



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

<i>Profesor:</i>	M.I. Marco Antonio Martínez Quintana
<i>Asignatura:</i>	Fundamentos de Programación
<i>Grupo:</i>	03
<i>No de Práctica(s):</i>	13
<i>Integrante(s):</i>	Teran García Rodolfo Mario
<i>No. de Equipo de cómputo empleado:</i>	No aplica
<i>No. de Lista o Brigada:</i>	
<i>Semestre:</i>	2021-1
<i>Fecha de entrega:</i>	18/01/2021
<i>Observaciones:</i>	M.I. Marco Antonio Martínez Quintana

Calificación:

Practica #13: Lectura y escritura de datos

Objetivos:

Elaborar programas en lenguaje C que requieran el uso de archivos de texto plano en la resolución de problemas, entendiendo a los archivos como un elemento de almacenamiento secundario.

Introducción:

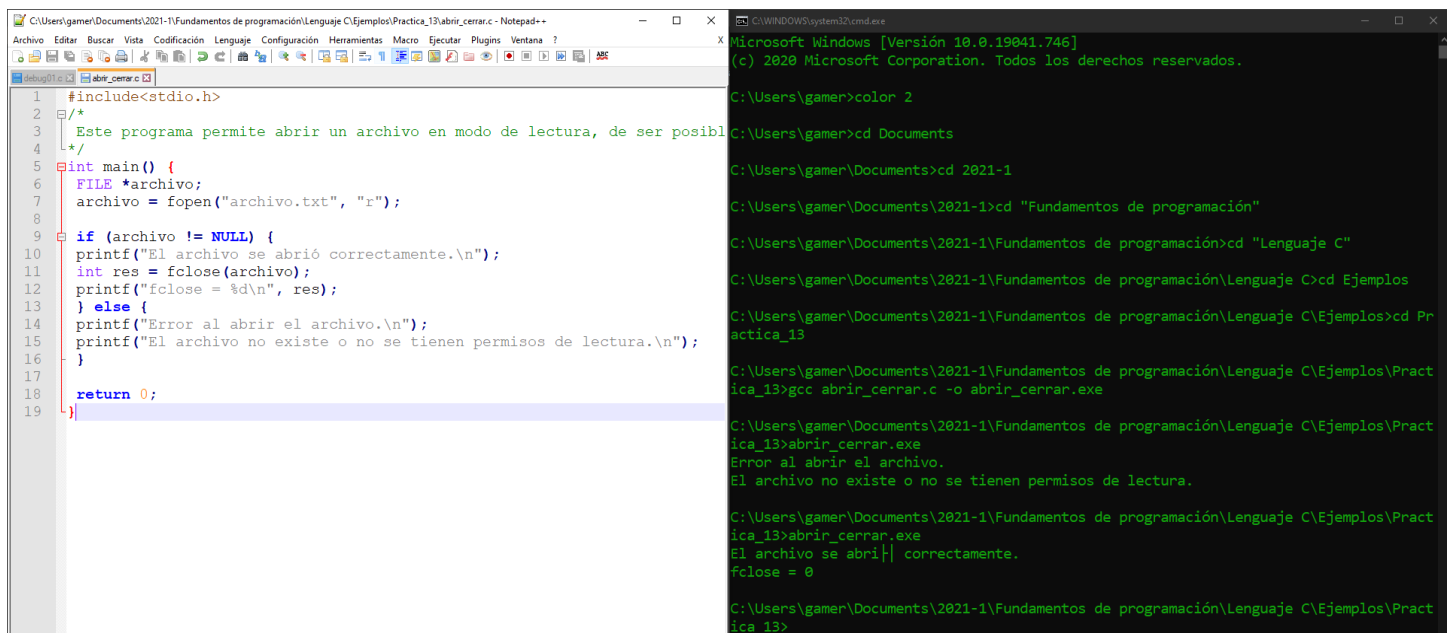
Un archivo es un conjunto de datos estructurados en una colección de entidades elementales o básicas denominadas registros que son del mismo tipo, pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Lenguaje C permite manejar la entrada y la salida de datos desde o hacia un archivo, respectivamente, a través del uso de la biblioteca de funciones de la cabecera `stdio.h`.

Desarrollo:

Apuntador a archivo: Un apuntador a un archivo es un hilo común que unifica el sistema de Entrada/Salida (E/S) con un buffer donde se transportan los datos. Un apuntador a archivo señala a la información que contiene y define ciertas características sobre él, incluyendo el nombre, el estado y la posición actual del archivo. Los apuntadores a un archivo se manejan en lenguaje C como variables apuntador de tipo `FILE` que se define en la cabecera `stdio.h`

Abrir archivo: La función `fopen()` abre una secuencia para que pueda ser utilizada y la asocia a un archivo.

Cerrar archivo: La función `fclose()` cierra una secuencia que fue abierta mediante una llamada a `fopen()`. Escribe la información que se encuentre en el buffer al disco y realiza un cierre formal del archivo a nivel del sistema operativo. Un error en el cierre de una secuencia puede generar todo tipo de problemas, incluyendo la pérdida de datos, destrucción de archivos y posibles errores intermitentes en el programa.



The image shows a C program in a Notepad++ window and its execution in a Windows Command Prompt.

Notepad++ Window:

```
1 #include<stdio.h>
2
3 /*
4  * Este programa permite abrir un archivo en modo de lectura, de ser posible
5  */
6 int main() {
7     FILE *archivo;
8     archivo = fopen("archivo.txt", "r");
9
10    if (archivo != NULL) {
11        printf("El archivo se abrió correctamente.\n");
12        int res = fclose(archivo);
13        printf("fclose = %d\n", res);
14    } else {
15        printf("Error al abrir el archivo.\n");
16        printf("El archivo no existe o no se tienen permisos de lectura.\n");
17    }
18    return 0;
19 }
```

Windows Command Prompt Window:

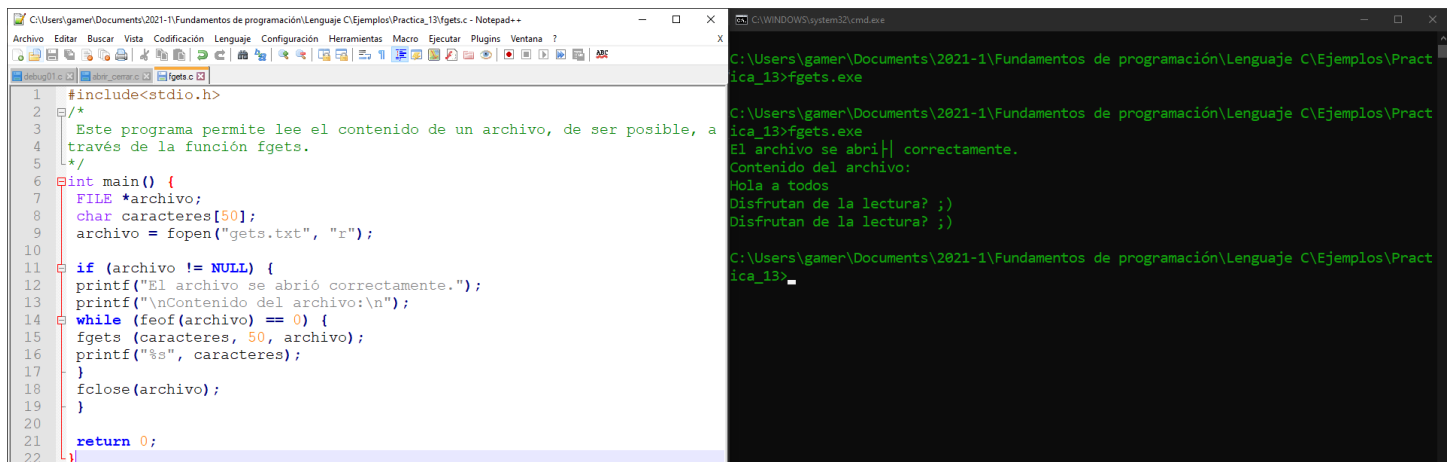
```
Microsoft Windows [Versión 10.0.19041.746]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

C:\Users\gamer>color 2
C:\Users\gamer>cd Documents
C:\Users\gamer\Documents>cd 2021-1
C:\Users\gamer\Documents\2021-1>cd "Fundamentos de programación"
C:\Users\gamer\Documents\2021-1\Fundamentos de programación>cd "Lenguaje C"
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C>cd Ejemplos
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos>cd Pr
actica_13
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Pract
ica_13>gcc abrir_cerrar.c -o abrir_cerrar.exe
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Pract
ica_13>abrir_cerrar.exe
Error al abrir el archivo.
El archivo no existe o no se tienen permisos de lectura.

C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Pract
ica_13>abrir_cerrar.exe
El archivo se abrió| correctamente.
fclose = 0

C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Pract
ica_13>
```

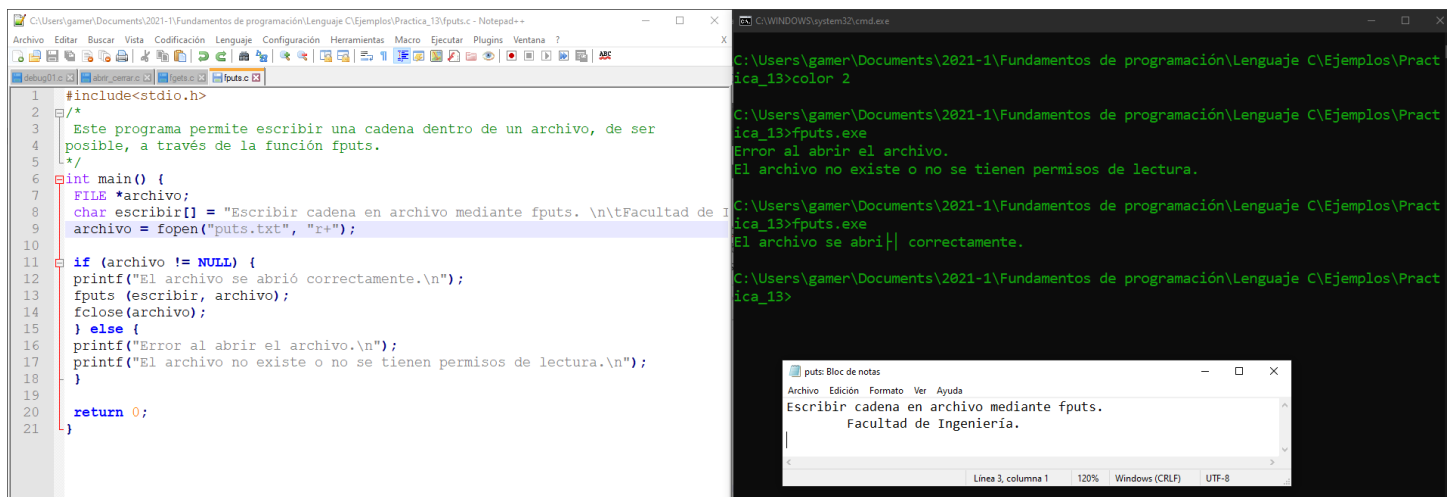
Funciones fgets y fputs: Las funciones fgets() y fputs() pueden leer y escribir, respectivamente, cadenas sobre los archivos. La función fputs() permite escribir una cadena en un archivo específico. La función fgets() permite leer una cadena desde el archivo especificado. Esta función lee un renglón a la vez.



The screenshot shows a C program in Notepad++ and its execution in a Windows command prompt. The program, named 'fgets.c', includes <stdio.h> and has a comment explaining its purpose: to read the content of a file using fgets(). The main function opens 'gets.txt' in read mode. It checks if the file was opened successfully and then enters a loop where it reads lines from the file using fgets(), printing each line. The command prompt shows the execution of 'C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>fgets.exe', which outputs the content of 'gets.txt': 'El archivo se abrió correctamente.', 'Contenido del archivo:', 'Hola a todos', 'Disfrutan de la lectura? ;)', and 'Disfrutan de la lectura? ;)'.

```
1 #include<stdio.h>
2 /*
3  Este programa permite leer el contenido de un archivo, de ser posible, a
4  través de la función fgets.
5  */
6 int main() {
7     FILE *archivo;
8     char caracteres[50];
9     archivo = fopen("gets.txt", "r");
10
11     if (archivo != NULL) {
12         printf("El archivo se abrió correctamente.");
13         printf("\nContenido del archivo:\n");
14         while (feof(archivo) == 0) {
15             fgets (caracteres, 50, archivo);
16             printf("%s", caracteres);
17         }
18         fclose(archivo);
19     }
20
21     return 0;
22 }
```

```
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>fgets.exe
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>fgets.exe
El archivo se abrió correctamente.
Contenido del archivo:
Hola a todos
Disfrutan de la lectura? ;)
Disfrutan de la lectura? ;)
```

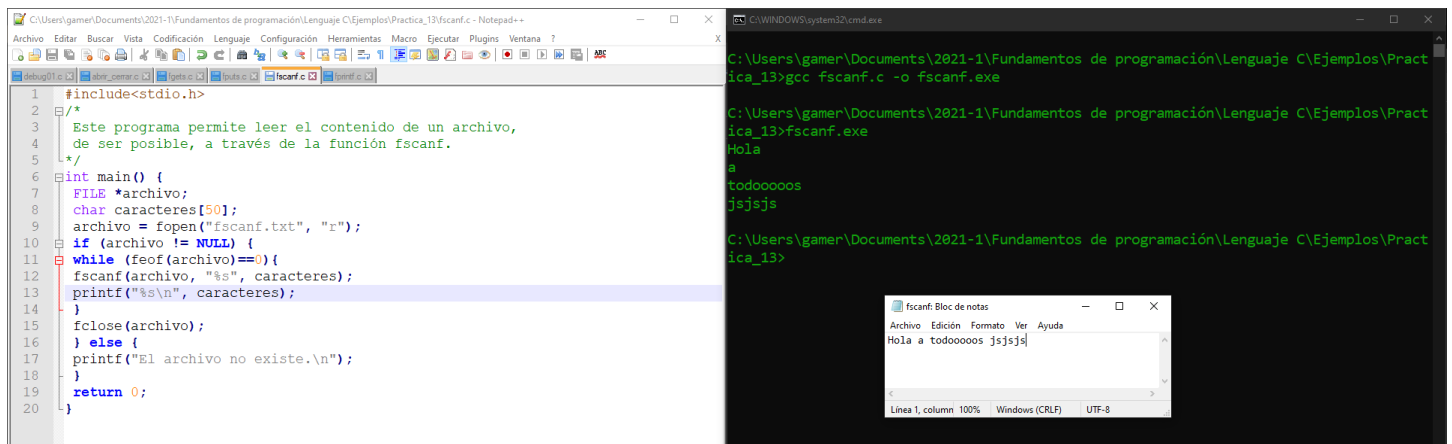


The screenshot shows a C program in Notepad++ and its execution in a Windows command prompt. The program, named 'fputs.c', includes <stdio.h> and has a comment explaining its purpose: to write a string to a file using fputs(). The main function opens 'puts.txt' in read mode (though the code says 'r+', it likely means 'w' for write). It checks if the file was opened successfully and then writes the string 'Escribir cadena en archivo mediante fputs. \n\tFacultad de Ingeniería.' to the file using fputs(). The command prompt shows the execution of 'C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>fputs.exe', which outputs 'Error al abrir el archivo. El archivo no existe o no se tienen permisos de lectura.' and 'El archivo se abrió correctamente.'.

```
1 #include<stdio.h>
2 /*
3  Este programa permite escribir una cadena dentro de un archivo, de ser
4  posible, a través de la función fputs.
5  */
6 int main() {
7     FILE *archivo;
8     char escribir[] = "Escribir cadena en archivo mediante fputs. \n\tFacultad de Ingeniería.";
9     archivo = fopen("puts.txt", "r+");
10
11     if (archivo != NULL) {
12         printf("El archivo se abrió correctamente.\n");
13         fputs (escribir, archivo);
14         fclose(archivo);
15     } else {
16         printf("Error al abrir el archivo.\n");
17         printf("El archivo no existe o no se tienen permisos de lectura.\n");
18     }
19
20     return 0;
21 }
```

```
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>color 2
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>fputs.exe
Error al abrir el archivo.
El archivo no existe o no se tienen permisos de lectura.
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>fputs.exe
El archivo se abrió correctamente.
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>
```

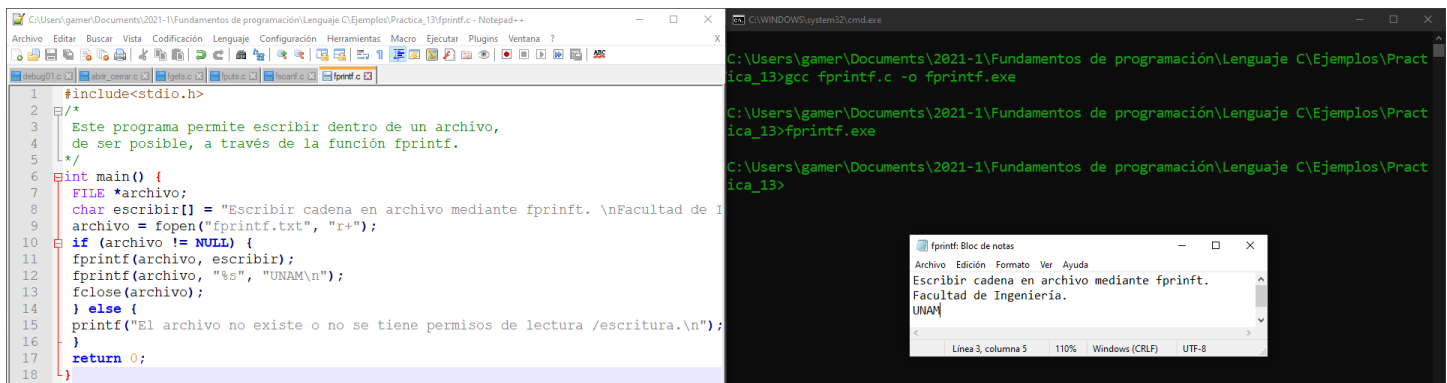
Funciones fscanf y fprintf: Las funciones fprintf() y fscanf() se comportan exactamente como printf() (imprimir) y scanf() (leer), excepto que operan sobre archivo.



The screenshot shows a C program in Notepad++ and its execution in a Windows command prompt. The program, named 'fscanf.c', includes <stdio.h> and has a comment explaining its purpose: to read the content of a file using fscanf(). The main function opens 'fscanf.txt' in read mode. It checks if the file was opened successfully and then enters a loop where it reads lines from the file using fscanf(), printing each line. The command prompt shows the execution of 'C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>gcc fscanf.c -o fscanf.exe' and 'C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>fscanf.exe', which outputs the content of 'fscanf.txt': 'Hola a todos', 'jsjsjs', and 'jsjsjs'.

```
1 #include<stdio.h>
2 /*
3  Este programa permite leer el contenido de un archivo,
4  de ser posible, a través de la función fscanf.
5  */
6 int main() {
7     FILE *archivo;
8     char caracteres[50];
9     archivo = fopen("fscanf.txt", "r");
10     if (archivo != NULL) {
11         while (feof(archivo) == 0) {
12             fscanf(archivo, "%s", caracteres);
13             printf("%s\n", caracteres);
14         }
15         fclose(archivo);
16     } else {
17         printf("El archivo no existe.\n");
18     }
19     return 0;
20 }
```

```
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>gcc fscanf.c -o fscanf.exe
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>fscanf.exe
Hola
a
todooooos
jsjsjs
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>
```



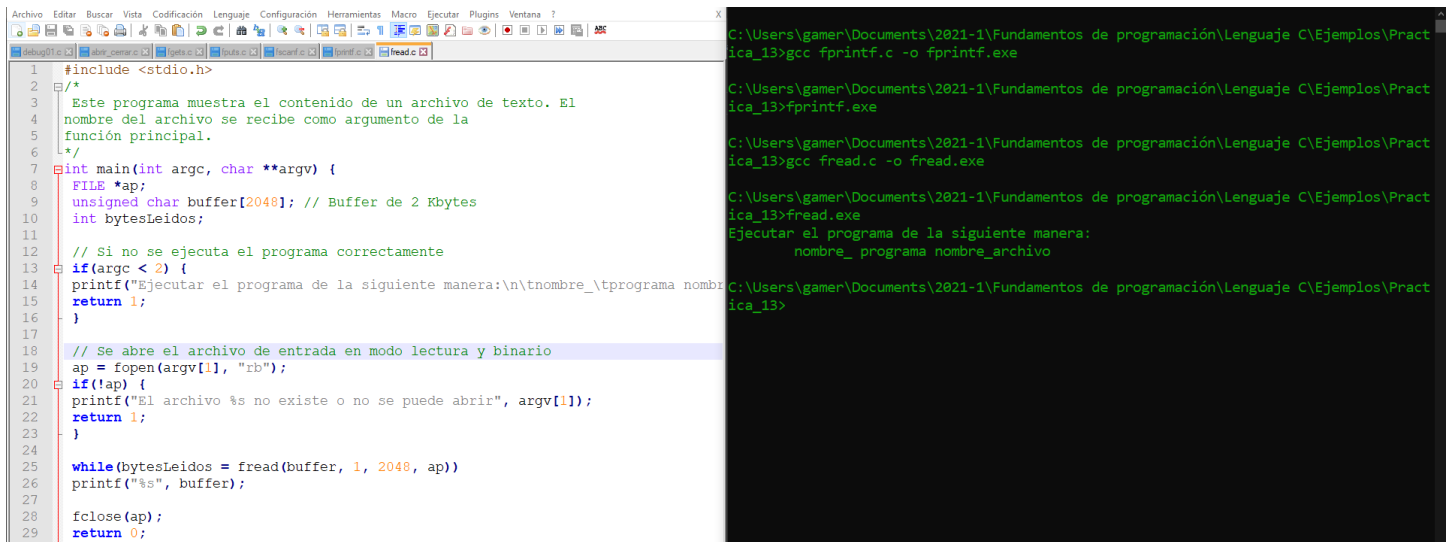
```
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13\fprintf.c - Notepad++
1 #include<stdio.h>
2
3 /*
4  Este programa permite escribir dentro de un archivo,
5  de ser posible, a través de la función fprintf.
6  */
7
8 int main() {
9     FILE *archivo;
10    char escribir[] = "Escribir cadena en archivo mediante fprintf. \nFacultad de Ingeniería.";
11    archivo = fopen("fprintf.txt", "w+");
12    if (archivo != NULL) {
13        fprintf(archivo, escribir);
14        fprintf(archivo, "%s", "UNAM\n");
15        fclose(archivo);
16    } else {
17        printf("El archivo no existe o no se tiene permisos de lectura /escritura.\n");
18    }
19    return 0;
20 }
```

```
C:\WINDOWS\system32\cmd.exe
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>gcc fprintf.c -o fprintf.exe
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>fprintf.exe
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>

fprintf: Bloc de notas
Archivo Edición Formato Ver Ayuda
Escribir cadena en archivo mediante fprintf.
Facultad de Ingeniería.
UNAM
Línea 3, columna 5 110% Windows (CRLF) UTF-8
```

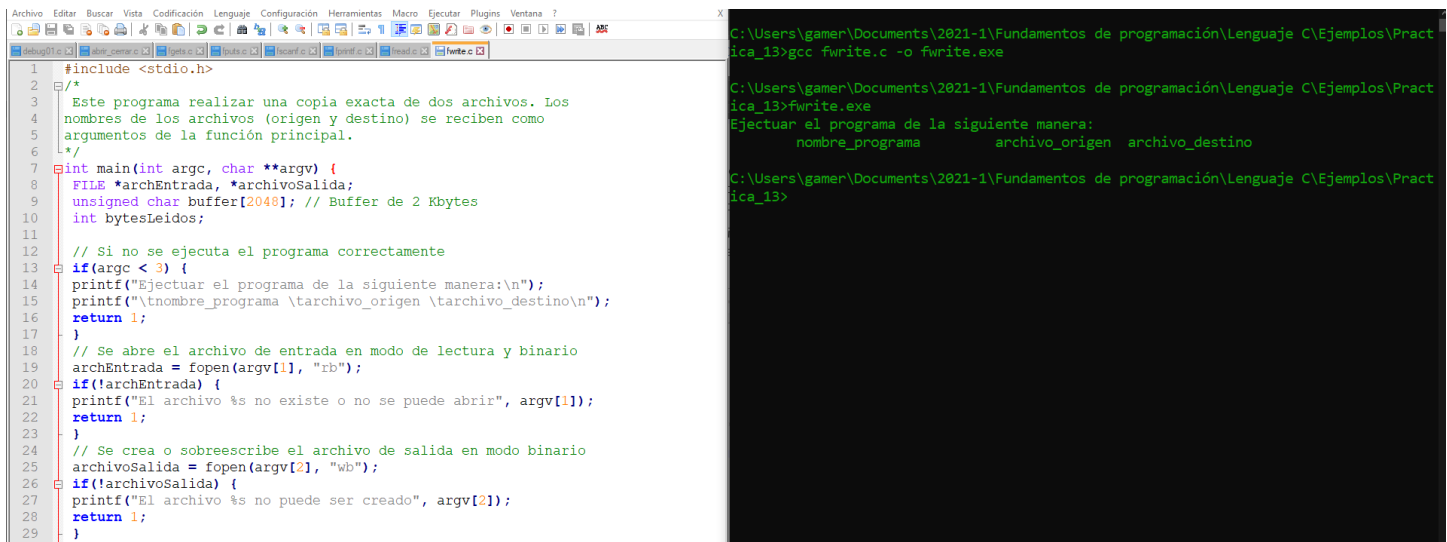
Funciones fread y fwrite: fread y fwrite son funciones que permiten trabajar con elementos de longitud conocida. fread permite leer uno o varios elementos de la misma longitud a partir de una dirección de memoria determinada (apuntador). El valor de retorno es el número de elementos (bytes) leídos.

fwrite permite escribir hacia un archivo uno o varios elementos de la misma longitud almacenados a partir de una dirección de memoria determinada. El valor de retorno es el número de elementos escritos.



```
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ?
1 #include <stdio.h>
2
3 /*
4  Este programa muestra el contenido de un archivo de texto. El
5  nombre del archivo se recibe como argumento de la
6  función principal.
7  */
8
9 int main(int argc, char **argv) {
10    FILE *ap;
11    unsigned char buffer[2048]; // Buffer de 2 Kbytes
12    int bytesLeídos;
13
14    // Si no se ejecuta el programa correctamente
15    if(argc < 2) {
16        printf("Ejecutar el programa de la siguiente manera:\n\\nombre_tprograma nombre_archivo");
17        return 1;
18    }
19
20    // Se abre el archivo de entrada en modo lectura y binario
21    ap = fopen(argv[1], "rb");
22    if(!ap) {
23        printf("El archivo %s no existe o no se puede abrir", argv[1]);
24        return 1;
25    }
26
27    while(bytesLeídos = fread(buffer, 1, 2048, ap))
28        printf("%s", buffer);
29
30    fclose(ap);
31    return 0;
32 }
```

```
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>gcc fread.c -o fread.exe
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>fread.exe
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>gcc fread.c -o fread.exe
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>fread.exe
Ejecutar el programa de la siguiente manera:
    nombre_programa nombre_archivo
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>
```



```
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ?
1 #include <stdio.h>
2
3 /*
4  Este programa realizar una copia exacta de dos archivos. Los
5  nombres de los archivos (origen y destino) se reciben como
6  argumentos de la función principal.
7  */
8
9 int main(int argc, char **argv) {
10    FILE *archEntrada, *archivosalida;
11    unsigned char buffer[2048]; // Buffer de 2 Kbytes
12    int bytesLeídos;
13
14    // Si no se ejecuta el programa correctamente
15    if(argc < 3) {
16        printf("Ejecutar el programa de la siguiente manera:\n");
17        printf("\\nombre_programa \\archivo_origen \\archivo_destino\n");
18        return 1;
19    }
20
21    // Se abre el archivo de entrada en modo de lectura y binario
22    archEntrada = fopen(argv[1], "rb");
23    if(!archEntrada) {
24        printf("El archivo %s no existe o no se puede abrir", argv[1]);
25        return 1;
26    }
27
28    // Se crea o sobrescribe el archivo de salida en modo binario
29    archivosalida = fopen(argv[2], "wb");
30    if(!archivosalida) {
31        printf("El archivo %s no puede ser creado", argv[2]);
32        return 1;
33    }
34
35    while(bytesLeídos = fread(buffer, 1, 2048, archEntrada))
36        fwrite(buffer, 1, bytesLeídos, archivosalida);
37
38    fclose(archEntrada);
39    fclose(archivosalida);
40    return 0;
41 }
```

```
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>gcc fwrite.c -o fwrite.exe
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>fwrite.exe
Ejecutar el programa de la siguiente manera:
    nombre_programa archivo_origen archivo_destino
C:\Users\gamer\Documents\2021-1\Fundamentos de programación\Lenguaje C\Ejemplos\Practica_13>
```

Conclusiones:

Este tema como todos los vistos a lo largo y ancho de las diferentes prácticas ha sido sumamente entretenido, tanto por contenido como por la cantidad de retos complejos que presenta para resolver, en un inicio me pareció un tanto difícil de analizar y comprender, pero una vez empezar a encontrar el hilo de la temática que toma esta práctica resulta difícil perderse, pues todo lo que nos presenta son sin excepción conocimiento útil y necesario para la vida en general, comenzando por todas las posibles situaciones en las que realmente tendremos que hacer uso de estas herramientas en la vida profesional, cuando no tenemos acceso a un editor de texto o a una terminal más avanzada que la predeterminada por la bios o ciertos sistemas operativos, de ahí que resulte tan importante encontrarse familiarizado con todo tipo de interacciones que se pueden presentar en las mismas, viendo al lenguaje de programación C como un medio para hacer prevalecer el conocimiento y la reinención humana a través de la codificación de ideas en bits e instrucciones no tan difíciles de comprender, pero que al hacerlo, genera cierta sensación de bienestar y de conexión con el mundo cada vez menos oculto de la informática aplicada a la ciencia y la industria. Al ser esta la última práctica, he de decir que admirablemente resultó ser un buen cierre, pues abre la mente y la perspectiva a todo aquello que seguramente se ignora o desconoce de manera involuntaria, y a su vez nos recuerda que hay un mundo aún por delante, sin olvidar todo lo que se ha recorrido ya.