

Ficheros/Archivos



Un archivo o fichero informático es un **conjunto de bits almacenado en un dispositivo**.

Un archivo es identificado por un nombre y la descripción de la carpeta o directorio que lo contiene.

Los archivos informáticos se llaman así porque son los equivalentes digitales de los archivos en tarjetas, papel o microfichas del entorno de oficina tradicional.

Los archivos informáticos facilitan una manera de organizar los recursos usados para almacenar permanentemente datos en un sistema informático.

Un archivo informático normalmente consiste de paquetes más pequeños de datos (a menudo llamados **registros** o líneas) que son individualmente diferentes pero que comparten algún rasgo en común.

Por ejemplo, un archivo de nóminas puede contener datos sobre todos los empleados de una empresa y los detalles de su nómina; cada registro del archivo de nóminas se refiere únicamente a un empleado, y todos los registros tienen la característica común de estar relacionados con las nóminas- esto es muy similar a colocar todos los datos sobre nóminas en un archivador concreto en una oficina.

Un archivo de texto puede contener líneas de texto, correspondientes a líneas impresas en una hoja de papel.

La manera en que se agrupan los datos en un archivo depende completamente de la persona que diseñe el archivo.

Esto ha conducido a múltiples estructuras de archivo más o menos estandarizadas para todos los propósitos imaginables, desde los más simples a los más complejos.

La mayoría de los archivos informáticos son usados por programas de computadora. Estos programas crean, modifican y borran archivos para su propio uso bajo demanda. Los programadores que crean los programas deciden qué archivos necesitan, cómo se van a usar, y (a menudo) sus nombres o tipos.

Archivo binario:

Estructura de datos permanente compuesto por **registros (filas)** y éstos a su vez por campos (columnas). Se caracteriza por tener un tipo de dato asociado, el cual define su estructura interna.

Archivo de texto:

Estructura de datos permanente no estructurado formado por una secuencia de caracteres ASCII.

Registro: paquete más pequeños de datos que es individualmente diferente pero que comparte algún rasgo en común con los demás.

Ej: datos de un alumno, del registro de una página web, de un paciente, de la nómina de alguien,...

Organización de ficheros

- Secuenciales.
 - Lineales.
 - Los registros se almacenan físicamente de forma contigua (uno a continuación de otro) siguiendo la secuencia lógica del fichero.
Orden físico = Orden lógico
 - Todas las operaciones que se realizan sobre el fichero se hacen según esta secuencia.
 - Es la única que admite un soporte físico de acceso secuencial no direccionable.
 - Encadenados.
 - Los registros se almacenan según se escriben, pero el orden lo define la posición del registro siguiente que se almacena en cada uno.
Orden físico distinto de Orden lógico
 - Todas las operaciones que se realizan sobre el fichero se hacen según esta secuencia.
 - Necesita un soporte físico de acceso direccionable.

Organización de ficheros

- Directos.

Cada registro tiene un tamaño fijo y está en una posición del fichero.

- Por posición.

Accedemos al lugar dando la posición.

¡Vete al registro 34!

- Por clave. (HASH)

Mediante una fórmula traducimos la clave por un número.

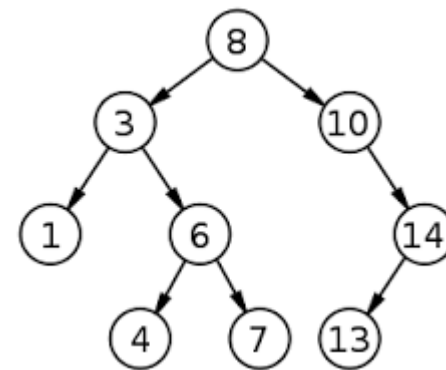
Necesitamos ocupar todo el espacio desde el principio.

Puede haber repeticiones.

- Indexados.

- ISAM (Indexed Sequential Access Mode).

- C-ISAM (Chained ISAM).



Tipos de Acceso a los Archivos

a.)Secuencial:

Se acceden uno a uno los registros desde el primero hasta el último o hasta aquel que cumpla con cierta condición de búsqueda. Se permite sobre archivos de Organización secuencial y Secuencial Indexada.

b.)Random:

Se acceden en primera instancia la tabla de índices de manera de recuperar la dirección de inicio de bloque en donde se encuentra el registro buscado. (dentro del área primaria o de overflow). Se permite para archivos con Organización Secuencial Indexada.

c.)Dinámico (Secuencial y Random):

Se acceden en primera instancia la tabla de índices de manera de recuperar la dirección de inicio de bloque en donde se encuentra el registro buscado o el siguiente al actual (dentro del área primaria o de overflow). Se permite para archivos con Organización Secuencial Indexada.

d.)Directo:

Es aquel que utiliza la función de Hashing para recuperar los registros. Sólo se permite para archivos con Organización Relativa.

Archivos de organización secuencial

Operaciones :

- Añadir. Sólo es posible escribir al final del fichero.
- Consulta. Se realiza en orden secuencial.
- Actualización. (Inserción, eliminación, modificación) generando de nuevo el archivo.

Un caso especial: Fichero de texto

- Utilizado para almacenar textos.
- Registros de tamaño variables denominados “líneas”. Cada línea almacena una cadena de
- Delimitador de registros: “EOL” (End Of Line)
- Actualizaciones mediante reescritura de todo el archivo.

Recorrido en C

```
#include <stdio.h>

int main()
{
    FILE *fichero;
    char letra;

    fichero = fopen("origen.txt","r");
    if (fichero==NULL)
    {
        printf( "No se puede abrir el fichero.\n" );
        getchar();
        return( 1 );
    }
    printf( "Contenido del fichero:\n" );
    letra=getc(fichero);
    while (feof(fichero)==0)
    {
        printf( "%c",letra );
        letra=getc(fichero);
    }
    if (fclose(fichero)!=0)
        printf( "Problemas al cerrar el fichero\n" );
    getchar();
}
```

Copiar archivo en C

```
#include <stdio.h>

int main()
{
    FILE *origen, *destino;
    char letra;

    origen=fopen("origen.txt","r");
    destino=fopen("destino.txt","w");
    if (origen==NULL || destino==NULL)
    {
        printf( "Problemas con los ficheros.\n" );
        getchar();
        return( 1 );
    }
    letra=getc(origen);
    while (feof(origen)==0)
    {
        putc(letra,destino);
        printf( "%c",letra );
        letra=getc(origen);
    }
    if (fclose(origen)!=0)
        printf( "Problemas al cerrar el fichero origen.txt\n" );
    if (fclose(destino)!=0)
        printf( "Problemas al cerrar el fichero destino.txt\n" );
    getchar();
}
```

Organización directa o aleatoria

- Existe una transformación conocida que genera la dirección de cada registro dentro del fichero a partir de una clave.
- El problema fundamental es la elección de dicha transformación o método de direccionamiento.
- Pueden aparecer dos situaciones no deseadas:
 - Direcciones que no corresponden a ninguna clave.
 - Direcciones que corresponden a varias claves.

Tres métodos usuales de direccionamiento:

- Direccionamiento directo.

- La dirección relativa es la propia clave (debe ser numérica y de rango igual al tamaño del fichero).

- Direccionamiento indexado (por clave).

- Cada clave tiene asociada una dirección en una tabla.
- Al añadir nuevos registros las claves se colocan al final de la tabla
- La tabla está desordenada, lo cual puede hacer que la recuperación de la posición no sea rápida. Para hacerlo más rápido se puede tener la tabla ordenada y almacenada en estructuras dinámicas (árboles binarios por ejemplo) o almacenada en memoria principal.

- Direccionamiento calculado (Hashing).

- Se utilizan técnicas de Hashing.

Operaciones:

- Creación.
 - Se debe reservar espacio en disco.
- Consulta.
 - Se realiza por clave.
 - Si procede hay que tratar sinónimos
- Borrado.
 - Borrado lógico. Se puede reutilizar el espacio del registro
- Modificación e Inserción
 - Siempre se pueden hacer, realizando la transformación de clave correspondiente.

Carga datos archivo directo en C

```
#include <stdio.h>
#include <string.h>
```

```
struct {
    char nombre[20];
    char apellido[20];
    char telefono[15];
} registro;
```

```
int main()
{
    FILE *fichero;
    fichero = fopen( "nombres.txt", "a" );
    do {
        printf( "Nombre: " ); fflush(stdout);
        gets(registro.nombre);
        if (strcmp(registro.nombre,""))
        {
            printf( "Apellido: " ); fflush(stdout);
            gets(registro.apellido);
            printf( "Teléfono: " ); fflush(stdout);
            gets(registro.telefono);
            fwrite( &registro, 1, sizeof(registro), fichero );
        }
    } while (strcmp(registro.nombre,"")!=0);
    fclose( fichero );
}
```

Recorrido archivo directo en C

```
#include <stdio.h>

struct {
    char nombre[20];
    char apellido[20];
    char telefono[15];
} registro;

int main()
{
    FILE *fichero;

    fichero = fopen( "nombres.txt", "r" );
    while (!feof(fichero)) {
        if (fread( &registro, sizeof(registro), 1, fichero )) {
            printf( "Nombre: %s\n", registro.nombre );
            printf( "Apellido: %s\n", registro.apellido);
            printf( "Teléfono: %s\n", registro.telefono);
        }
    }
    fclose( fichero );
    getchar();
}
```


Búsqueda aleatoria archivo directo en C

```
#include <stdio.h>
struct {
    char nombre[20];
    char apellido[20];
    char telefono[15];
} registro;
int main()
{
    FILE *fichero;
    long posicion;
    fichero = fopen( "nombres.txt", "r" );
    printf( "¿Qué posición quieres leer? " ); fflush(stdout);
    scanf( "%D", &posicion );
    fseek ( fichero , (posicion-1)*sizeof(registro) , SEEK_SET );

    if (fread( &registro, sizeof(registro), 1, fichero )) {
        printf( "En la posición %D está: \n", posicion );
        printf( "Nombre: %s\n", registro.nombre );
        printf( "Apellido: %s\n", registro.apellido);
        printf( "Teléfono: %s\n", registro.telefono);
    }
    else
        printf( "Problemas posicionando el cursor.\n" );

    fclose( fichero );
    getchar();getchar();
}
```