

Modificación de archivos en C

INFO II – Profesor: Ing. Federico Weber

Modificación de Archivos

- Modificar el contenido de un archivo existente puede ser complejo dependiendo de si el cambio altera el tamaño del archivo o no.
- Veremos dos enfoques:
 - **modificación directa en el mismo archivo (en modo binario)**, útil cuando los registros son de tamaño fijo y sobrescribimos datos sin cambiar longitudes, y
 - **modificación usando un archivo temporal**, que es más general y seguro para inserciones, eliminaciones o cambios de tamaño variable.

Modificación directa en el archivo (modo binario, registros fijos)

- Si tenemos un archivo binario de registros de longitud fija (por ejemplo, estructuras de tamaño constante), podemos actualizar un registro específico *in-place*.
- Esto requiere abrir el archivo en modo lectura/escritura ("r+b") y usar fseek para posicionarnos justo en el lugar del registro a modificar.
- Se sobrescribe entonces con los nuevos datos.
- **Importante:** Esta técnica solo funciona correctamente si el nuevo contenido ocupa exactamente el mismo espacio que el antiguo; de lo contrario, truncaría o desalinearía el resto del archivo.

Ejemplo01.c:

- Supongamos un archivo **productos.dat** con registros de esta estructura:

```
typedef struct {  
    int codigo;  
    char descripcion[41];  
    float precio;  
} Producto;
```

```
Producto productos[] = {  
    {1001, "higiene", 120.50},  
    {1002, "higiene", 85.30},  
    {1003, "higiene", 95.75},  
    {1004, "higiene", 150.00},  
    {2001, "alimentos", 250.00},  
    {2002, "alimentos", 310.45},  
    {2003, "alimentos", 199.99},  
    {3001, "bebida", 180.00},  
    {3002, "bebida", 220.50}  
};
```

Ejemplo01.c:

- Buscar el elemento cuyo código es 1003 y duplicar el precio:

```
typedef struct {  
    int codigo;  
    char descripcion[41];  
    float precio;  
} Producto;
```

```
Producto productos[] = {  
    {1001, "higiene", 120.50},  
    {1002, "higiene", 85.30},  
    {1003, "higiene", 95.75},  
    {1004, "higiene", 150.00},  
    {2001, "alimentos", 250.00},  
    {2002, "alimentos", 310.45},  
    {2003, "alimentos", 199.99},  
    {3001, "bebida", 180.00},  
    {3002, "bebida", 220.50}  
};
```

Ejemplo01.c: Buscar el elemento cuyo código es 1003 y duplicar el precio:

- Debo utilizar el modo de apertura **r+b**

Abro el archivo y automáticamente la ventana se coloca en la primer posición.

1. Leo

1001
1002
1003
1004
2001
2002
2003
3001
3002

1001

Ejemplo01.c: Buscar el elemento cuyo código es 1003 y duplicar el precio:

- Debo utilizar el modo de apertura **r+b**

1001
1002
1003
1004
2001
2002
2003
3001
3002

1001

Abro el archivo y automáticamente la ventana se coloca en la primer posición.

1. Leo
2. La ventana se mueve automáticamente
3. ¿Es 1001 el numero que buscamos?
 - No
4. Leo

Ejemplo01.c: Buscar el elemento cuyo código es 1003 y duplicar el precio:

- Debo utilizar el modo de apertura **r+b**

1001
1002
1003
1004
2001
2002
2003
3001
3002

1002

Abro el archivo y automáticamente la ventana se coloca en la primer posición.

1. Leo 1001.
2. La ventana se mueve automáticamente
3. ¿Es 1001 el numero que buscamos?
 - No
4. Leo.
5. La ventana se mueve automáticamente
6. ¿Es 1002 el numero que buscamos?
 - No
7. Leo.

Ejemplo01.c: Buscar el elemento cuyo código es 1003 y duplicar el precio:

- Debo utilizar el modo de apertura **r+b**

1001	
1002	
1003	
1004	
2001	
2002	
2003	
3001	
3002	

1003

Abro el archivo y automáticamente la ventana se coloca en la primer posición.

1. Leo 1001.
2. La ventana se mueve automáticamente
3. ¿Es 1001 el numero que buscamos?
 - No
4. Leo.
5. La ventana se mueve automáticamente
6. ¿Es 1002 el numero que buscamos?
 - No
7. Leo.
8. La ventana se mueve automáticamente
9. ¿Es 1003 el numero que buscamos?
 - SI
 - ¿La ventana esta bien posicionada para modificar?
 - NO

Ejemplo01.c: Buscar el elemento cuyo código es 1003 y duplicar el precio:

- Debo utilizar el modo de apertura **r+b**

1. Duplico el precio
2. Ubico la ventana

1001	
1002	
1003	
1004	
2001	
2002	
2003	
3001	
3002	

1003

Ejemplo01.c: Buscar el elemento cuyo código es 1003 y duplicar el precio:

- Debo utilizar el modo de apertura **r+b**

1. Duplico el precio
2. Ubico la ventana
3. Escribo el nuevo dato

1001	
1002	
1003	
1004	
2001	1003
2002	
2003	
3001	
3002	

A close-up, slightly blurred photograph of a person's hands typing on a laptop keyboard. The hands are positioned over the keyboard, with fingers pressing down on keys. The laptop screen is visible in the background, displaying a code editor with syntax-highlighted text. The text on the screen includes variables like 'options', 'size', 'line', 'mainColor', and 'planFeaturesSize'. The overall lighting is dim, and the focus is on the hands and the keyboard.

ejemplo01.c

Precauciones

- El archivo debe abrirse en modo que permita escritura sin truncar (r+ o r+b, *no* w porque borra el contenido).
- Si los datos a modificar pudieran cambiar de longitud (por ejemplo, una cadena más larga que la original), *no debe hacerse in-place* a menos que se reserve espacio máximo de antemano (como en este ejemplo, la descripción tiene un tamaño fijo de 40 caracteres).
- Si se reduce la longitud de un campo y se quiere aprovechar el espacio, normalmente no se “achica” el archivo in-place; queda fragmentación interna o se marca el registro como borrado.
- Usar fflush si se requiere asegurar que los datos se escribieron inmediatamente (por ejemplo, antes de leer de nuevo en ciertas situaciones). Al cerrar con fclose de todos modos se vacían los buffers.
- Mientras se modifica un archivo, es buena práctica bloquear el archivo (file locking) si hay posibilidad de que otro proceso/parte del programa acceda concurrentemente, para evitar lecturas inconsistentes. En C estándar no hay locks de alto nivel, pero en POSIX se podría usar **fcntl()** o similares. En Windows tenemos **LockFile()** (y su compañera **UnlockFile()**).

Modificación usando un archivo temporal

- Para casos más generales, especialmente en archivos de texto o cuando la modificación cambia el tamaño del contenido (p. ej., agregar o quitar registros), la técnica común es *no modificar directamente el archivo original*, sino crear un **archivo temporal** donde se va volcando la información modificada, y luego reemplazar el original por el temporal.
- Aunque implica más I/O (reescribir posiblemente todo el archivo), es mucho más sencillo de implementar de manera correcta y reduce riesgos de corrupción de datos si algo falla en medio (ya que siempre conservamos el original hasta tener la nueva versión completa).

Estrategia típica:

1. Abrir el archivo original en modo lectura (solo lectura o lectura/escritura).
2. Abrir un archivo temporario en modo escritura (por ejemplo, usando **tmpfile()** o creando un nombre como "temp.dat").
3. Recorrer el archivo original leyendo registro por registro (o línea por línea).
4. Escribir en el temporal ya sea el registro modificado o el original sin cambios, según corresponda. Si se quiere **eliminar** un registro, simplemente se omite escribirlo en el temporal. Si se quiere **insertar** uno nuevo, se escribe en el momento apropiado en el temporal.
5. Cerrar ambos archivos.
6. Reemplazar el archivo original con el temporario: esto puede hacerse usando `remove()` para borrar el original y `rename()` para renombrar el temporal con el nombre original. (En Windows, es posible renombrar sobre el existente directamente con ciertas funciones, pero en POSIX la secuencia `remove+rename` es común).

Ejemplo02.c

Aumentamos 10% el precio de todos los artículos. La idea es crear un archivo temporal para hacer la modificación.

A close-up, slightly blurred photograph of a person's hands typing on a laptop keyboard. The hands are positioned over the keyboard, with fingers pressing down on keys. The laptop screen is visible in the background, displaying a code editor with syntax-highlighted text. The text on the screen includes variables like 'options', 'size', and 'line', along with some numerical values and a '1825' in orange. The overall lighting is dim, and the focus is on the hands and the keyboard.

ejemplo02.c

Ejemplo03:
Modificamos todos
los productos con la
descripción
“alimentos”





FIN