

# Arquitectura de sistemas

**Abelardo Pardo**

University of Sydney  
School of Electrical and Information Engineering  
NSW, 2006, Australia  
*Autor principal del curso de 2009 a 2012*

**Iria Estévez Ayres**

**Damaris Fuentes Lorenzo**

**Pablo Basanta Val**

**Pedro J. Muñoz Merino**

**Hugo A. Parada**

**Derick Leony**

Universidad Carlos III de Madrid  
Departamento de Ingeniería Telemática  
Avenida Universidad 30, E28911 Leganés (Madrid), España

---

© Universidad Carlos III de Madrid | Licencia Creative Commons



## **Tabla de contenidos**

### 9.1. Actividades

- [9.1.1. Manejando ficheros en C \(5 puntos\)](#)
- [9.1.2. Recuperando cadenas de un fichero \(con fread\)](#)
- [9.1.3. Guardando cadenas en un fichero \(con fwrite\)](#)
- [9.1.4. Leer un texto de fichero, contar sus caracteres y escribirlos en otro fichero](#)
- [9.1.5. Leer un texto de fichero, contar sus líneas y escribir ese número en otro fichero](#)
- [9.1.6. Leer una línea por pantalla y escribir en un fichero](#)
- [9.1.7. Leer un número desconocido de palabras de un fichero de texto](#)
- [9.1.8. Leer un número desconocido de enteros de un fichero](#)
- [9.1.9. Leer una línea por pantalla y escribir en un fichero](#)
- [9.1.10. Leer una línea por pantalla y escribir en un fichero](#)
- [9.1.11. Manejando ficheros en C](#)
- [9.1.12. Abrir y cerrar un fichero](#)
- [9.1.13. Leer un número conocido de enteros de un fichero](#)
- [9.1.14. Obtención del tamaño de un fichero](#)

---

## **9.1. Actividades**

---

### **9.1.1. Manejando ficheros en C (5 puntos)**

---



#### **Recursos**

- Documento [Lectura y escritura de ficheros](#)



#### **Plan de trabajo**

Realizarás un programa dado por tu profesor en clase, relacionado con el manejo de ficheros y con lo visto hasta ahora. Podrás usar el material de clase.



#### **Evaluación**

Cuando termines, enséñaselo a tu tutor para que pueda evaluarlo.

---

### **9.1.2. Recuperando cadenas de un fichero (con fread)**

---



#### **Recursos**

- Sección 3 de [Lectura y escritura de ficheros](#)



#### **Plan de trabajo**

En este ejercicio se hará la inversa del ejercicio `fwrite_getline.c`. Dado el fichero en el que has guardado las cadenas, precedidas cada una por su tamaño, deberás recuperarlas y mostrarlas por

pantalla, tanto dichas cadenas como su tamaño. Para ello deberás usar las funciones `fread` y `malloc`. El fichero deberás llamarlo `fread_variable_strings.c`

Sube el programa a la carpeta `fread` de tu espacio de trabajo en Subversion.

### 9.1.3. Guardando cadenas en un fichero (con `fwrite`)

---



#### Recursos

- Sección 3 de [Lectura y escritura de ficheros](#)



#### Plan de trabajo

1. Escribe un programa con nombre `fwrite_getline.c` (que encontrarás en tu carpeta `fwrite`) que realice las siguientes operaciones:

1. Compruebe que el programa se ejecuta con sólo un argumento. En caso de que no sea así, imprime un mensaje y termina.
2. Abre el fichero cuyo nombre es la cadena pasada como único argumento. Si la operación falla, notificar mediante un mensaje y terminar la ejecución.
3. Va pidiéndole al usuario por pantalla que introduzca introduzca una cadena o CTRL+D para terminar el programa (usando la función `getline`). `getline`).

Calcula el tamaño de la cadena (con `strlen`) y guarda en el fichero con la función `fwrite` tanto el tamaño como la cadena. Cuando el usuario quiere terminar el programa debe pulsar a la vez CTRL+D.

El programa comprobará que ha introducido CTRL+D comprobando el valor devuelto por `getline` (lee sobre `getline` en su página de manual).

Se recuerda que `getline` realiza internamente un `malloc` (si no lo hace explícitamente el usuario), por lo que deberá realizarse tantas llamadas a `free` como llamadas a `malloc` se hayan hecho.

4. Cierra el fichero y si la operación produce un error, notifícalo con un mensaje por pantalla.
2. Sube el programa a la carpeta `fwrite` de tu espacio de trabajo en Subversion.

### 9.1.4. Leer un texto de fichero, contar sus caracteres y escribirlos en otro fichero

---



#### Recursos

- Fichero `Entrada_caracteres.txt` del directorio `Files_exam`
- Sección 3 de [Lectura y escritura de ficheros](#)



#### Plan de trabajo

Escribe un programa que lea el texto contenido en el archivo `Entrada_caracteres.txt` , contabilice el número de caracteres del mismo y escriba ese número tanto por pantalla como en un

nuevo fichero llamado `Resultado.bin`.

Sube el programa a la carpeta `Files_exam` de tu espacio de trabajo en Subversion.

### 9.1.5. Leer un texto de fichero, contar sus líneas y escribir ese número en otro fichero

---



#### Recursos

- Fichero `entrada_lineas.txt` del directorio `Files_exam`
- Sección 3 de [Lectura y escritura de ficheros](#)



#### Plan de trabajo

Escribe un programa que lea el texto contenido en el archivo `Entrada_lineas.txt`, contabilice el número de líneas del mismo y escriba ese número tanto por pantalla como en un nuevo fichero llamado `Resultado.bin`.

Sube el programa a la carpeta `Files_exam` de tu espacio de trabajo en Subversion.

### 9.1.6. Leer una línea por pantalla y escribir en un fichero

---



#### Recursos

- Sección 3 de [Lectura y escritura de ficheros](#)



#### Plan de trabajo

Escribe un programa que lea una línea del usuario por teclado (con `getline`), cuente la frecuencia con la que aparece cada letra de esa línea (obvia la letra 'ñ' y considera el abecedario de 26 letras) y escriba esas frecuencias en un fichero (para probar que las has escrito correctamente, prueba a leerlas del fichero).

Sube el programa a la carpeta `Files_exam` de tu espacio de trabajo en Subversion.

### 9.1.7. Leer un número desconocido de palabras de un fichero de texto

---



#### Recursos

- Sección 3 de [Lectura y escritura de ficheros](#)



#### Plan de trabajo

Escribe un programa que lea un fichero de texto (hasta fin de archivo) e indique el porcentaje de palabras que tienen menos de 5 caracteres y el porcentaje de palabras con 5 o más caracteres. Puedes crearte tú mismo el archivo escribiendo en él una línea.

### 9.1.8. Leer un número desconocido de enteros de un fichero

---



## Recursos

- Para esta actividad puedes partir del código que has escrito para el ejercicio `fread_fixed_integers.c` que leía enteros desde un fichero. Copia ese código en la carpeta `fread_variable` de tu carpeta compartida.
- Sección 3 de [Lectura y escritura de ficheros](#)



## Plan de trabajo

Los ficheros en Linux son una secuencia de bytes. Si en un fichero se almacenan datos de un tamaño arbitrario, nos encontramos con el problema de saber la cantidad de datos a leer para pasárselo como parámetro a la función `fread`. Hay varias formas de solventar esto, pero una de ellas consiste en anteponer a los datos un entero con su tamaño. De esta forma, el primer dato que se lee es el entero que indica cuántos datos están almacenados a continuación. La siguiente figura muestra esta técnica para un fichero con 5 enteros.

```
TAMANHO|ENTERO_0|ENTERO_1|ENTERO_2
```

Modifica el programa de `fread_fixed_integers.c`, llamándole `fread_variable_integers.c` para que lea los enteros de un fichero binario con esta estructura (números en binario almacenados en posiciones consecutivas precedidos por un entero con el tamaño):

Tu código deberá leer el tamaño y después pedir memoria con la función `malloc` para el array de enteros. No olvides utilizar la llamada a la función `free` para liberar la memoria antes de que termine el programa.

Prueba el programa con los siguientes ficheros `fich_10_enteros` y `fich_enteros` que encontrarás en la sub-carpeta `fread_variable` de tu carpeta compartida.

Sube el programa a la carpeta `fread_variable` de tu espacio de trabajo en Subversion.

### **9.1.9. Leer una línea por pantalla y escribir en un fichero**

---



## Recursos

- Sección 3 de [Lectura y escritura de ficheros](#)



## Plan de trabajo

Escribe un programa que lea una línea del usuario por teclado (con `getline`) y que la almacene en un fichero carácter a carácter, pasando todos esos caracteres previamente a mayúscula (para probar que los has escrito correctamente, lee el fichero).

Sube el programa a la carpeta `Files_exam` de tu espacio de trabajo en Subversion.

### **9.1.10. Leer una línea por pantalla y escribir en un fichero**

---



## Recursos

- Sección 3 de [Lectura y escritura de ficheros](#)



## Plan de trabajo

Escribe un programa que lea una línea del usuario por teclado (con `getline`), indique qué letras no han aparecido en el texto y escriba esas letras en un fichero (para probar que las has escrito correctamente, prueba a leerlas del fichero).

Sube el programa a la carpeta `Files_exam` de tu espacio de trabajo en Subversion.

---

### 9.1.11. Manejando ficheros en C



## Recursos

- Documento [Lectura y escritura de ficheros](#)



## Plan de trabajo

Lee detenidamente el documento de manejo de ficheros en C que se te ofrece como recurso.

Asegúrate de que eres capaz de:

- Definir cual es el modelo de entrada/salida en C.
- Definir qué es el "indicador de posición del fichero" y qué función debes utilizar para moverlo a un punto concreto del fichero.
- Conocer las funciones para abrir y cerrar ficheros.
- Conocer las funciones para leer datos de un fichero y escribirlos en ellos, con las funciones `fread` y `fwrite`.
- Indicar qué función devuelve la posición del indicador de posición de un fichero.

---

### 9.1.12. Abrir y cerrar un fichero



## Recursos

- Sección 1 y 2 de [Lectura y escritura de ficheros](#)
- Copia de la plantilla `CFile.templ` del directorio `Plantillas` en tu carpeta compartida a la subcarpeta `fopen_fclose`.



## Plan de trabajo

Escribe un programa con nombre `main.c` que realice las siguientes operaciones (parte de la plantilla en el fichero `CFile.templ` que encontrarás en la subcarpeta `fopen_fclose` de la carpeta compartida):

1. Cualquier programa en C se puede ejecutar desde la línea de comandos añadiendo palabras separadas por espacios. La función `main` recibe como parámetros un entero (típicamente se le pone el nombre `argc`) con el número de argumentos que se han escrito al invocar el programa (el nombre del programa se cuenta) y como segundo parámetro un array con los punteros a las

... nombre del programa de ejemplo, como se sigue puntualmente en el código, con los parámetros de las cadenas que se han escrito en la línea de comandos (por tanto de tipo `char **`).

Escribe código en el `main` para que compruebe que el programa se ejecuta con sólo un argumento. En caso de que no sea así, imprime un mensaje y termina.

2. Abre el fichero cuyo nombre es la cadena pasada como único argumento. Si la operación falla, notificar mediante un mensaje y terminar la ejecución.
3. Cierra el fichero y si la operación produce un error, notifícalo con un mensaje por pantalla.

Prueba el programa con diferentes nombres de ficheros, así como con nombres que no corresponden a fichero alguno.

Sube el programa a la carpeta `fopen_fclose` de tu espacio de trabajo en Subversion.

### 9.1.13. Leer un número conocido de enteros de un fichero

---



#### Recursos

- Sección 3, 4 y 5 de [Lectura y escritura de ficheros](#)
- Fichero `fread_fixed_integers.c` almacenado en la sub-carpeta `fread_fixed` de tu carpeta compartida.
- Ficheros de datos en formato binario en la sub-carpeta `fread_fixed` de tu carpeta compartida.



#### Plan de trabajo

1. Escribe un programa en el fichero `fread_fixed_integers.c` almacenado en el directorio `fread_fixed` de la carpeta compartida que realice las siguientes operaciones:
  1. Comprobar que el programa se ejecuta con sólo un argumento. En caso de que no sea así, imprime un mensaje y termina.
  2. Abrir el fichero cuyo nombre es la cadena pasada como único argumento. Si la operación falla, notificar mediante un mensaje y terminar la ejecución.
  3. Leer del fichero como mucho 10 enteros y los guarda en un array de tamaño 10. El programa debe funcionar con cualquier número de enteros de 0 a 10. Si la operación de lectura produce un error, debes notificarlo con un mensaje por pantalla. Si todo ha ido bien, muestra por pantalla los números leídos.
  4. Cerrar el fichero y si la operación produce un error, notifícalo con un mensaje por pantalla.
2. Prueba el programa con los ficheros `fich_0_enteros` (fichero vacío), `fich_07_enteros` (fichero con 7 enteros) y `fich_10_enteros` (fichero con 10 enteros) que encontrarás en tu carpeta de trabajo compartida.
3. Sube el programa a la carpeta `fread_fixed` de tu espacio de trabajo en Subversion.

### 9.1.14. Obtención del tamaño de un fichero

---



#### Recursos

- Sección 4 de [Lectura y escritura de ficheros](#). Más concretamente las funciones de biblioteca

fseek y ftell. Puedes utilizar el comando **man** desde un terminal de comandos para acceder a esta información.

- Fichero `main.c` de la actividad " " en la carpeta `fopen_fclose`.



## Plan de trabajo

Escribe una función con el siguiente prototipo:

```
static long get_file_size(FILE * file);
```

Esta función devuelve el tamaño en bytes del fichero que se pasa como parámetro y que debe estar previamente abierto. Añade en el `main` una línea que llama a esta función e imprime por pantalla el tamaño del fichero dado. Comprueba con ficheros almacenados en ese mismo directorio, que la cifra coincide con la que se obtiene con el comando `ls -l` (en la cuarta columna muestra el tamaño en bytes de un fichero).



## Evaluación

Una vez escrita la función y verificado su funcionamiento con un programa, subela a Subversion. Luego, muéstrasela al tutor.