

fopen
fwrite
fread
fputs
fgets
fprintf
fscanf

Curso de programación en C moderno (II Edición)

Neira Ayuso, Pablo Falgueras García, Carlos

Tema 10

Entrada/Salida

Portada

Streams

Funciones

fopen
fwrite
fread
fputs
fgets
fprintf
fscanf

Funciones peligrosas

1 ¿Qué es un stream?

2 Funciones básicas

- fopen
- fwrite
- fread
- fputs
- fgets
- fprintf
- fscanf

3 Funciones peligrosas

¿Qué es un stream?

Portada

Streams

Funciones

fopen
fwrite
fread
fputs
fgets
fprintf
fscanf

Funciones peligrosas

Flujo de bytes de longitud indeterminada, al que se accede de forma **secuencial**.

En un *stream*:

- Al **leer** uno o más bytes, en la próxima lectura obtendremos los siguientes.
- Al **escribir** uno o más bytes, en la próxima escritura los añadiremos a continuación.

Streams estándar:

- **stdout**: *Salida estándar* (normalmente por consola)
- **stdin**: *Entrada estándar* (normalmente por teclado)
- **stderr**: *Salida estándar de errores* (normalmente por consola)

Funciones básicas

Portada

Streams

Funciones

fopen
fwrite
fread
fputs
fgets
fprintf
fscanf

Funciones peligrosas

- **Apertura y cierre de ficheros:**
 - `FILE *f = fopen("file.txt", "r");`
 - `fclose(f);`
- **Lectura/Escritura en crudo:**
 - `size_t read = fread(buf, sizeof(char), bufsize, f);`
 - `size_t written = fwrite(buf, sizeof(char), bufsize, f);`
- **Lectura/Escritura sin formato:**
 - `fputs("Hola Mundo", f);`
 - `fgets(buf, bufsize, f);`
- **Lectura/Escritura con formato:**
 - `fprintf(f, "Hola", name);`
 - `fscanf(f, "%10s %d", str, &i);`

Para más información ver:

<http://es.cppreference.com/w/c/io>

fopen

Portada

Streams

Funciones

fopen
fwrite
fread
fputs
fgets
fprintf
fscanf

Funciones peligrosas

```
FILE *fopen(const char *fname, const char *mode);
int fclose(FILE *stream);
```

Modos:

modo	significado	si ya existe	si no existe
"r"	Abre para leer	lee desde el principio	error
"w"	Crea para escribir	descarta el contenido	lo crea
"a"	Crea para añadir	escribe al final	lo crea
"r+"	Abre para leer/escribir	lee/escribe desde el principio	error
"w+"	Crea para leer/escribir	descarta el contenido	lo crea
"a+"	Crea para leer/añadir	lee*/escribe desde el final	lo crea

fwrite

Portada

size_t fwrite(const void *buf, size_t size, size_t count, FILE *strm);

Streams

Funciones

fopen
fwrite
fread
fputs
fgets
fprintf
fscanf

Funciones peligrosas

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 #define FNAME "file"
6
7 int main() {
8     FILE *f;
9     char buf[] = "Hola Mundo";
10
11     f = fopen(FNAME, "w+");
12     if (!f) {
13         perror("Error al abrir");
14         exit(EXIT_FAILURE);
15     }
16
17     fwrite(buf, sizeof(char), strlen(buf), f);
18     if (ferror(f)) {
19         perror("Error al escribir");
20         exit(EXIT_FAILURE);
21     }
22
23     fclose(f);
24     return 0;
25 }
```

fread

Portada

size_t fread(void *buf, size_t size, size_t cnt, FILE *strm);

Streams

Funciones

fopen
fwrite
fread
fputs
fgets
fprintf
fscanf

Funciones peligrosas

```
1 #define FNAME "file"
2 #define BUFSIZE 256
3
4 int main() {
5     FILE *f;
6     char buf[BUFSIZE];
7     int read;
8
9     f = fopen(FNAME, "r");
10    if (!f) {
11        perror("No se ha podido abrir el fichero");
12        exit(EXIT_FAILURE);
13    }
14
15    read = fread(buf, sizeof(char), BUFSIZE - 1, f);
16    if (ferror(f)) {
17        perror("Error al leer");
18        exit(EXIT_FAILURE);
19    }
20
21    buf[read] = '\0';
22    printf("%s", buf);
23
24    fclose(f);
25    return 0;
26 }
```

fputs

Portada

```
int fputs(const char *str, FILE *strm);
```

Streams

Funciones

fopen
fwrite
fread
fputs
fgets
fprintf
fscanf

Funciones peligrosas

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 #define FNAME "file"
6
7 int main()
8 {
9     FILE *f;
10    char buf[] = "Hola Mundo";
11
12    f = fopen(FNAME, "w+");
13    if (!f) {
14        perror("Error al abrir");
15        exit(EXIT_FAILURE);
16    }
17
18    fputs(buf, f);
19    if (ferror(f)) {
20        perror("Error al escribir");
21        exit(EXIT_FAILURE);
22    }
23
24    fclose(f);
25    return 0;
26 }
```


fgets

Portada

```
char * fgets(char *buf, int bufsize, FILE *strm);
```

Streams

Funciones

fopen
fwrite
fread
fputs
fgets
fprintf
fscanf

Funciones peligrosas

```
1 #define FNAME "file"
2 #define BUFSIZE 256
3
4 int main()
5 {
6     FILE *f;
7     char buf[BUFSIZE];
8
9     f = fopen(FNAME, "r");
10    if (!f) {
11        perror("Error al abrir");
12        exit(EXIT_FAILURE);
13    }
14
15    int i = 0;
16    while (fgets(buf, BUFSIZE, f))
17        printf("%d: \"%s\\n\"", i++, buf);
18    if (ferror(f)) {
19        perror("Error al leer");
20        exit(EXIT_FAILURE);
21    }
22
23    fclose(f);
24    return 0;
25 }
```

fprintf

Portada

```
int fprintf (FILE *strm, const char *frmt, ...);
```

Streams

Funciones

fopen
fwrite
fread
fputs
fgets
fprintf
fscanf

Funciones peligrosas

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 #define FNAME "file"
6
7 int main()
8 {
9     FILE *f;
10
11     f = fopen(FNAME, "w+");
12     if (!f) {
13         perror("Error al abrir");
14         exit(EXIT_FAILURE);
15     }
16
17     fprintf(f, "Hola Mundo\n 2 + 2 = %d", 2 + 2);
18     if (ferror(f)) {
19         perror("Error al escribir");
20         exit(EXIT_FAILURE);
21     }
22
23     fclose(f);
24     return 0;
25 }
```

fscanf

Portada

```
int fscanf(FILE *strm, const char *fmt, ...);
```

Streams

Funciones

fopen
fwrite
fread
fputs
fgets
fprintf
fscanf

Funciones peligrosas

```
1 int main()
2 {
3     FILE *f;
4     char str1[10], str2[10];
5     int a, b, c, ret;
6
7     f = fopen(FNAME, "r");
8     if (!f) {
9         perror("Error al abrir");
10        exit(EXIT_FAILURE);
11    }
12
13    while ((ret = fscanf(f, "%10s %10s %d + %d = %d",
14                        str1, str2, &a, &b, &c)) == 5) {
15        printf("%s %s\n %d + %d = %d\n", str1, str2, a, b, c);
16    }
17    if (ferror(f)) {
18        perror("Error al leer");
19        exit(EXIT_FAILURE);
20    }
21    else if (ret != EOF) {
22        fprintf(stderr, "Error al analizar: %d/%d\n", ret, 5);
23    }
24
25    fclose(f);
26    return 0;
27 }
```

Funciones peligrosas

Portada

Streams

Funciones

fopen
fwrite
fread
fputs
fgets
fprintf
fscanf

Funciones peligrosas

Todas las funciones que lean un número indefinido de bytes sin comprobar el tamaño del buffer de destino.

- `scanf` y sus variantes si no especificamos el tamaño al escanear una cadena: `"%S"`
- `gets`, que lee una cadena de `stdin` sin posibilidad de especificar ningún límite para su tamaño. Eliminada en el estándar de 2011.