



**T
E
M
A

5**

Ficheros

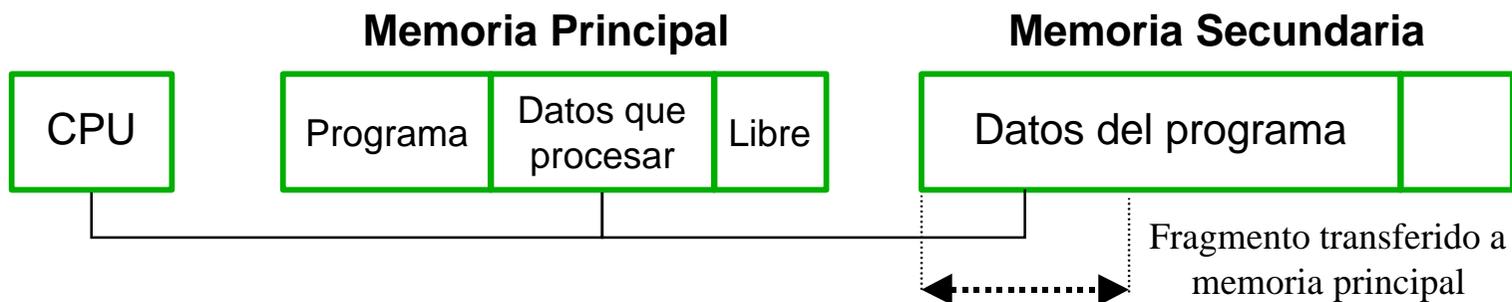
Contenido del Tema

- 5.1. Introducción.**
- 5.2. Conceptos básicos.**
- 5.3. Tipos de ficheros.**
- 5.4. Organización de ficheros.**
- 5.5. Primitivas de acceso.**
- 5.6. Ordenación externa.**



Introducción

- Necesidad de las memorias secundarias.
 - La Memoria Principal es rápida pero cara, de poca capacidad y generalmente volátil.
 - La Memoria Secundaria es lenta pero barata, de alta capacidad y no volátil.
- Para el procesamiento es necesario transferir la información a memoria principal.



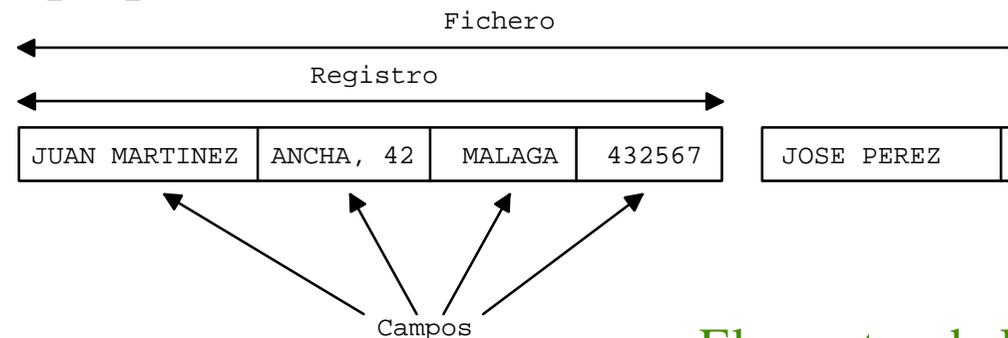
Elementos de Programación II



Conceptos básicos

Definición de fichero

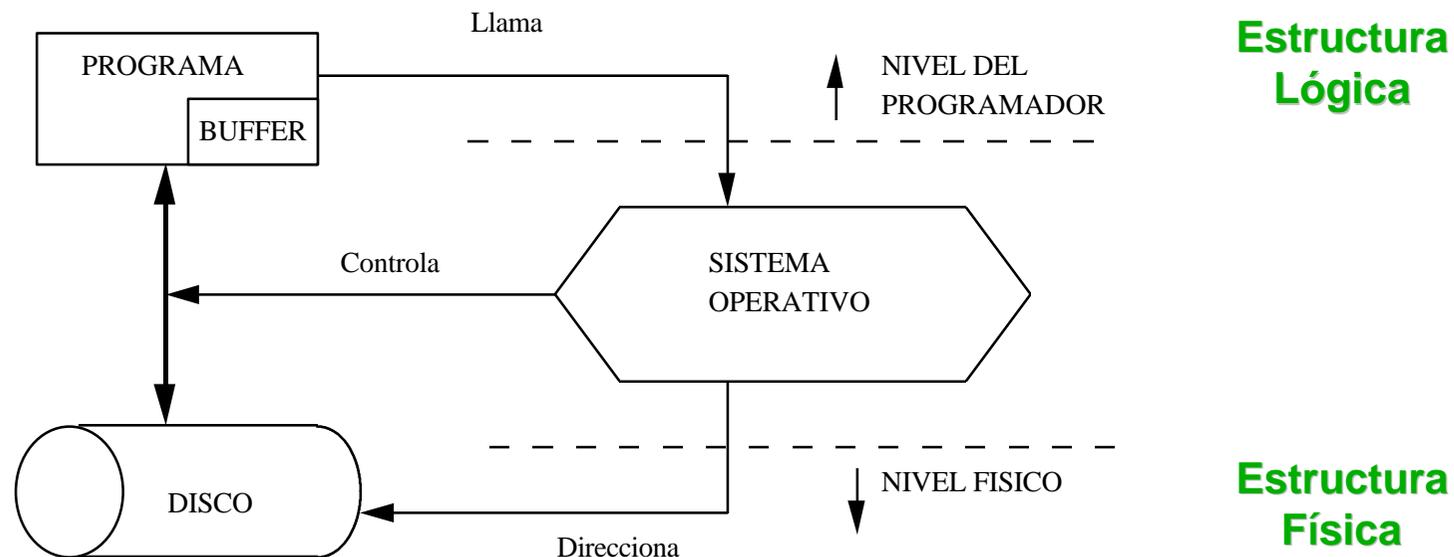
- Conjunto de información relacionada, tratada como una unidad de almacenamiento en memoria secundaria y organizada de forma estructurada para facilitar la búsqueda de datos individuales.
- Un fichero está compuesto por registros homogéneos que contienen información organizada en campos.
- campo la mínima unidad de procesamiento con significado propio.





Conceptos básicos

En un lenguaje de alto nivel, el fichero no es manejado directamente por el propio programa, sino por el Sistema Operativo. Esto facilita que los programas sean **trasladables**.





Conceptos básicos

- **Forma de acceso.** Forma en la que puede leerse o escribirse la información en un dispositivo de memoria secundaria.
 - **De acceso secuencial** (Dispositivos no direccionables).
 - **De acceso directo** (Dispositivos direccionables).
- **Dirección.** Es la forma de referenciar un bloque o registro de información dentro de un dispositivo.
 - **Dirección física.** Localización del registro en el dispositivo expresada en número de bytes.
 - **Dirección relativa.** Posición del registro respecto del principio del fichero expresada en número de bytes.



Conceptos básicos

- **Estructura lógica.** Es la forma de manipular los
 - **Cursor del fichero.** Se trata de una variable interna que contiene la dirección al registro actual del fichero.
 - **Clave o identificativo.** Campo que identifica a un registro o grupo de registros.
 - **Llave.** Cuando la clave se usa como campo de localización (dirección simbólica).
 - **Directorio.** Índice de los ficheros de un dispositivo.



Tipos de Ficheros

- Los ficheros se pueden clasificar atendiendo a distintos parámetros:
 - Por el tipo de sus registros.
 - Por la forma de almacenar los datos.
 - Por su contenido.
 -
 - Por su organización.



Tipos de Ficheros

- Por el **tipo de sus registros**:
 - **Ficheros con formato**. Tienen registros de longitud fija.
 - **Ficheros sin formato**. Tienen registros de longitud variable.
- Por la **forma de almacenamiento**:
 - **Ficheros binarios**. Almacenan la información en el mismo formato que en memoria central.
 - **Ficheros de texto**. Almacenan la información en forma de cadenas de caracteres.
- Por **su contenido**:
 - **De programa**.
 - **De datos**.



Organización de ficheros

- Hay varios motivos para estructurar la información en los ficheros:
 - Acceso rápido a los registros.
 - Facilitar la actualización de los registros.
 - Economía de almacenamiento.
 - La estructura permite reflejar la organización real de la
- Se debe, pues, optar por una u otra organización, atendiendo a la forma en que se va a usar el



Organización de ficheros

- Secuenciales.
 - Lineales.
 - Encadenados.
- Directos.
 - Por posición.
 - Por clave.
- Indexados.
 - ISAM (Indexed Sequential Access Mode).
 - C-ISAM (Chained ISAM).



Organización de ficheros

Organización secuencial lineal

- Los registros se almacenan físicamente **de forma contigua** (uno a continuación de otro) siguiendo la secuencia lógica del fichero.

Orden físico = Orden lógico

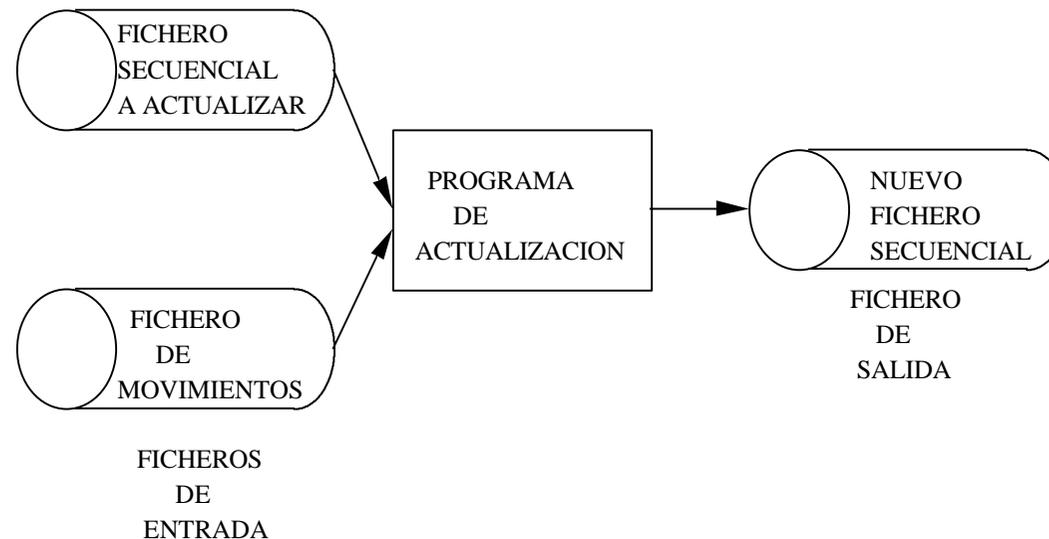
- Todas las operaciones que se realizan sobre el fichero se hacen según esta secuencia.
- Es la única que admite un **soporte físico no direccionable**.



Organización de ficheros

- **Operaciones :**

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





Organización de ficheros

- Si el fichero está en un dispositivo físico direccionable, es posible realizar actualizaciones directas, y también:
 - Consulta. Si el fichero contiene registros de longitud fija, es posible determinar la posición de comienzo de cada uno a partir de la posición relativa en el fichero.
 - Modificación. Una vez localizado un registro, se puede reescribir este en el propio fichero, siempre que al modificar el registro no aumente su longitud.
 - Borrado. No es posible eliminar un registro del fichero.
borrado lógico.



Organización de ficheros

- Una caso especial: **Fichero de texto**
 - Utilizado para almacenar textos.
 - Registros de tamaño variables denominados `registros de tamaño variable`.
Cada línea almacena una cadena de caracteres.
 - Delimitador de registros: **“EOL” (End Of Line)**
 - Los lenguajes de programación ofrecen primitivas para reconocer e insertar el delimitador.



Organización de ficheros

Organización directa o aleatoria

- Existe una transformación conocida que genera la dirección de cada registro dentro del fichero a partir
- 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 llave.
 -



Organización de ficheros

- Tres métodos fundamentales de **direccionamiento**:
 - Direccionamiento **directo**.
 - La dirección es la propia llave (debe ser numérica y de rango igual al tamaño del fichero).
 - Direccionamiento **asociado** (por clave).
 - Cada llave tiene asociada una dirección en una tabla.
 - Al añadir nuevos registros las llaves se colocan al final de la tabla
 - La tabla está desordenada, lo cual puede hacer lento el proceso. Para hacerlo más rápido se puede tener la tabla ordenada o bien almacenada en memoria principal
 - Direccionamiento **calculado** (Hashing).
 - Se utilizan técnicas de Hashing.

Llave	Dir.
ABC	1028
ZHA	4532
ALP	3456
MAX	7231
etc	etc



Organización de ficheros

- Dos formas de resolver los **sinónimos**:
 1. **Búsqueda** de una posición libre.
 - Secuencialmente.
 - Aplicando otro método de direccionamiento.

– Ambos métodos son lentos
 2. Mediante **zona de desbordamientos**. Esta se puede gestionar:
 - Secuencialmente.
 - Encadenada con la zona principal.



Organización de ficheros

- **Operaciones :**

- Creación.

- Se debe reservar espacio en disco.

- Consulta.

- Se realiza por llave.
- 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 llave



Primitivas de acceso

- Definición de *tipos* de ficheros y registros.
 - En pseudolenguaje sólo vamos a definir primitivas para ficheros **binarios sin formato, secuenciales y directos**.
 - Cuando el formato de almacenamiento de los registros del fichero coincide con el utilizado en memoria principal se dice que el **binario**.
 - No todos los lenguajes permiten definir estructuras de registros (**Ficheros sin formato**).
El formato de las componentes del fichero se establecerá en las correspondientes primitivas de acceso.
 - Sintaxis:

TipoFichero = FICHERO



Primitivas de acceso

- Descriptor de fichero.

- El descriptor es una variable de un tipo especial (*Tipo Fichero*) definida en el programa, y desde la que se puede acceder a un fichero.
- Contiene el buffer y el cursor del fichero.

Tipos

Complejo = **REGISTRO** real, imag : **R FINREGISTRO**

FicheroComplejos = **FICHERO**

Variables

Mifichero : FicheroComplejos

FicheroEnteros : **FICHERO**



Primitivas de acceso

- **Apertura**: Asocia un fichero físico existente con un `FILE` y sitúa el cursor al principio del mismo.
ABRIR (c: ARRAY DE \$): FICHERO
- **Apertura para añadir**: Asocia un fichero existente con un `FILE` y sitúa el cursor al final del mismo.
AÑADIR (c: ARRAY DE \$): FICHERO
- **Creación**: Crea y abre un fichero. Si el fichero existe se destruye su contenido.
CREAR (c: ARRAY DE \$): FICHERO
- **Cierre**: Libera los recursos asociados al `FILE`.
CERRAR (f: FICHERO)



Primitivas de acceso (Acceso secuencial)

- Primitivas de *acceso secuencial*

- Lectura.

- Recoge del fichero tantos bytes como sea el tamaño de “T”.
 - Transfiere esos bytes a la variable que se le pasa como parámetro interpretándolos como del tipo de la variable.
 - Mueve el cursor al siguiente registro (si existe).
 - Si se ha alcanzado el final del fichero la función EOF

LEERBIN (F: FICHERO; VAR v: T)

Donde “T” es cualquier tipo y se lee directamente del fichero en la



Primitivas de acceso (Acceso secuencial)

– Escritura.

- Transfiere al fichero la información contenida en la variable que se pasa como parámetro en el formato de la misma.
- Escribe en el fichero tantos bytes como sea el tamaño de “T”.
- Mueve el cursor al siguiente hueco libre a escribir.

ESCRIBIRBIN (F: FICHERO; v: T)

Donde “T” es cualquier tipo y se escribe la representación interna de v.

– Fin de fichero (EOF: End Of File).

- Función que indica si la última operación realizada sobre un fichero ha alcanzado el final del mismo.

EOF (f: FICHERO): B



Primitivas de acceso (Acceso secuencial)

- Ejemplo de acceso secuencial.
 - Copiar un fichero de números enteros.

Algoritmo Copiar

Variables

```
fich1, fich2 : FICHERO  
dato : Z
```

Inicio

```
fich1 := ABRIR ("FICHERO1.DAT")  
fich2 := CREAR ("FICHERO2.DAT")  
MIENTRAS NOT EOF (fich1) HACER  
    LEERBIN (fich1, dato)  
    ESCRIBIRBIN (fich2, dato)  
FINMIENTRAS  
CERRAR (fich1)  
CERRAR (fich2)
```

Fin



Primitivas de acceso (Acceso directo)

- Primitivas de *acceso directo*.

- Buscar.

- Sitúa el cursor del fichero en el registro que se indique mediante su dirección relativa (número del byte donde comienza el
 - La próxima operación de lectura o escritura se realizará sobre la posición de dicho registro actual.
 - Si la posición a localizar no existe, la función EOF devolverá

BUSCAR (f: FICHERO; pos: N)

- Lectura, Escritura y Fin de Fichero.

- Igual que para el acceso secuencial.



Primitivas de acceso (Acceso directo)

– Posición.

- Devuelve la posición en bytes (dirección relativa) donde está el cursor del fichero.

POSICION (f: FICHERO): N

– Eliminar.

- Borra del registro actual los bytes indicados.
- Mueve el cursor del fichero al registro siguiente al borrado, si

ELIMINAR (f: FICHERO; numbytes: N)

– Longitud.

- Devuelve el número de bytes del fichero (N).

LONGITUD (f: FICHERO): N



Primitivas de acceso (Acceso directo)

- Ejemplo de acceso directo.
 - Búsqueda binaria en un fichero directo ordenado de

Algoritmo Búsqueda

Variables

```
fich1 : FICHERO
buscado, dato : Z
izqda, dcha, medio : N
Encontrado : B
```

Inicio

```
Escribir ("Introducir el valor a buscar: ")
```

```
ABRIR ("FICHERO1.DAT")
```

```
izqda := 0
```

```
dcha := LONGITUD (fich1) DIV Tamaño (Z)
```

```
Encontrado := FALSE
```



Primitivas de acceso (Acceso directo)

```
MIENTRAS NOT Encontrado AND izqda <= dcha HACER  
    medio := (izqda + dcha) DIV 2  
    BUSCAR (fich1, medio * Tamaño (Z))  
    LEERBIN (fich1, dato)  
    SI dato = buscado ENTONCES  
        Encontrado := TRUE  
    EN OTRO CASO  
        SI dato > buscado ENTONCES  
            dcha := medio - 1  
        EN OTRO CASO  
            izqda := medio + 1  
        FINSI  
    FINSI  
FINMIENTRAS  
CERRAR (fich1)  
SI Encontrado ENTONCES  
    Escribir ("Se encuentra en la posición", medio)  
EN OTRO CASO  
    Escribir ("El valor no se encuentra")  
FINSI
```

Fin



Primitivas de acceso

- Manipulación de ficheros.
 - **Renombrar.** Cambia el nombre de un fichero.
RENOMBRAR (nombre antiguo, nombre nuevo)
 - **Borrar.** Elimina físicamente el fichero cuyo nombre se indica.
BORRAR (nombre fichero)
 - Operaciones sobre **directorios**: Crear, Borrar.
CREARDIR (nombre directorio)
BORRARDIR (nombre directorio)



Primitivas de acceso (Ficheros de Texto)

- *Ficheros de texto.*

- **Definición.** Fichero secuencial compuesto por una secuencia de caracteres subdivida en registros de longitud variable llamados “**líneas**”.
- Cada línea almacena una cadena de caracteres que representa un dato concreto.
- Los registros o líneas se separan mediante un delimitador **EOL** (End Of Line).
- Sintaxis:

`TipoFicheroTexto = FICHERO`



Primitivas de acceso (Ficheros de Texto)

– Operaciones.

En un fichero de texto, los datos siempre se almacenan como secuencias de caracteres.

- Abrir un fichero de texto para leer o escribir.

ABRIR (c: ARRAY DE \$): FICHERO

- Abrir un fichero de texto para añadir al final.

AÑADIR (c: ARRAY DE \$): FICHERO

- Cerrar un fichero de texto.

CERRAR (f: FICHERO)

- Detectar fin de fichero.

EOF (f: FICHERO): B



Primitivas de acceso (Ficheros de Texto)

- Leer un dato.

LEER (F: FICHERO; VAR v: T)

Donde “T” es un tipo simple predefinido o bien un

- Detectar delimitador de fin de línea.

EOL (f: FICHERO): B

- Escribir un dato.

ESCRIBIR (F: FICHERO; v: T)

Donde “T” es un tipo simple predefinido o bien un

Los datos siempre se almacenan como secuencias de caracteres.

- Almacenar delimitador de fin de línea.

ESCRIBIR (f: FICHERO, EOL)



Primitivas de acceso (Ficheros de Texto)

- Ejemplo. Procedimiento para leer cadenas.

Algoritmo LeeCadena (VAR fich: FICHERO; VAR cad: TipoCad)

Variables

i : N
car : \$

Inicio

```
i := 1
MIENTRAS NOT EOL (fich) AND NOT EOF (fich) HACER
    LEER (fich, car)
    SI i <= MAXCAD ENTONCES
        cad [i] := car
        i := i + 1
    FINSI
FINMIENTRAS
(* Lee y salta el delimitador en el fichero *)
SI EOL (fich) ENTONCES LEER (fich, car) FINSI
(* Pone el carácter de fin de cadena *)
SI i <= MAXCAD ENTONCES cad [i] := CHR (0) FINSI
```

Fin



Ordenación externa

- Tarea fundamental en procesamiento de datos.
- No se puede acceder a todos los datos, es necesario realizar la clasificación por partes.
- Se generan secuencias ordenadas de registros al mezclar repetidamente secuencias más pequeñas. Cada mezcla produce una secuencia ordenada más larga que su entrada.
- El proceso se realiza siempre sobre secuencias (ficheros secuenciales).



Ordenación externa

Algoritmo MezclaDosFicheros

Variables

```
ent1,ent2: FICHERO
dato1, dato2 : TReg
```

Inicio

```
ent1 := ABRIR ("FENT1.DAT")
ent2 := ABRIR ("FENT2.DAT")
sal := CREAM ("FSAL.DAT")
LEERBIN (ent1, dato1)
LEERBIN (ent2, dato2)
MIENTRAS NOT EOF (ent1) AND NOT
EOF (ent2) HACER
  SI dato1 < dato2 ENTONCES
    ESCRIBIRBIN (sal, dato1)
    LEERBIN (ent1, dato1)
  EN OTRO CASO
    ESCRIBIRBIN (sal, dato2)
    LEERBIN (ent2, dato2)
  FINSI
FINMIENTRAS
```

– **Proceso de mezcla de ficheros:**
Es una tarea básica a realizar
en todos los algoritmos de
ordenación externa.

```
MIENTRAS NOT EOF (ent1) HACER
  ESCRIBIRBIN (sal, dato1)
  LEERBIN (ent1, dato1)
FINMIENTRAS
MIENTRAS NOT EOF (ent2) HACER
  ESCRIBIRBIN (sal, dato2)
  LEERBIN (ent2, dato2)
FINMIENTRAS
```

Fin



Bibliografía

- Estructuras de datos. Lewis & Smith. Paraninfo.
- Algoritmos y estructura de datos. .
- Curso de programación. Castro, .
- Introducción moderna a la ciencia de la & .
- . .