pearson.
- Garg, R. K. (2013). Concurrent and Distributed Computing in Java. John Wiley & Sons.