



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Marco Antonio Martínez Quintana

*Asignatura:* Fundamentos de Programación

*Grupo:* 3

*No de Práctica(s):* 13

*Integrante(s):* Ramírez Pérez Daniela Itzel

*No. de Equipo de  
cómputo empleado:* No aplica

*No. de Lista o Brigada:* 36

*Semestre:* 1er

*Fecha de entrega:* Viernes 22 de Diciembre

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## Objetivo:

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 del mismo tipo 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**.

## 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.

La sintaxis para obtener una variable apuntador de archivo es la siguiente:

FILE \*F;

## Abrir archivo

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

Su estructura es la siguiente:

\*FILE fopen(char \*nombre\_archivo, char \*modo);

Donde **nombre\_archivo** es un puntero a una cadena de caracteres que representan un nombre válido del archivo y puede incluir una especificación del directorio. La cadena a la que apunta modo determina cómo se abre el archivo.

Existen diferentes modos de apertura de archivos, los cuales se mencionan a continuación, además de que se pueden utilizar más de uno solo:

r	Abre un archivo de texto para lectura.
w	Crea un archivo de texto para escritura.
a	Abre un archivo de texto para añadir.
r+	Abre un archivo de texto para lectura / escritura.
w+	Crea un archivo de texto para lectura / escritura
a+	Añade o crea un archivo de texto para lectura / escritura.
rb	Abre un archivo en modo lectura y binario.
wb	Crea un archivo en modo escritura y binario.

## Cerrar archivo

La función `fclose()` cierra una secuencia que fue abierta mediante la función `fopen()`.

La firma de esta función es:

int fclose(FILE \*apArch);

Si se devuelve un valor cero significa que la operación de cierre ha tenido éxito.

## Código (abrir cerrar archivo)

```
#include<stdio.h>
/*
Este programa permite abrir un archivo en modo de lectura, de ser posible.
*/
int main() {
    FILE *archivo;
    archivo = fopen("archivo.txt", "r");

    if (archivo != NULL) {
        printf("El archivo se abrió correctamente.\n");
        int res = fclose(archivo);
        printf("fclose = %d\n", res);
    } else {
        printf("Error al abrir el archivo.\n");
        printf("El archivo no existe o no se tienen permisos de lectura.\n");
    }

    return 0;
}
```

## Funciones `fgets` y `fputs`

<b>Fgets</b> Escribe una cadena en un archivo especifico	<code>char *fgets(char *buffer, int tamaño, FILE *apArch);</code>
<b>Fputs</b> Lee una cadena desde el archivo especificado. Esta función lee un renglón a la vez.	<code>char *fputs(char *buffer, FILE *apArch);</code>

## Código (fgets)

```
#include<stdio.h>
/*
    Este programa permite leer el contenido de un archivo, de ser posible, a
    través de la función fgets.
*/
int main() {
    FILE *archivo;
    char caracteres[50];
    archivo = fopen("gets.txt", "r");

    if (archivo != NULL) {
        printf("El archivo se abrió correctamente.");
        printf("\nContenido del archivo:\n");
        while (feof(archivo) == 0) {
            fgets (caracteres, 50, archivo);
            printf("%s", caracteres);
        }
        fclose(archivo);
    }

    return 0;
}
```

## Código (fputs)

```
#include<stdio.h>
/*
    Este programa permite escribir una cadena dentro de un archivo, de ser
    posible, a través de la función fputs.
*/
int main() {
    FILE *archivo;
    char escribir[] = "Escribir cadena en archivo mediante fputs. \n\tFacultad
de Ingeniería.\n";
    archivo = fopen("puts.txt", "r+");

    if (archivo != NULL) {
        printf("El archivo se abrió correctamente.\n");
        fputs (escribir, archivo);
        fclose(archivo);
    } else {
        printf("Error al abrir el archivo.\n");
        printf("El archivo no existe o no se tienen permisos de lectura.\n");
    }

    return 0;
}
```

## Funciones fscanf y fprintf

<b>fscanf</b> (leer)	int fscanf(FILE *apArch, char *formato, ...);
<b>fprintf</b> (imprimir)	int fprintf(FILE *apArch, char *formato, ...);

- **fprintf() y fscanf()** : dirigen sus operaciones de E/S al archivo al que apunta apArch.
- **Formato:** cadena que puede incluir texto o especificadores de impresión de variables.

En los puntos suspensivos se agregan las variables cuyos valores se quieren escribir en el archivo.

### Código (fscanf)

```
#include<stdio.h>
/*
Este programa permite leer el contenido de un archivo,
de ser posible, a través de la función fscanf.
*/
int main() {
    FILE *archivo;
    char caracteres[50];
    archivo = fopen("fscanf.txt", "r");
    if (archivo != NULL) {
        while (feof(archivo)==0){
            fscanf(archivo, "%s", caracteres);
            printf("%s\n", caracteres);
        }
        fclose(archivo);
    } else {
        printf("El archivo no existe.\n");
    }
    return 0;
}
```

### Código (fprintf)

```
#include<stdio.h>
/*
Este programa permite escribir dentro de un archivo,
de ser posible, a través de la función fprintf.
*/
int main() {
    FILE *archivo;
    char escribir[] = "Escribir cadena en archivo mediante fprintf. \nFacultad
de Ingeniería.\n";
    archivo = fopen("fprintf.txt", "r+");
    if (archivo != NULL) {
        fprintf(archivo, escribir);
        fprintf(archivo, "%s", "UNAM\n");
        fclose(archivo);
    } else {
        printf("El archivo no existe o no se tiene permisos de lectura /
escritura.\n");
    }
    return 0;
}
```

## Funciones fread y fwrite

<b>fread</b> Lee uno o varios elementos de la misma longitud a partir de una dirección de memoria determinada	<code>int fread(void *ap, size_t tam, size_t nelem, FILE *archivo)</code>  El valor de retorno es el número de elementos (bytes) leídos
<b>fwrite</b> Escribe hacia un archivo uno o varios elementos de la misma longitud almacenados a partir de una dirección de memoria determinada.	<code>int fwrite(void *ap, size_t tam, size_t nelem, FILE *archivo)</code>  El valor de retorno es el número de elementos escritos

## Código (fread)

```
#include <stdio.h>
/*
Este programa muestra el contenido de un archivo de texto. El
nombre del archivo se recibe como argumento de la
función principal.
*/
int main(int argc, char **argv) {
    FILE *ap;
    unsigned char buffer[2048]; // Buffer de 2 Kbytes
    int bytesLeidos;

    // Si no se ejecuta el programa correctamente
    if(argc < 2) {
        printf("Ejecutar el programa de la siguiente
manera:\n\tnombre_\tprograma nombre_archivo\n");
        return 1;
    }

    // Se abre el archivo de entrada en modo lectura y binario
    ap = fopen(argv[1], "rb");
    if(!ap) {
        printf("El archivo %s no existe o no se puede abrir", argv[1]);
        return 1;
    }

    while(bytesLeidos = fread(buffer, 1, 2048, ap))
        printf("%s", buffer);

    fclose(ap);
    return 0;
}
```

## Código (fwrite)

```
#include <stdio.h>
/*
Este programa realizar una copia exacta de dos archivos. Los
nombres de los archivos (origen y destino) se reciben como
argumentos de la función principal.
*/
int main(int argc, char **argv) {
    FILE *archEntrada, *archivoSalida;
    unsigned char buffer[2048]; // Buffer de 2 Kbytes
    int bytesLeidos;

    // Si no se ejecuta el programa correctamente
    if(argc < 3) {
        printf("Ejectuar el programa de la siguiente manera:\n");
        printf("\tnombre_programa \tarchivo_origen \tarchivo_destino\n");
        return 1;
    }
    // Se abre el archivo de entrada en modo de lectura y binario
    archEntrada = fopen(argv[1], "rb");
    if(!archEntrada) {
        printf("El archivo %s no existe o no se puede abrir", argv[1]);
        return 1;
    }
    // Se crea o sobrescribe el archivo de salida en modo binario
    archivoSalida = fopen(argv[2], "wb");
    if(!archivoSalida) {
        printf("El archivo %s no puede ser creado", argv[2]);
        return 1;
    }

    // Copia archivos
    while (bytesLeidos = fread(buffer, 1, 2048, archEntrada))
        fwrite(buffer, 1, bytesLeidos, archivoSalida);

    // Cerrar archivos
    fclose(archEntrada);
    fclose(archivoSalida);
    return 0;
}
```

## Archivos Ejercicios

```
#include<stdio.h>
int main()
{
    //Declarar variable
    char au=163, sp=168, aa=160;
    int n,res;
    //Apuntador a archivo
    FILE *a;
    a=fopen("resultadosGauss.txt","w");
    //Mensaje de bienvenida
    printf("\n\n\t\tSuma de los primeros n n%cmeros\n\n",au);

    //Solicitar el numero de los elementos a sumar
    printf("%cCu%ontos n%cmeros deseas sumar?",sp,aa,au);
    scanf("%d",&n);

    //Sumar los n numeros
    res=0;
    for(int i=1;i<=n;i++)
    {
        fprintf(a,"%d + %d = ",res,i);
        res=res+i;
        fprintf(a,"%d\n",res);
    }

    //Mostrar el resultado
    printf("La suma de los primeros %d n%cmeros es: %d\n",n,au,res);
    fclose(a);
    return 0;
}
```

```
0 + 1 = 1
1 + 2 = 3
3 + 3 = 6
6 + 4 = 10
10 + 5 = 15
```

```
1 #include<stdio.h>
2 int main()
3 {
4     //Declarar variable
5     char au=163, sp=168, aa=160, ae=130;
6     int n,res;
7
8     //Apuntador a archivo
9     FILE *a;
10    a=fopen("factorial.txt","w");
11
12    //Mensaje de bienvenida
13    printf("\n\n\t\tFactorial de n%cmero n\n\n",au);
14
15    //Solicitar el numero de los elementos a sumar
16    printf("%cDe qu%c n%cmero se calculara la factorial?",sp,ae,au);
17    scanf("%d",&n);
18
19    //Sumar los n numeros
20    res=1;
21    for(int i=1;i<=n;i++)
22    {
23        fprintf(a,"%d + %d = ",res,n);
24        res=res*i;
25        fprintf(a,"%d\n",res);
26    }
27
28    //Mostrar el resultado
29    printf("La factorial de %d es: %d\n",n,res);
30
31    fclose(a);
32    return 0;
33 }
34
```

```
1 + 5 = 1
1 + 5 = 2
2 + 5 = 6
6 + 5 = 24
24 + 5 = 120
```



## **Conclusion**

Con la practica adquirimos más conocimiento en la forma en la cual el lenguaje C puede ser usado para crear y modificar archivos para nuestros proyectos, la ventaja que nos traen estos archivos creados es la capacidad de que podamos resolver problemas o simplemente como funcionan la resolución de los problemas gracias al almacenamiento secundario que trae la inclusión de estos archivos.

## **Referencias**

- Nakayama Cervantes, A., Castañeda Perdomo, M., Solano Gálvez, J. A., García Cano, E. E., Sandoval Montaña, L., & Arteaga Ricci, T. I. (2018, 6 abril). Manual de prácticas del laboratorio de Fundamentos de programación (p.135-147). Laboratorio de Computación Salas A y B. [http://odin.fib.unam.mx/salac/practicasp/MADO-17\\_FP.pdf](http://odin.fib.unam.mx/salac/practicasp/MADO-17_FP.pdf)