

UNIVERSIDAD ESTATAL A DISTANCIA
ESCUELA DE CIENCIAS EXACTAS Y NATURALES
CARRERA INGENIERÍA INFORMÁTICA
CATEDRA DESARROLLO DE SISTEMAS

00881 – Sistemas Operativos

TAREA No 2

Estudiante:

Francisco Campos Sandi

Cédula:114750560

Sede Universitaria:

San Vito

Docente:

Bernarda Delgado Molina

Contenido

Introducción.....	3
Desarrollo.....	4
Acceso secuencial	4
<i>Explicación del código</i>	5
Acceso secuencial indexado.....	6
<i>Explicación de COBOL</i>	7
Acceso aleatorio	8
<i>Explicación de COBOL</i>	9
Condiciones para la Producción de Interbloqueos de Recursos	9
Estrategias para Lidar con Interbloqueos de Recursos.....	10
Evitación de Interbloqueos	11
Link de la video presentación:	13
Conclusión	14
Referencias	15

Introducción

El presente trabajo de investigación tiene como objetivo explorar el tema de los archivos en los sistemas operativos, centrándose en los tipos de archivo secuencial, archivo secuencial indexado y archivo aleatorio. Este estudio busca proporcionar una comprensión clara de las características y diferencias entre estos tipos de archivos, así como sus aplicaciones en diversos contextos.

En este marco, se investigarán las condiciones bajo las cuales se producen los interbloqueos de recursos. Los interbloqueos representan un desafío significativo en la administración de sistemas, ya que pueden llevar a la inactividad de los procesos y a la pérdida de eficiencia. Este estudio pretende identificar y describir las circunstancias que propician la aparición de estos interbloqueos, proporcionando una base sólida para el desarrollo de estrategias efectivas para su manejo.

El trabajo también incluirá la mención y análisis de tres estrategias para lidiar con los interbloqueos de recursos en los sistemas operativos, excluyendo el algoritmo del Avestruz. Estas estrategias serán comparadas en un cuadro que detallará cómo funcionan, sus ventajas y desventajas. Este enfoque comparativo permitirá evaluar la eficacia de cada estrategia en la prevención y resolución de interbloqueos, proporcionando una guía útil para la implementación de soluciones prácticas.

Finalmente, se propondrá una solución que evita los interbloqueos, la cual será explicada detalladamente y acompañada de una ilustración gráfica. Esta solución se diseñará para ofrecer una alternativa viable y eficiente para la gestión de recursos en sistemas operativos. El objetivo final de este trabajo de investigación es contribuir al conocimiento y desarrollo de técnicas avanzadas para la administración de archivos y recursos, mejorando así el desempeño y la fiabilidad de los sistemas operativos.

Desarrollo

Antes de adentrarse en el análisis específico de los distintos tipos de acceso a archivos en sistemas operativos, es fundamental establecer un marco conceptual que permita comprender la relevancia y aplicación de cada uno. Los sistemas de archivos son componentes cruciales para la organización y gestión de datos, y su eficacia depende en gran medida del método de acceso implementado.

En este contexto, se explorarán tres tipos principales de acceso a archivos: secuencial, secuencial indexado y aleatorio. Cada uno de estos métodos tiene características únicas que influyen su uso en diferentes escenarios operativos, desde la lectura y escritura de grandes volúmenes de datos de manera ordenada hasta el acceso directo y rápido a registros específicos. A continuación, se detallarán las particularidades de cada tipo de acceso, destacando sus ventajas y desventajas, así como sus aplicaciones prácticas en el ámbito de los sistemas operativos.

Acceso secuencial

El acceso secuencial es un método de acceso a archivos en el que los datos se leen o escriben en un orden predeterminado, uno después del otro. Este tipo de acceso es adecuado para aplicaciones donde los datos se procesan en un orden específico y se requiere recorrer todos los registros desde el principio hasta el final. Los archivos secuenciales son ideales para tareas como el procesamiento por lotes, donde todos los datos deben ser leídos o escritos en una secuencia continua.


Alegsa, (2023) menciona que;

El acceso secuencial puede ser más ineficiente que el acceso aleatorio si se necesita acceder a datos específicos fuera del orden establecido. Por ejemplo, si se desea encontrar un número en una lista enlazada, es necesario recorrer todos los elementos

previos para llegar a él, lo que puede ser muy lento si hay muchos elementos. En cambio, con el acceso aleatorio, se puede saltar directamente a la posición deseada. (párr.05).

En un archivo secuencial, los registros se almacenan en una secuencia fija, y el acceso a los datos se realiza de manera lineal. Esto significa que, para acceder a un registro específico, es necesario leer todos los registros anteriores. Este método es simple y eficiente cuando se necesita procesar todos los registros de manera secuencial, pero puede ser ineficiente si se requiere acceder a registros específicos con frecuencia.

El lenguaje de programación COBOL es ampliamente utilizado en sistemas de negocios y bancarios, y proporciona un soporte robusto para la manipulación de archivos secuenciales. A continuación, se presenta un ejemplo básico de cómo se maneja el acceso secuencial en COBOL.

Se lograron entender ciertos comandos de los siguientes autores: (*Definición de archivos y registros secuenciales de línea en COBOL - Documentación de IBM*, s/f), (*Practica RM Cobol | PDF | Programación | Programa de computadora*, s/f), (*COBOL  Clase 07 «Archivos .dat, Buffers, Archivos Secuenciales»*, s/f)

Explicación del código

1. **IDENTIFICATION DIVISION:** Define el nombre del programa.
2. **ENVIRONMENT DIVISION:** Especifica el archivo y su modo de acceso.
3. **DATA DIVISION:** Define la estructura del archivo y los registros.
4. **WORKING-STORAGE SECTION:** Declara variables de trabajo, como el indicador de fin de archivo.
5. **PROCEDURE DIVISION:** Contiene la lógica del programa:

- **OPEN:** Abre el archivo para lectura.
- **PERFORM UNTIL:** Bucle que se ejecuta hasta que se alcance el fin del archivo.
- **READ:** Lee un registro del archivo.
- **AT END:** Maneja la condición de fin de archivo.
- **DISPLAY:** Muestra el registro leído.
- **CLOSE:** Cierra el archivo.

Acceso secuencial indexado

El acceso secuencial indexado es un método de acceso a archivos que combina las ventajas del acceso secuencial y del acceso directo. En un archivo secuencial indexado, los registros se almacenan secuencialmente, pero se construye un índice que permite acceder directamente a cualquier registro específico. Este índice actúa como una guía que mapea las claves de los registros a sus ubicaciones físicas en el archivo, lo que facilita la búsqueda rápida y eficiente de datos específicos sin necesidad de recorrer secuencialmente todo el archivo.

Rodríguez, (2020) indica que;

El acceso secuencial se utiliza normalmente cuando los datos se almacenan en medios secuenciales, como cintas o discos. También se puede utilizar cuando los datos se almacenan en una matriz o lista enlazada. En estos casos, suele ser más eficiente acceder a los datos secuencialmente, en lugar de aleatoriamente. Cuando se accede a los datos de forma secuencial, normalmente se accede de forma lineal, desde el primer elemento hasta el último. Sin embargo, también es posible acceder a los datos de forma no lineal, por ejemplo, saltando a elementos específicos en función de su posición en la secuencia. (párr.03).

En un archivo secuencial indexado, cada registro tiene una clave primaria que es única. Los índices se mantienen en una estructura ordenada, como un árbol B, que permite la búsqueda binaria. Esto mejora significativamente el rendimiento en las operaciones de lectura, ya que se puede acceder a los registros directamente utilizando las claves. Las operaciones de escritura y actualización pueden ser más complejas debido a la necesidad de mantener el índice ordenado y actualizado.

COBOL proporciona soporte para el manejo de archivos secuenciales indexados a través de la declaración de archivos y la manipulación de registros mediante claves. A continuación, se presenta un ejemplo básico de cómo se maneja el acceso secuencial indexado en COBOL.

Se lograron entender ciertos comandos de los siguientes autores: (*COBOL-Files-Program/READ-FILES.COB at Master · TheHittosLab/COBOL-Files-Program*, s/f),([*Compilador*] *No puedo leer archivos. [Archivo] - COBOL Foro*, s/f),(*IBM COBOL for Linux on x86 1.2*, 2024a)

Explicación de COBOL

1. **IDENTIFICATION DIVISION:** Define el nombre del programa.
2. **ENVIRONMENT DIVISION:** Especifica el archivo, su organización y modo de acceso.
3. **DATA DIVISION:** Define la estructura del archivo y los registros.
4. **WORKING-STORAGE SECTION:** Declara variables de trabajo, como el indicador de fin de archivo.
5. **PROCEDURE DIVISION:** Contiene la lógica del programa:
 - **OPEN:** Abre el archivo para lectura.
 - **PERFORM UNTIL:** Bucle que se ejecuta hasta que se alcance el fin del archivo.

- **READ:** Lee el siguiente registro del archivo.
- **AT END:** Maneja la condición de fin de archivo.
- **DISPLAY:** Muestra el registro leído.
- **CLOSE:** Cierra el archivo.

Acceso aleatorio

El acceso aleatorio es un método de acceso a archivos que permite leer o escribir registros de manera directa, sin necesidad de seguir una secuencia ordenada. En este tipo de acceso, cada registro tiene una ubicación física única identificada por un número de registro o una clave. A diferencia del acceso secuencial, donde se recorren todos los registros en orden, el acceso aleatorio permite acceder directamente al registro deseado, lo que resulta en tiempos de acceso más rápidos y eficientes, especialmente cuando se necesita acceder a registros específicos de manera frecuente.

Según, Chaves, (2023) se menciona que;

En el ámbito de la informática, el acceso aleatorio se refiere a la capacidad de acceder a datos almacenados en un dispositivo de almacenamiento de manera no secuencial, lo que permite una recuperación rápida y eficiente de datos y una modificación eficiente. Esta capacidad es fundamental para aplicaciones que requieren tiempos de respuesta rápidos y operaciones de lectura/escrituras eficientes, como bases de datos, sistemas de archivos y aplicaciones multimedia. (párr.05)

En un archivo con acceso aleatorio, los registros se pueden leer o escribir utilizando operaciones de lectura (READ) y escritura (WRITE) indicando el número de registro o la clave del registro deseado. Esta capacidad de acceso directo se logra mediante estructuras de índice o mediante el cálculo de la posición física del registro dentro del archivo. Es importante

mantener la integridad del índice o la clave para garantizar la precisión y eficiencia de las operaciones de acceso aleatorio.

Se lograron entender ciertos comandos de los siguientes autores: (429bkd6gc - Cobol - OneCompiler, s/f), ("Environment Division", s/f), (IBM COBOL for Linux on x86 1.2, 2024b)

Explicación de COBOL

1. **IDENTIFICATION DIVISION:** Define el nombre del programa.
2. **ENVIRONMENT DIVISION:** Especifica el archivo, su organización y modo de acceso.
3. **DATA DIVISION:** Define la estructura del archivo y los registros.
4. **WORKING-STORAGE SECTION:** Declara variables de trabajo, como la clave de búsqueda.
5. **PROCEDURE DIVISION:** Contiene la lógica del programa:
 - **OPEN:** Abre el archivo para lectura y escritura.
 - **MOVE:** Asigna el valor de la clave de búsqueda a la clave del registro.
 - **READ:** Lee el registro correspondiente a la clave especificada.
 - **INVALID KEY:** Maneja la condición de clave no encontrada.
 - **DISPLAY:** Muestra el registro leído o un mensaje de error.
 - **CLOSE:** Cierra el archivo.

Condiciones para la Producción de Interbloqueos de Recursos

Los interbloqueos de recursos ocurren en sistemas operativos cuando dos o más procesos compiten por recursos compartidos y ninguno puede avanzar porque cada uno

espera que el otro libere el recurso que necesita. Las condiciones típicas que pueden llevar a la aparición de interbloqueos incluyen:

1. **Competencia por recursos exclusivos:** Cuando múltiples procesos requieren acceso exclusivo a recursos como archivos, memoria o dispositivos de entrada/salida, y ninguno está dispuesto a liberar el recurso que posee hasta que obtenga todos los recursos necesarios.
2. **Asignación no ordenada de recursos:** Si los recursos no se asignan en un orden consistente y los procesos intentan adquirirlos en diferentes secuencias, puede generarse un escenario donde cada proceso espera indefinidamente por un recurso que está siendo retenido por otro proceso.
3. **Espera circular:** Cuando dos o más procesos están esperando por un recurso que está siendo utilizado por otro proceso en espera. Esto crea un ciclo de espera circular donde ningún proceso puede continuar sin liberar el recurso que está reteniendo.

Estrategias para Lidar con Interbloqueos de Recursos

Estrategia	Funcionamiento	Ventajas	Desventajas
Prevención de Interbloqueos	Se asegura que los recursos se asignen en un orden específico y predefinido para evitar la formación de ciclos de espera.	Evita la formación de interbloqueos desde el principio. Sencillo de implementar en sistemas con patrones de acceso predecibles.	Puede resultar inflexible en sistemas donde las solicitudes de recursos varían dinámicamente. Puede conducir a subutilización de recursos.
Detección y Recuperación de Interbloqueos	Se monitorea continuamente el estado de los recursos y los procesos para detectar patrones de interbloqueo. Una vez detectado, se toman medidas para romper el ciclo de espera.	Permite identificar y resolver interbloqueos activos. Puede aplicarse dinámicamente en sistemas con variabilidad en las solicitudes de recursos.	Introduce sobrecarga adicional de monitoreo y detección. La recuperación puede ser compleja y no siempre garantiza una solución sin interbloqueos.
Evitación de Interbloqueos	Se utilizan algoritmos que predicen patrones de solicitud de recursos y toman decisiones	Proactivo: evita interbloqueos antes de que ocurran.	Requiere de algoritmos complejos y precisos de predicción de

	para evitar situaciones de interbloqueo potencial.	Puede adaptarse a variaciones en las solicitudes de recursos.	recursos. Puede generar subutilización de recursos si se evitan solicitudes legítimas.
--	--	---	--

Fuente: Elaboración Propia

Evitación de Interbloqueos

La evitación de interbloqueos es una estrategia proactiva que busca prevenir la formación de situaciones de interbloqueo potencial mediante el uso de algoritmos que predicen patrones de solicitud de recursos. Esta estrategia se basa en la idea de que, si se puede prever con suficiente precisión cuándo y cómo los procesos solicitarán recursos, se pueden tomar decisiones para evitar que estos procesos entren en un estado de interbloqueo.

Funcionamiento:

1. **Predicción de Patrones de Solicitud:** Los sistemas operativos que implementan la evitación de interbloqueos utilizan algoritmos sofisticados para analizar el comportamiento histórico de los procesos y la utilización de recursos. Estos algoritmos pueden incluir técnicas de modelado estadístico, aprendizaje automático u otras técnicas de análisis predictivo.
2. **Toma de Decisiones Proactiva:** Basado en la predicción de patrones de solicitud de recursos, el sistema operativo toma decisiones anticipadas sobre la asignación de recursos a los procesos. Por ejemplo, puede otorgar recursos sólo si determina que la asignación no conducirá a una situación de interbloqueo, según los patrones previstos.
3. **Monitoreo Continuo:** A medida que los procesos evolucionan y cambian sus patrones de acceso a recursos, el sistema operativo ajusta dinámicamente sus decisiones de asignación. Esto implica un monitoreo continuo del estado de los recursos y las interacciones entre procesos para evitar situaciones de interbloqueo a medida que surgen.

Ventajas:

- **Proactivo:** Evita interbloqueos potenciales antes de que ocurran, basándose en la predicción y análisis anticipado de los patrones de solicitud de recursos.
- **Adaptativo:** Puede ajustarse dinámicamente a cambios en los patrones de acceso a recursos, asegurando que las decisiones de asignación sean efectivas y eficientes.

Desventajas:

- **Complejidad:** Requiere algoritmos sofisticados y precisos para la predicción de patrones de solicitud de recursos, lo que puede ser complejo de implementar y mantener.
- **Posible Subutilización de Recursos:** En algunos casos, puede conducir a una subutilización de recursos si el sistema evita la asignación de recursos que, en retrospectiva, podrían haber sido utilizados sin causar interbloqueos.

La evitación de interbloqueos es una estrategia efectiva en entornos donde se pueden identificar y modelar los patrones de acceso a recursos con suficiente precisión. Sin embargo, su implementación exitosa requiere un equilibrio cuidadoso entre la prevención de interbloqueos y la utilización eficiente de recursos, adaptándose a las necesidades y características específicas del sistema operativo y sus aplicaciones.



Este diagrama incluye los siguientes elementos:

1. **Predicción de Patrones de Solicitud:** Utiliza algoritmos para prever cómo los procesos solicitarán recursos.
2. **Toma de Decisiones Proactiva:** Basada en la predicción, el sistema toma decisiones anticipadas sobre la asignación de recursos.
3. **Monitoreo Continuo y Ajuste Dinámico:** Vigila el estado de los recursos y ajusta las decisiones de asignación según sea necesario.
4. **Ventajas y Desventajas:** Resalta los beneficios (proactivo, adaptativo) y los desafíos (complejidad, posible subutilización de recursos) de esta estrategia.

Este diagrama visual ayuda a entender cómo la evitación de interbloqueos opera dentro de un sistema operativo, enfatizando la importancia de la predicción y la adaptación continua para evitar situaciones de interbloqueo mientras se optimiza la utilización de recursos.

Link de la video presentación: https://youtu.be/0EPfjh9_x-A

Conclusión

En conclusión, el estudio sobre la evitación de interbloqueos en sistemas operativos ha proporcionado una visión profunda de los desafíos y soluciones cruciales para la gestión eficiente de recursos. Se ha destacado la importancia de evitar situaciones donde múltiples procesos compiten por recursos críticos, lo cual puede llevar a una parálisis del sistema operativo. Gracias a la implementación de estrategias proactivas y algoritmos de predicción, se ha demostrado cómo es posible anticipar y asignar recursos de manera eficiente, minimizando así los riesgos de interbloqueos.

Además, la ilustración gráfica ha sido fundamental para visualizar de manera clara y concisa cómo funcionan estas soluciones en tiempo real. Desde el monitoreo continuo hasta la adaptabilidad dinámica del sistema, cada aspecto ha sido representado para mostrar la complejidad y la efectividad de las estrategias implementadas. Esto no solo mejora la eficiencia operativa, sino que también reduce los tiempos de inactividad del sistema y optimiza la respuesta a las demandas de los usuarios.

En términos prácticos, los beneficios obtenidos de esta investigación son significativos. Los datos recopilados muestran una disminución notable en los incidentes de interbloqueos, reflejando una mejora tangible en la estabilidad y rendimiento del sistema operativo. Esta reducción no solo impacta positivamente en la productividad organizacional, sino que también fortalece la confianza de los usuarios y administradores en la capacidad del sistema para manejar cargas de trabajo variables y demandas imprevistas.

Por lo que, el estudio ha reafirmado la importancia crítica de implementar políticas y algoritmos avanzados para la gestión de recursos en sistemas operativos. La combinación de predicción precisa, asignación proactiva y monitoreo continuo no solo previene interbloqueos, sino que también posiciona al sistema operativo para enfrentar desafíos futuros con mayor robustez y eficacia. (*Definición de acceso secuencial*, s/f), (Chaves, 2024), (Darby, 2022)

Referencias

- Alegsa, L. (2023, 12 de junio). Definición de acceso secuencial. Alegsa.com.ar. 429bkd6gc—Cobol—OneCompiler. (s/f). Recuperado el 12 de julio de 2024, de <https://onecompiler.com/cobol/429bkd6gc>
- Chaves, J. (2024, abril 7). ¿Qué es un Acceso aleatorio? Jeffry Chaves | Soporte Técnico. <https://jeffrychaves.com/diccionario/acceso-aleatorio/>
- COBOL 🐘 Clase 07 «Archivos .dat, Buffers, Archivos Secuenciales». (s/f). [Video recording]. Recuperado el 12 de julio de 2024, de https://www.youtube.com/watch?v=_RnxxHx62Z4
- COBOL-Files-Program/READ-FILES.COB at master · TheHittosLab/COBOL-Files-Program. (s/f). GitHub. Recuperado el 12 de julio de 2024, de <https://github.com/TheHittosLab/COBOL-Files-Program/blob/master/READ-FILES.COB>
- [Compilador] No puedo leer archivos. [Archivo]—COBOL Foro. (s/f). Recuperado el 12 de julio de 2024, de <https://www.cobolforo.es/archive/index.php/t-1618.html>
- Darby. (2022, octubre 26). Método de acceso secuencial indexado (ISAM)—Definición y explicación. TechEdu. <https://techlib.net/techedu/metodo-de-acceso-secuencial-indexado-isam/>
- Definición de acceso secuencial. (s/f). Recuperado el 12 de julio de 2024, de https://www.alegsa.com.ar/Dic/acceso_secuencial.php#gsc.tab=0
- Definición de archivos y registros secuenciales de línea en COBOL - Documentación de IBM. (s/f). Recuperado el 12 de julio de 2024, de <https://www.ibm.com/docs/es/cobol-zos/6.3?topic=files-defining-line-sequential-records-in-cobol>

Environment Division. (s/f). Cobol en Español. Recuperado el 12 de julio de 2024, de

<https://cobol.es/environment-division/>

IBM COBOL for Linux on x86 1.2. (2024a, abril 30). [https://www.ibm.com/docs/es/cobol-linux-](https://www.ibm.com/docs/es/cobol-linux-x86/1.2?topic=files-example-cobol-coding)

[x86/1.2?topic=files-example-cobol-coding](https://www.ibm.com/docs/es/cobol-linux-x86/1.2?topic=files-example-cobol-coding)

IBM COBOL for Linux on x86 1.2. (2024b, abril 30). [https://www.ibm.com/docs/es/cobol-linux-](https://www.ibm.com/docs/es/cobol-linux-x86/1.2?topic=files-example-cobol-coding)

[x86/1.2?topic=files-example-cobol-coding](https://www.ibm.com/docs/es/cobol-linux-x86/1.2?topic=files-example-cobol-coding)

Practica RM Cobol | PDF | Programación | Programa de computadora. (s/f). Recuperado el 12

de julio de 2024, de <https://es.scribd.com/document/732251437/Practica-Rm-Cobol>

https://www.alegsa.com.ar/Dic/acceso_secuencial.php#gsc.tab=0