

Programación I

Ficheros de texto

Iván Cantador

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Contenidos

1

- Ficheros
- Lectura de ficheros de texto
- Escritura de ficheros de texto
- Aspectos avanzados
 - Posicionamiento en un fichero
 - Lectura de *tokens* de cadenas de caracteres

Contenidos

2

- **Ficheros**
- Lectura de ficheros de texto
- Escritura de ficheros de texto
- Aspectos avanzados
 - Posicionamiento en un fichero
 - Lectura de *tokens* de cadenas de caracteres

Ficheros (I)

3

- Un **fichero** o **archivo** es un conjunto de bits que es tratado por el sistema operativo como única unidad lógica
 - Un fichero está identificado por su nombre (en general, con extensión) y por la ruta de la carpeta/directorio en el que se encuentra

<C:/Programacion1/Ejemplos/archivo.txt>
 - El tamaño de un fichero se expresa generalmente en Bytes
- Una **carpeta** o **directorio** es en realidad un fichero, cuya finalidad es la de “almacenar” otras carpetas y ficheros

- Existen **2 tipos de ficheros**:

- Ficheros de texto (plano)**: contienen sólo líneas de texto sin formato, sin otro tipo de datos (p.e. imágenes, gráficas, etc.)
 - Con posibles codificaciones: ASCII, UTF-8, EBCDIC ...
 - Normalmente asociados a la extensión .txt
- Ficheros binarios**: contienen datos almacenados en un formato arbitrario (generalmente asociado a una extensión concreta: .doc, .pdf, .jpg, .mp3) representando todo tipo de medios: texto, imagen, sonido, video

- Desde un programa informático la lectura y escritura de un fichero se realizan en **3 pasos**:
 - Apertura
 - Lectura/escritura
 - Cierre
- En programación, un fichero se suele representar mediante una estructura, denominada “**manejador**” (*handler* en inglés)
 - En C, se usa el tipo de datos **FILE**, definido en **<stdio.h>**
 - Entre otras cosas, contiene un puntero a la próxima posición de lectura/escritura del fichero

- Apertura

```
FILE *fopen(char *rutaFichero, char *modoApertura)
```

- Lectura/cierre

```
int fscanf(FILE *fichero, char *formato, ... <variables> ...)
```

```
char *fgets(char *cadena, int numeroBytes, FILE *fichero)
```

```
int fprintf(FILE *fichero, char *formato, ... <variables> ...)
```

- Cierre

```
void fclose(FILE *fichero)
```

- Apertura de un fichero

```
FILE *fopen(char *rutaFichero, char *modoApertura)
```

- Modos de apertura principales**

- "r"**: apertura del fichero en modo lectura
- "w"**: apertura del fichero en modo escritura - sobreescritura; si el fichero no existe, lo crea; si el fichero sí existe, escribe al principio del mismo y eliminando el texto existente
- "a"**: apertura del fichero en modo escritura - actualización; si el fichero no existe, lo crea; si el fichero sí existe, escribe al final del mismo sin eliminar el texto existente

- Ficheros
- **Lectura de ficheros de texto**
- Escritura de ficheros de texto
- Aspectos avanzados
 - Posicionamiento en un fichero
 - Lectura de *tokens* de cadenas de caracteres

• Lectura de un fichero de texto: **fscanf**

```
int fscanf(FILE *fichero, char *formato, ... <variables> ...)
```

- **Ejemplo:** lectura de un fichero con una línea que contiene un número entero, un número real, y una cadena de caracteres separados por espacios en blanco

```
include <stdio.h>
```

```
void main() {
    FILE *fichero = NULL;
    int retorno, n;
    double r;
    char s[256];

    fichero = fopen("C:/Programacion1/prueba.txt", "r");
    if ( fichero ) {
        retorno = fscanf(fichero, "%d %f %s\n", &n, &r, s);
        if ( retorno == 3 ) printf("Leidos: %d %f %s\n", n, r, s);
        fclose(fichero);
    }
}
```

• Lectura de un fichero de texto: **fscanf**

```
int fscanf(FILE *fichero, char *formato, ... <variables> ...)
```

- **Ejemplo:** lectura de varias líneas de un fichero (con el formato de línea del ejemplo anterior)

```
include <stdio.h>
```

```
void main() {
    FILE *fichero = NULL;
    int retorno, n;
    double r;
    char s[256];

    fichero = fopen("C:/Programacion1/prueba.txt", "r");
    if ( fichero ) {
        // Mientras no sea fin de fichero, leemos linea
        while ( !feof(fichero) ) {
            retorno = fscanf(fichero, "%d %f %s\n", &n, &r, s);
            if ( retorno == 3 ) printf("Leidos: %d %f %s\n", n, r, s);
        }
        fclose(fichero);
    }
}
```

- Ficheros
- Lectura de ficheros de texto
- **Escritura de ficheros de texto**
- Aspectos avanzados
 - Posicionamiento en un fichero
 - Lectura de *tokens* de cadenas de caracteres

- Lectura de un fichero de texto: **fprintf**

```
int fprintf(FILE *fichero, char *formato, ... <variables> ...)
```

- **Ejemplo:** creación de un fichero con una línea que contiene un número entero, un número real, y una cadena de caracteres separados por espacios en blanco

```
include <stdio.h>
```

```
void main() {  
    FILE *fichero = NULL;  
    int n = 10;  
    double r = 8.5;  
    char s[256] = "abcde";  
  
    // Abrimos el fichero en modo escritura - sobreescritura  
    fichero = fopen("C:/Programacion1/prueba.txt", "w");  
  
    if ( fichero ) {  
        fprintf(fichero, "%d %f %s\n", n, r, s);  
        fclose(fichero);  
    }  
}
```

- Lectura de un fichero de texto: **fprintf**

```
int fprintf(FILE *fichero, char *formato, ... <variables> ...)
```

- **Ejemplo:** actualización de un fichero añadiendo una línea que contiene un número entero, un número real, y una cadena de caracteres separados por espacios en blanco

```
include <stdio.h>
```

```
void main() {  
    FILE *fichero = NULL;  
    int n = 10;  
    double r = 8.5;  
    char s[256] = "abcde";  
  
    // Abrimos el fichero en modo escritura - actualizacion  
    fichero = fopen("C:/Programacion1/prueba.txt", "a");  
  
    if ( fichero ) {  
        fprintf(fichero, "%d %f %s\n", n, r, s);  
        fclose(fichero);  
    }  
}
```

Contenidos

14

- Ficheros
- Lectura de ficheros de texto
- Escritura de ficheros de texto
- Aspectos avanzados
 - Posicionamiento en un fichero
 - Lectura de *tokens* de cadenas de caracteres

Posicionamiento en un fichero

15

- Posicionamiento en un fichero: **fseek**

```
fseek(<fichero>, <offset>, <origen>)
```

```
fseek(f, 0L, SEEK_SET); // Al comienzo del fichero  
fseek(f, 0L, SEEK_CUR); // En la posicion actual  
fseek(f, 0L, SEEK_END); // Al final del fichero
```

```
fseek(f, 256L, SEEK_SET); // En el Byte 256
```

- Funciones relacionadas

- **ftell**: devuelve la posición actual de lectura/escritura
- **rewind**: equivale a `fseek(f, 0L, SEEK_SET)`

- Ficheros
- Lectura de ficheros de texto
- Escritura de ficheros de texto
- Aspectos avanzados
 - Posicionamiento en un fichero
 - Lectura de *tokens* de cadenas de caracteres

- La función **strtok** (de <string.h>)
 - “Tokeniza” (trocea) una cadena de caracteres atendiendo a un conjunto de caracteres *delimitadores*

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

void main() {
    char *cadena = "primero segundo, tercero\tcuarto";
    char *delimitadores = " ,\t";
    char *token = NULL;

    printf("Tokens de %s:\n", cadena);

    token = strtok(cadena, delimitadores);
    while ( token ) {
        printf("%s\n", token);
        token = strtok(NULL, delimitadores);
    }
}
```

- La función **strtok** (de <string.h>)

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

void main() {
    FILE *fichero = NULL;
    char linea[256];
    char *delimitadores = " ,\t";
    char *token = NULL;

    fichero = fopen("C:/miFichero.txt", "r");
    if ( fichero ) {
        while ( !feof(fichero) ) {
            fgets(linea, 256, fichero); // Leemos la proxima linea
            // Leemos tokens de la linea
            token = strtok(linea, delimitadores);
            while ( token ) {
                printf("%s\n", token);
                token = strtok(NULL, delimitadores);
            }
        }
        fclose(fichero);
    }
}
```