

Universidad de Guadalajara



Centro Universitario de Ciencias Exactas e Ingeniería

División de Tecnologías para la Integración Ciber-humana

Materia: Sistemas Operativos

Profesor: Violeta Del Rocío Becerra Velázquez

Alumno: Denice Estefania Rico Morones

Código: 219421171

Carrera: Ingeniería en Computación

Sección: D04

Título: Ent-Sal

Fecha: 10/11/2024

Contenido

Quiz:	3
Manejo de archivos y directorios	4
Tablas de archivos abiertos	5
Bloqueo de archivos	6
Indices	7
Dispersión (Hashing)	7
Métodos de dispersión	8
Resolución de colisiones	8
Tabla de paginas	9
Buffers: importancia y manejo	10
Conclusión:	10
Bibliografía:	11

Quiz:

Summary of your previous attempts

Attempt	State	Marks / 4.00	Grade / 10.00	Review
1	Finished Submitted Sunday, 29 September 2024, 6:38 PM	4.00	10.00	Review
2	Finished Submitted Sunday, 29 September 2024, 6:44 PM	2.00	5.00	Review
3	Finished Submitted Sunday, 29 September 2024, 6:48 PM	4.00	10.00	Review

Highest grade: 10.00 / 10.00.

Manejo de archivos y directorios

Las principales operaciones para manejar archivos en el sistema operativo incluyen:

1. Eliminar: elimina archivos del directorio y libera espacio en el dispositivo.
2. Abrir: El sistema comprueba si el archivo existe y permite el acceso según el modo (lectura, escritura, etc.). Cuando se abre, se asigna un descriptor de archivo para identificar la relación entre el proceso y el archivo.
3. Cerrar: Indica que el proceso ha terminado de usar el archivo. El sistema escribe el búfer en el disco y elimina el procesamiento de archivos de la tabla activa.
4. Leer: Copie el segmento del archivo al portapapeles. La función ``fread()`` en C permite leer una determinada cantidad de caracteres.
5. Escribir: Utilice ``fwrite()`` en C para escribir datos en el archivo desde el principio o el final.
6. Mover: Utilice `"fseek()"` en C para establecer el puntero de lectura/escritura en una nueva ubicación en el archivo.

Estas operaciones son comunes en los sistemas operativos y permiten procesar archivos secuencialmente como si fueran cintas moviéndose hacia adelante o atrás para leer o escribir en diferentes puntos del archivo.

Tablas de archivos abiertos

En un sistema operativo, el propio sistema y cada proceso mantienen una tabla de archivos abiertos, estas tablas almacenan información sobre todos los archivos en uso y le dan a cada proceso un identificador de archivo para usar en lugar de su nombre, esto permite que múltiples procesos abran el mismo archivo simultáneamente, aunque el acceso y las posibles operaciones dependen de la semántica del sistema.

Razones para admitir tablas de archivos en dos niveles:

Sistema operativo:

- Recuento de usuarios de archivos: controla operaciones como deshabilitar particiones o eliminar el acceso a archivos , evitando que se realicen operaciones mientras está en uso.
- Modo de acceso: evita inconsistencias durante la edición, como evitar que dos procesos escriban el mismo archivo simultáneamente.
- Ubicación del disco: la ubicación donde se almacena el archivo para que cada proceso no tenga que acceder a la ubicación para cada operación.
- Bloque de datos: Establezca el bloqueo para evitar modos de acceso conflictivos.

Proceso:

- Manejo de archivos: asocia el nombre del archivo con un identificador único utilizado por el proceso. Cada proceso tiene su propio procesamiento del mismo archivo.
- Permisos: define permisos específicos del proceso, que pueden ser un subconjunto de los permisos generales del archivo.

Bloqueo de archivos

Los archivos se pueden utilizar como mecanismo de comunicación entre procesos no relacionados, incluso a lo largo del tiempo. En un sistema multitarea, acceder a un archivo es tan simple como especificar el nombre del archivo o la ruta, por lo que se implementa un mecanismo de bloqueo para evitar que procesos concurrentes interactúen y corrompan el archivo.

Algunos sistemas permiten bloquear ciertas áreas de un archivo. archivo, aunque la práctica común es bloquearlo por completo.

Tipo de bloqueo:

- Bloqueo compartido: permite la lectura simultánea de archivos. Varios procesos pueden adquirirlo al mismo tiempo, lo que significa que el archivo no se modificará mientras el bloqueo esté vigente.
- Bloqueo exclusivo: solo lo puede mantener un proceso, cuyo proceso puede modificar el archivo o parte del archivo, dependiendo

Hay dos mecanismos de bloqueo:

- Bloqueo forzado: cuando el proceso obtiene un bloqueo forzado. Cuando, el sistema impone restricciones de acceso a todos los procesos, independientemente de si están destinados a estar bloqueados.
- Bloqueo recomendado: este tipo de bloqueo depende de la comunicación entre procesos; su aplicación e implementación depende del desarrollador de cada proceso involucrado.

Desde una perspectiva de concurrencia, mecanismos como mutex, semáforos o variables de estado son opcionales, mientras que los monitores son obligatorios. No todos los sistemas operativos permiten las cuatro combinaciones de bloqueo posibles (universal forzado, universal recomendado, exclusivo forzado y exclusivo recomendado), normalmente, Windows utiliza un bloqueo obligatorio, mientras que Unix utiliza un bloqueo de asesoramiento.

El uso de bloqueos de archivos requiere precaución, similar a la administración de recursos. Dos procesos que intentan bloquear dos archivos en modo exclusivo pueden bloquearse, al igual que con otros recursos.

Indices

Un índice es una estructura de datos que permite un acceso rápido a registros de bases de datos o elementos de recopilación de datos sin atravesar toda la estructura.

Dispersión (Hashing)

Hashing es una técnica que convierte una clave o valor en una posición única en una tabla hash. Se utiliza comúnmente para implementar diccionarios, bases de

datos y cachés. Una función hash toma una clave y la convierte en un índice de tabla, permitiendo el acceso directo al valor almacenado en esa ubicación.

Métodos de dispersión

- **Función Módulo:** una forma directa de obtener deseado porcentaje de la clave podría ser identificar el resto de la clave después de la división por un número. Por ejemplo, para una clave de 7259 y $M = 63$, el índice resultante es igual a 27 como $M = 64-1$ o 63.
- **Función Cuadrado:** elevar al cuadrado el valor numérico entero c y seleccionar los bits centrales del resultado. Por ejemplo, para la clave de 5678, su cuadrado es 32239684 y sus dígitos centrales son iguales a 39.
- **Función Plegado o de Compresión:** dividir partes al número y luego realizar cualquier otra operación entre ellos, como XOR. Este último caso es particularmente útil cuando se utilizan claves alfanuméricas. Aquí tenemos un ejemplo para la clave "UNED", convirtiendo todos los caracteres en dos bits binarios, que luego se someten a una operación XOR. Como resultado, obtendremos un solo número representando un índice.
- **Función Multiplicación:** esta función normaliza su clave, multiplicándola por una constante, con el siguiente paso que la función está multiplicando por el tamaño de la tabla y luego devolviendo la parte que no es fracciones de la operación producto. Por ejemplo, asumiendo que el valor es 3, con la constante que se iguala a 0.618 y el tamaño de tabla igual a 1000, el índice obtenido es igual a 854.

Resolución de colisiones

Open Hashing: utiliza una estructura dinámica fuera de la tabla hash (como una lista vinculada) para almacenar claves en conflicto. Si bien es conceptualmente agradable, acceder y buscar en estas estructuras externas puede resultar ineficiente debido al recorrido lineal requerido.

Hash cerrado: los conflictos se resuelven buscando ubicaciones alternativas en la misma tabla hasta que se encuentre espacio libre. Esto requiere colocar una bandera en ubicaciones vacías para marcarlas. Sin embargo, a medida que la tabla se llena, aumenta la probabilidad de conflictos, lo que afecta su desempeño.

Los principales métodos para resolver conflictos en tablas hash son:

- Recorrido lineal: según la fórmula $s(k, i) = (h(k) + i) \bmod m$, agregue un incremento lineal a la dirección obtenida por la función hash. m , donde m es el tamaño de la mesa. Esto le permite avanzar repetidamente hasta encontrar un espacio vacío.
- Recorrido cuadrático: utilice la expresión cuadrática $g(i) = i^2$ para eliminar mejor las colisiones. La nueva dirección se calcula como $dir = (h(k) + c * g(i)) \bmod m$, donde c es una constante e i es el número de colisiones para la clave k . Este método reduce los conflictos mejor que el recorrido lineal.
- Doble hash: utilizando una segunda función hash auxiliar h' , la dirección se calcula como $dir = (h(k) + c * h'(k)) \bmod m$, donde c es una constante. Para evitar conflictos, la función h' debe ser diferente de h y producir valores relativamente simples para m . Si m es prima, entonces cualquier función puede actuar como h' .

Tabla de paginas

El sistema operativo mantiene una tabla de páginas para mostrar la ubicación del marco por cada página del proceso y está conformada por un número de páginas y un desplazamiento dentro de esta. Con la paginación el procesador debe conocer cómo acceder a la tabla del proceso actual esto se hace gracias a una dirección lógica (**número de página, desplazamiento**), y el procesador utiliza esta tabla para producir una dirección física (**número de marco, desplazamiento**).

Podemos resumir esto a que en dicha tabla indicamos el número de página, el proceso que se ha asignado y los atributos de la misma administración también se puede indicar si la memoria es compartida y los datos para los algoritmos de intercambio de almacenamiento secundario y el valor para el registro base del CPU; si se desea hacer cualquier operación de memoria y en otra página diferente debemos de modificar el valor en el registro de su tabla de páginas.

Con la tabla de páginas podemos traducir los identificadores de página los cuales son utilizados por programas en sus direcciones lógicas y por las direcciones de marco de página esto al usarse en las direcciones físicas al momento de querer acceder a la memoria, Otras cosas a las cuales también podemos acceder son las siguientes:

- Una bandera para indicar si ha sido referenciada.
- Una bandera para indicar si la página ha modificado su contenido desde La última vez que se ha escrito.
- Información de protección lo cual es una serie de bits para controlar los aspectos ya sea si queremos que la página sea solamente para lectura, si se trata de un código de programa, si permitimos modificaciones o si será de memoria compartida.

Buffers: importancia y manejo

los buffers al ser partes temporales de almacenamiento mientras existen un intercambio de datos entre diferentes partes de un sistema son de suma importancia si queremos mejorar la eficiencia y estabilidad en el procesamiento de los datos; los buffers realizan sincronización entre las velocidades para poder trabajar juntos sin interrupciones, además de reducir la latencia, evitar la pérdida de datos y la eficiencia en operaciones ya sean de entrada/salida.

- Determinar el tamaño óptimo del buffer.
- Es muy común que los buffers funcionen con estructuras FIFO.
- Utilizar mecanismos de sincronización es muy importante cuando múltiples procesos acceden al buffer.
- Es importante liberar el espacio del buffer para evitar los problemas de rendimiento o overflow.

Conclusión:

Cómo conclusión puedo decir que esta investigación es de gran importancia para conocer los métodos que podemos implementar para la resolución de colisiones que es un tema muy importante a considerar, ya que es importante tener una buena implementación y rendimientos; además, el manejo de archivos es muy importante para poder tomar las rutas mas sencillas al momento de manejar y crear archivos tanto en las entradas y salidas de igual manera.

Bibliografía:

Stallings, W. (2005). *Sistemas operativos. Aspectos internos y principios de diseño*. Madrid: Pearson Education.

Demonics. (2016, 24 octubre). Tablas de dispersión (Hash). Aprende y Programa. <https://aprendeyprogramablog.wordpress.com/2016/10/24/tablas-de-dispersion-hash/>

Wolf, G. (s. f.). Sistemas Operativos — Sistemas de archivos. https://sistop.gwolf.org/html/05_sistemas_de_archivos.html