

SISTEMAS OPERATIVOS – PARCIAL 3

GESTIÓN DE ARCHIVOS

FICHEROS Y SISTEMAS DE FICHEROS

El sistema de ficheros permite al usuario crear colecciones de datos, llamadas ficheros, con propiedades deseables, tales como las siguientes:

- Existencia a largo plazo: los ficheros se almacenan en almacenamiento secundario y no desaparecen cuando el usuario se desconecta.
- Compartible entre procesos: pueden tener permisos de acceso asociados que permitan controlar la compartición.
- Estructura: un fichero puede tener una estructura interna que es conveniente para aplicaciones particulares. Adicionalmente, los ficheros se pueden organizar en estructuras jerárquicas o más complejas para reflejar las relaciones entre los mismos.

Un sistema de ficheros proporciona una colección de funciones que se pueden llevar a cabo sobre ficheros:

- **Crear:** se define un nuevo fichero y se posiciona dentro de la estructura de ficheros.
- **Borrar:** se elimina un fichero de la estructura de ficheros y se destruye.
- **Abrir:** un fichero se declara 'abierto' por un proceso, permitiendo al proceso realizar funciones sobre el fichero.
- **Cerrar:** un proceso cierra un fichero de forma que no puede volver a realizar funciones sobre él, salvo abrirlo de nuevo.
- **Leer:** un proceso lee de un fichero todos sus datos o parte de ellos.
- **Escribir:** un proceso actualiza un fichero, modificando su contenido.

ESTRUCTURA DE UN FICHERO

Un **campo** es el elemento básico de los datos. Se caracteriza por su longitud y el tipo de datos.

Un **registro** es una colección de campos relacionados que pueden tratarse como una unidad por una aplicación.

Un **fichero** es una colección de campos similares. El fichero se trata como una entidad única por parte de los usuarios y las aplicaciones.

Una **base de datos** es una colección de datos relacionados. Una base de datos está formada por uno o más ficheros.

SISTEMAS DE GESTIÓN DE FICHEROS

Un sistema de gestión de ficheros es aquel conjunto de software del sistema que proporciona servicios a los usuarios y aplicaciones en el uso de ficheros.

Arquitectura de un sistema de ficheros

En el nivel más bajo, los **manejadores de dispositivos** se comunican directamente con los dispositivos periféricos o sus controladores o canales.

Un controlador de dispositivo es el responsable de iniciar las operaciones de E/S de un dispositivo y procesar la finalización de una petición de E/S. Para las operaciones sobre ficheros, los dispositivos típicos son los discos y las cintas.

El siguiente nivel se denomina **sistema de ficheros básico** o nivel de **E/S físico**. Es la interfaz primaria con el entorno fuera del sistema de computación.

El **supervisor de E/S básico** se encarga de todas las inicializaciones y finalizaciones de E/S. También se encarga de la planificación de disco y cinta para optimizar el rendimiento.

La **E/S lógica** permite a los usuarios y las aplicaciones acceder a los registros.

El nivel del sistema de ficheros más cercano al usuario es frecuentemente denominado **método de acceso**. Proporciona una interfaz estándar entre las aplicaciones y los sistemas de ficheros y dispositivos que contienen los datos.

ORGANIZACIÓN Y ACCESO A LOS FICHEROS

El término **organización de fichero** hace referencia a la estructura lógica de los registros determinados por la forma en la que se acceden.

Para escoger una organización de ficheros, son importantes varios criterios:

- Tiempo de acceso corto.
- Facilidad de actualización.
- Economía de almacenamiento.
- Mantenimiento sencillo.
- Fiabilidad.

A continuación, se describen cinco organizaciones de ficheros fundamentales.

LA PILA

En esta organización, los datos se almacenan en el orden en el que llegan. Cada registro está formado por un conjunto de datos. El propósito de la pila es acumular la masa de datos y guardarlos.

En esta organización, cada campo debe ser autodescriptivo, incluyendo el nombre del campo y el valor. La longitud de cada campo debe ser implícitamente indicada por delimitadores, explícitamente incluida como un subcampo o conocida por defecto para cada tipo de campo.

En esta organización, el acceso a los registros se hace mediante búsqueda exhaustiva.

EL FICHERO SECUENCIAL

La forma más común de estructura de fichero es el fichero secuencial. Se usa un formato fijo para los registros.

El primer campo de cada registro se llama **campo clave**. El campo clave identifica de forma única el registro, por tanto, los valores de la clave de diferentes registros son siempre diferentes. Los registros se almacenan en secuencia según la clave.

EL FICHERO SECUENCIAL INDEXADO

El fichero secuencial indexado mantiene las características clave del fichero secuencial: los registros se organizan en secuencia, basándose en un campo clave. Pero se añaden dos características:

- Un índice al fichero que da soporte al acceso aleatorio.
- Un fichero de desbordamiento.

En la estructura secuencial indexada más sencilla, se usa un único nivel de indexación. El índice en este caso es un fichero secuencial simple. Cada registro del fichero índice está formado por dos campos: un campo clave, que es el mismo campo clave del fichero principal y un puntero al fichero principal. Para encontrar un campo específico, se busca el índice que contenga el mayor valor clave que sea igual o preceda al valor de clave deseado. La búsqueda continua en el fichero principal en la ubicación indicada por el puntero.

El fichero secuencial indexado reduce enormemente el tiempo requerido para acceder a un único registro, sin sacrificar la naturaleza secuencial del fichero.

EL FICHERO INDEXADO

El fichero secuencial solo permite buscar un registro por su campo clave. Para lograr la flexibilidad que no obligue a buscar por el campo clave, necesitamos una estructura que emplea múltiples índices, uno por cada tipo de campo que puede estar sujeto a una búsqueda. Permite utilizar registros de longitud variable.

Se usan dos tipos de índice: un índice exhaustivo contiene una entrada por cada registro del fichero principal. Un índice parcial contiene entradas a registros donde el campo de interés existe.

Cuando se añade un nuevo registro al fichero principal, todos los ficheros índices deben actualizarse.

EL FICHERO DE ACCESO DIRECTO O HASH

El fichero de acceso directo, explota la capacidad encontrada en los discos para acceder directamente a cualquier bloque en una dirección conocida. Se requiere una clave para cada registro.

El fichero directo hace uso de una función *hash* sobre su valor clave.

Los registros son de tamaño fijo y se acceden de uno en uno.

DIRECTORIOS

CONTENIDO

El directorio contiene información sobre los ficheros, incluyendo atributos, ubicación y propiedad. Gran parte de esta información, la gestiona el S.O.

El directorio es un fichero, accesible por varias rutinas de gestión de ficheros.

ESTRUCTURA

La forma en la que la información se almacena difiere ampliamente entre varios sistemas. Parte de la información se puede almacenar en un registro cabecera asociado con el fichero. Esto reduce la cantidad de almacenamiento requerido para el directorio,

haciendo más fácil almacenar el directorio o parte del directorio en memoria principal, a fin de incrementar la velocidad.

La forma más sencilla de estructura de estructura para un directorio es una lista de entradas, una por cada fichero.

Los directorios se almacenan en forma de árbol, con una estructura jerárquica. Hay un directorio maestro, que tiene bajo dicho directorio varios directorios de usuario. Cada uno de estos directorios de usuario a su vez pueden tener subdirectorios y ficheros como entradas.

El enfoque más sencillo de organización de cada directorio o subdirectorio es almacenando cada directorio como un fichero secuencial. Si los directorios tuvieran un gran número de entradas, dicha organización llevaría a unos tiempos de búsqueda innecesariamente largos. En dicho caso, es preferible una estructura *hash*.

NOMBRADO

Los usuarios necesitan poder referenciar un fichero mediante un nombre simbólico. Claramente, cada fichero en el sistema debe tener un nombre único a fin de que las referencias al mismo no sean ambiguas.

Cualquier fichero del sistema se puede localizar siguiendo un camino desde el directorio raíz y bajando por las ramas hasta alcanzar el fichero.

El conjunto de nombres de directorios, finalizando en el nombre del fichero, constituye un **nombre de camino** para el fichero.

Un usuario interactivo o un proceso está asociado con un directorio actual, que se suele denominar **directorio de trabajo**.

COMPARTICIÓN DE FICHEROS

En un sistema multiusuario existe casi siempre el requisito de permitir que los ficheros se compartan entre varios usuarios.

DERECHOS DE ACCESO

El sistema de ficheros debería proporcionar una herramienta flexible para permitir la compartición de ficheros extensiva entre los usuarios.

Esta es la lista de los posibles derechos de acceso que se asigna a un usuario particular para un determinado fichero:

- Ninguno.
- Conocimiento.
- Ejecución.
- Lectura.
- Adición.
- Actualización.
- Cambio de protección.
- Borrado.

Se pueden proporcionar diferentes accesos a distintas clases de usuarios.

- Usuario específico.
- Grupo de usuarios.

- Todos.

ACCESO SIMULTÁNEO

Cuando se garantiza acceso de adición o actualización de un fichero a más de un usuario, el sistema operativo o sistema de gestión de ficheros debe forzar una disciplina.

Una técnica de fuerza bruta consiste en permitir al usuario bloquear el fichero completo cuando se va a actualizar.

BLOQUES Y REGISTROS

Los registros son las unidades lógicas de acceso de un fichero estructurado, mientras que los bloques son las unidades de E/S con almacenamiento secundario. Para que la E/S se pueda realizar, los registros se deben organizar como bloques.

En general, los bloques tienen un tamaño fijo, esto simplifica la E/S, la asignación de *buffers* en memoria principal y la organización de bloques en almacenamiento secundario.

Cuanto mayor sea el bloque, más registros se transferirán en una operación de E/S. dado el tamaño de un bloque, se pueden utilizar tres métodos:

- **Bloques fijos:** se usan registros de longitud fija y se almacenan en un bloque un número integral de registros. Puede haber fragmentación interna.
- **Bloques expandidos de longitud variable:** se usan registros de longitud variable y se empaquetan en bloques sin dejar espacio no utilizado. Por tanto, algunos registros deben expandirse a lo largo de dos bloques, con su continuación indicada por un puntero al bloque sucesor.
- **Bloques no expandidos de longitud variable:** se usan registros de longitud variable, pero no se emplea expansión. Hay espacio malgastado en la mayoría de los bloques debido a la incapacidad para utilizar el resto de un bloque si el siguiente registro es mayor que el espacio no utilizado restante.

GESTIÓN DE ALMACENAMIENTO SECUNDARIO

En almacenamiento secundario, un fichero está compuesto por una colección de bloques. El S.O. es responsable de asignar bloques a los ficheros.

ASIGNACIÓN DE FICHEROS

Hay varios aspectos que están involucrados en la asignación de ficheros:

- Cuando se crea un fichero nuevo, ¿se asigna de una vez el espacio máximo requerido para el fichero?
- El espacio se asigna a un fichero como una o más unidades contiguas, lo que se denomina porción. El tamaño de una porción puede ir desde un único bloque al fichero completo. ¿Qué tamaño de porción debería utilizarse para asignación de ficheros?
- ¿Qué clase de estructura de datos o tabla se utiliza para guardar traza de las porciones asignadas para un fichero? Un ejemplo de dicha estructura es una **tabla de asignación de ficheros** (FAT), encontrado en DOS y otros sistemas.

Preasignación frente a asignación dinámica

Una política de preasignación requiere que el tamaño máximo de un fichero sea declarado en tiempo de creación de fichero.

En algunas situaciones es difícil o imposible estimar fiablemente el tamaño máximo potencial del fichero.

Hay ventajas por tanto en el uso de gestión dinámica, que asigna espacio a un fichero en porciones cuando se necesite.

Tamaño de porción

Para escoger un tamaño de porción debe existir un compromiso entre la eficiencia desde el punto de vista de un único fichero y la eficiencia del sistema completo.

Existen dos alternativas principales:

- **Porciones variables, grandes y contiguas:** esta alternativa proporciona el mejor rendimiento. El tamaño variable evita malgastar espacio, y las tablas de asignación de ficheros son pequeñas. Sin embargo, el espacio es difícil de reutilizar.
- **Bloques:** pequeñas porciones fijas proporcionan mayor flexibilidad. Podrían requerir grandes tablas o estructuras complejas para su asignación. La contigüidad ha sido abandonada como meta primaria. Los bloques se asignan según se necesite.

Cada opción es compatible con preasignación o asignación dinámica.

Con porciones de tamaño variable, es necesario preocuparse de la fragmentación del espacio libre. Las siguientes estrategias son posibles:

- **Primer ajuste:** escoger el primer grupo contiguo no utilizado de bloques de tamaño suficiente desde una lista de bloques libres.
- **Siguiente ajuste:** escoger el grupo más pequeño no utilizado que sea de suficiente tamaño.
- **Ajuste más próximo:** escoger el grupo no utilizado de tamaño suficiente que sea más cercano a la asignación previa para el fichero de manera que se incremente la proximidad.

Métodos de asignación de ficheros

Con **asignación contigua** se asigna un único conjunto contiguo de bloques en tiempo de creación de ficheros. Por tanto, hay una estrategia de preasignación que utiliza porciones de tamaño variable. La tabla de asignación de ficheros necesita solo una entrada para cada fichero, mostrando el bloque inicial y la longitud del fichero.

La **asignación encadenada** se realiza a nivel de bloques individuales. Cada bloque contiene un puntero al siguiente bloque de la cadena. La tabla de asignación de ficheros necesita solo una entrada para cada fichero, mostrando el bloque inicial y la longitud del fichero. Aunque la preasignación es posible, es más común asignar bloques cuando se necesita. La selección de bloques es ahora una cuestión sencilla: cualquier bloque libre se puede añadir a una cadena. No hay fragmentación externa.

En la **asignación indexada**, la tabla de asignación de ficheros contiene un índice separado de un nivel por cada fichero. El índice tiene una entrada por cada porción

asignada al fichero. La asignación puede realizarse mediante bloques de tamaño fijo o porciones de tamaño variable. La asignación por bloques elimina la fragmentación externa mientras que la asignación por porciones de tamaño variable mejora la proximidad.

GESTIÓN DE ESPACIO LIBRE

Necesitamos una **tabla de asignación de disco** que nos diga que bloques están disponibles en el disco.

Tablas de bits

Este método usa un vector que está formado por un bit por cada bloque en disco. Cada bit a 0 corresponde con un bloque libre y cada bit a 1 corresponde con un bloque en uso.

Porciones libres encadenadas

Las porciones libres se pueden encadenar usando un puntero y valor de longitud en cada porción libre. Este método tiene una sobrecarga de espacio insignificante, porque no se necesita una tabla de asignación de disco, sino simplemente un puntero al comienzo de la cadena y la longitud de la primera porción.

Este método tiene como inconveniente que después de cierto uso, el disco se quedará bastante fragmentado y muchas porciones serán de la longitud de un único bloque.

Indexación

Esta técnica trata el espacio libre como un fichero y utiliza una tabla de índices. Hay una entrada en la tabla de índices por cada porción libre en disco.

Lista de bloques libres

A cada bloque le asignamos un número secuencialmente y la lista de los números de todos los bloques libres se mantiene en una porción reservada del disco.

La lista de bloques libres es muy grande por lo que debe almacenarse en disco.

FIABILIDAD

Para prevenir errores, cuando se solicita una asignación de ficheros, se pueden llevar a cabo los siguientes pasos:

- Bloquear la tabla de asignación de disco en disco. Esto previene a otro usuario de causar alteraciones a la tabla hasta que esta asignación se complete.
- Buscar espacio disponible en la tabla de asignación de disco. Esto supone que una copia de la tabla de asignación de disco siempre se guarda en memoria principal. Si no, debe primero traerse a disco.
- Asignar espacio, actualizar la tabla de asignación de disco y actualizar el disco. Actualizar el disco supone escribir la tabla de asignación de disco en disco. Para la asignación de disco encadenado, también supone actualizar algunos punteros en disco.
- Actualizar la tabla de asignación de disco y actualizar el disco.
- Desbloquear la tabla de asignación de disco.