



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): Emiliano Guevara Chávez

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

No. De Lista o Brigada: 20

Semestre: Ciclo 2021-1

Fecha de entrega: 18/01/2021

Observaciones: _____

CALIFICACIÓN: _____

Práctica #13

Lectura y escritura de datos

Objetivo:

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

Introducción:

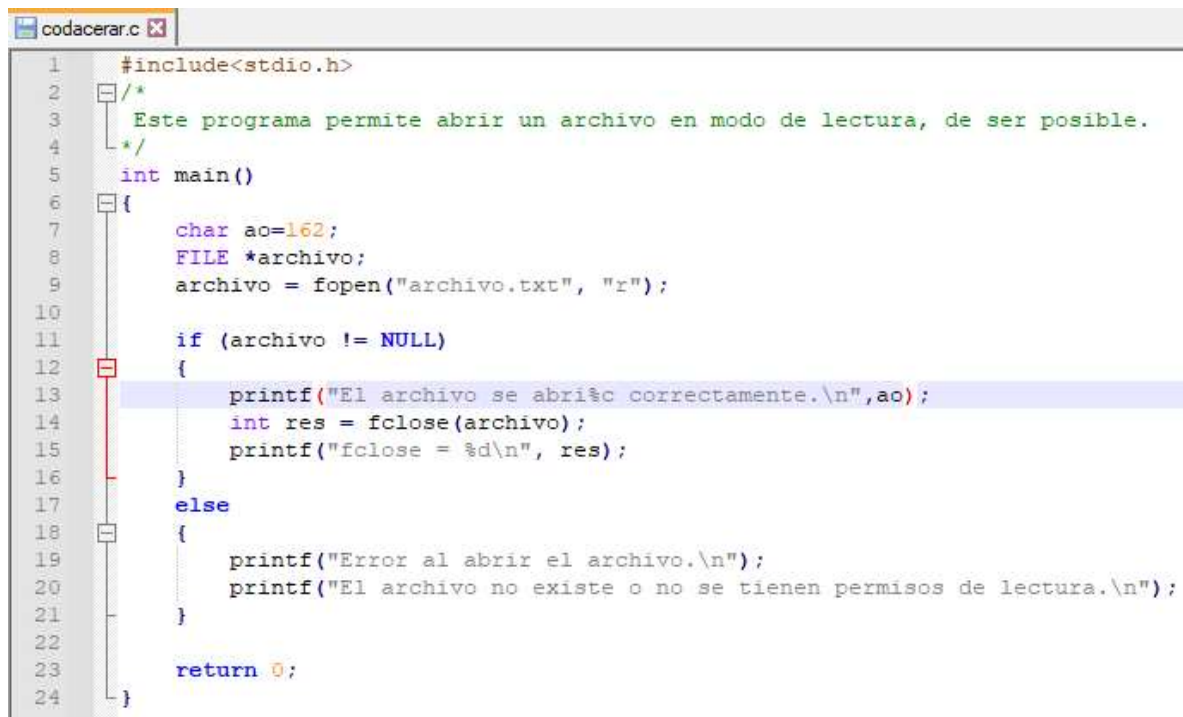
Un archivo es un conjunto de datos estructurado son una colección de entidades elementales o básicas denominadas registros que son del mismo tipo, pertenecientes a un mismo contexto y almacenados sistemáticas para su posterior uso.

Lenguaje C permite manejar la entrada y la salida de datos desde o hacia un archivo, respectivamente, a través del uso de ña biblioteca de funciones de la cabecera stdio.h.

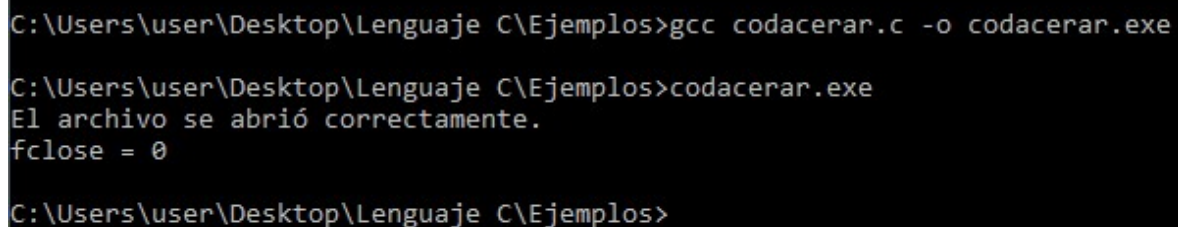
Desarrollo de actividades:

Cerrar archivo

Código (abrir cerrar archivo)



```
1  #include<stdio.h>
2  /*
3   * Este programa permite abrir un archivo en modo de lectura, de ser posible.
4   */
5  int main()
6  {
7      char ao=162;
8      FILE *archivo;
9      archivo = fopen("archivo.txt", "r");
10
11      if (archivo != NULL)
12      {
13          printf("El archivo se abrió correctamente.\n",ao);
14          int res = fclose(archivo);
15          printf("fclose = %d\n", res);
16      }
17      else
18      {
19          printf("Error al abrir el archivo.\n");
20          printf("El archivo no existe o no se tienen permisos de lectura.\n");
21      }
22
23      return 0;
24  }
```



```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>gcc codacerar.c -o codacerar.exe
C:\Users\user\Desktop\Lenguaje C\Ejemplos>codacerar.exe
El archivo se abrió correctamente.
fclose = 0
C:\Users\user\Desktop\Lenguaje C\Ejemplos>
```

Funciones fgets y fputs

Código (fgets)

```
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 {
8     char ao=162;
9     FILE *archivo;
10    char caracteres[50];
11    archivo = fopen("gets.txt", "r");
12
13    if (archivo != NULL)
14    {
15        printf("El archivo se abrió correctamente.",ao);
16        printf("\nContenido del archivo:\n");
17        while (feof(archivo) == 0)
18        {
19            fgets (caracteres, 50, archivo);
20            printf("%s", caracteres);
21        }
22        fclose(archivo);
23    }
24
25    return 0;
26 }
```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>gcc cofgets.c -o cofgets.exe
```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>cofgets.exe
```

```
El archivo se abrió correctamente.
```

```
Contenido del archivo:
```

```
hola
```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>_
```

Código (fputs)

```
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 {
8     char ao=162;
9     char ai=161;
10    FILE *archivo;
11    char escribir[] = "Escribir cadena en archivo mediante fputs. \n\tFacultad de Ingenieria.\n";
12    archivo = fopen("puts.txt", "r+");
13
14    if (archivo != NULL)
15    {
16        printf("El archivo se abrió correctamente.\n",ao);
17        fputs (escribir, archivo);
18        fclose(archivo);
19    }
20    else
21    {
22        printf("Error al abrir el archivo.\n");
23        printf("El archivo no existe o no se tienen permisos de lectura.\n");
24    }
25
26    return 0;
27 }
28
```

```

C:\Users\user\Desktop\Lenguaje C\Ejemplos>gcc cofputs.c -o cofputs.exe

C:\Users\user\Desktop\Lenguaje C\Ejemplos>cofputs.exe
Error al abrir el archivo.
El archivo no existe o no se tienen permisos de lectura.

C:\Users\user\Desktop\Lenguaje C\Ejemplos>cofputs.exe
El archivo se abrió correctamente.

C:\Users\user\Desktop\Lenguaje C\Ejemplos>

```

Funciones fscanf y fprintf

Código (fscanf)



```

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  {
8      FILE *archivo;
9      char caracteres[50];
10     archivo = fopen("fscanf.txt", "r");
11     if (archivo != NULL)
12     {
13         while (feof(archivo)==0)
14         {
15             fscanf(archivo, "%s", caracteres);
16             printf("%s\n", caracteres);
17         }
18         fclose(archivo);
19     }
20     else
21     {
22         printf("El archivo no existe.\n");
23     }
24     return 0;
25 }
26

```

```

C:\Users\user\Desktop\Lenguaje C\Ejemplos>gcc cofscanf.c -o cofscanf.exe

C:\Users\user\Desktop\Lenguaje C\Ejemplos>cofscanf.exe
El archivo no existe.

C:\Users\user\Desktop\Lenguaje C\Ejemplos>cofscanf.exe
Tarea
de
programaci|n

C:\Users\user\Desktop\Lenguaje C\Ejemplos>

```

Código (fprintf)

```
1 #include<stdio.h>
2 /*
3  Este programa permite escribir dentro de un archivo,
4  de ser posible, a través de la función fprintf.
5  */
6 int main()
7 {
8     FILE *archivo;
9     char escribir[] = "Escribir cadena en archivo mediante fprintf. \nFacultad de Ingenieria.\n";
10    archivo = fopen("fprintf.txt", "r+");
11    if (archivo != NULL)
12    {
13        fprintf(archivo, escribir);
14        fprintf(archivo, "%s", "UNAM\n");
15        fclose(archivo);
16    }
17    else
18    {
19        printf("El archivo no existe o no se tiene permisos de lectura /escritura.\n");
20    }
21    return 0;
22 }
```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>gcc cofprintf.c -o cofprintf.exe
C:\Users\user\Desktop\Lenguaje C\Ejemplos>cofprintf.exe
C:\Users\user\Desktop\Lenguaje C\Ejemplos>
```

fprintf: Bloc de notas

Archivo Edición Formato Ver Ayuda

Escribir cadena en archivo mediante fprintf.
Facultad de Ingenieria.
UNAM

Funciones fread y fwrite

Código (fread)

```
1 #include <stdio.h>
2 /*
3  Este programa muestra el contenido de un archivo de texto. El
4  nombre del archivo se recibe como argumento de la
5  función principal.
6  */
7 int main(int argc, char **argv)
8 {
9     FILE *ap;
10    unsigned char buffer[2048]; // Buffer de 2 Kbytes
11    int bytesLeidos;
12
13    // Si no se ejecuta el programa correctamente
14    if(argc < 2)
15    {
16        printf("Ejecutar el programa de la siguiente manera:\n\tnombre _tprograma nombre_archivo\n");
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    {
24        printf("El archivo %s no existe o no se puede abrir", argv[1]);
25        return 1;
26    }
27
28    while(bytesLeidos = fread(buffer, 1, 2048, ap))
29        printf("%s", buffer);
30
31    fclose(ap);
32    return 0;
33 }
```



```

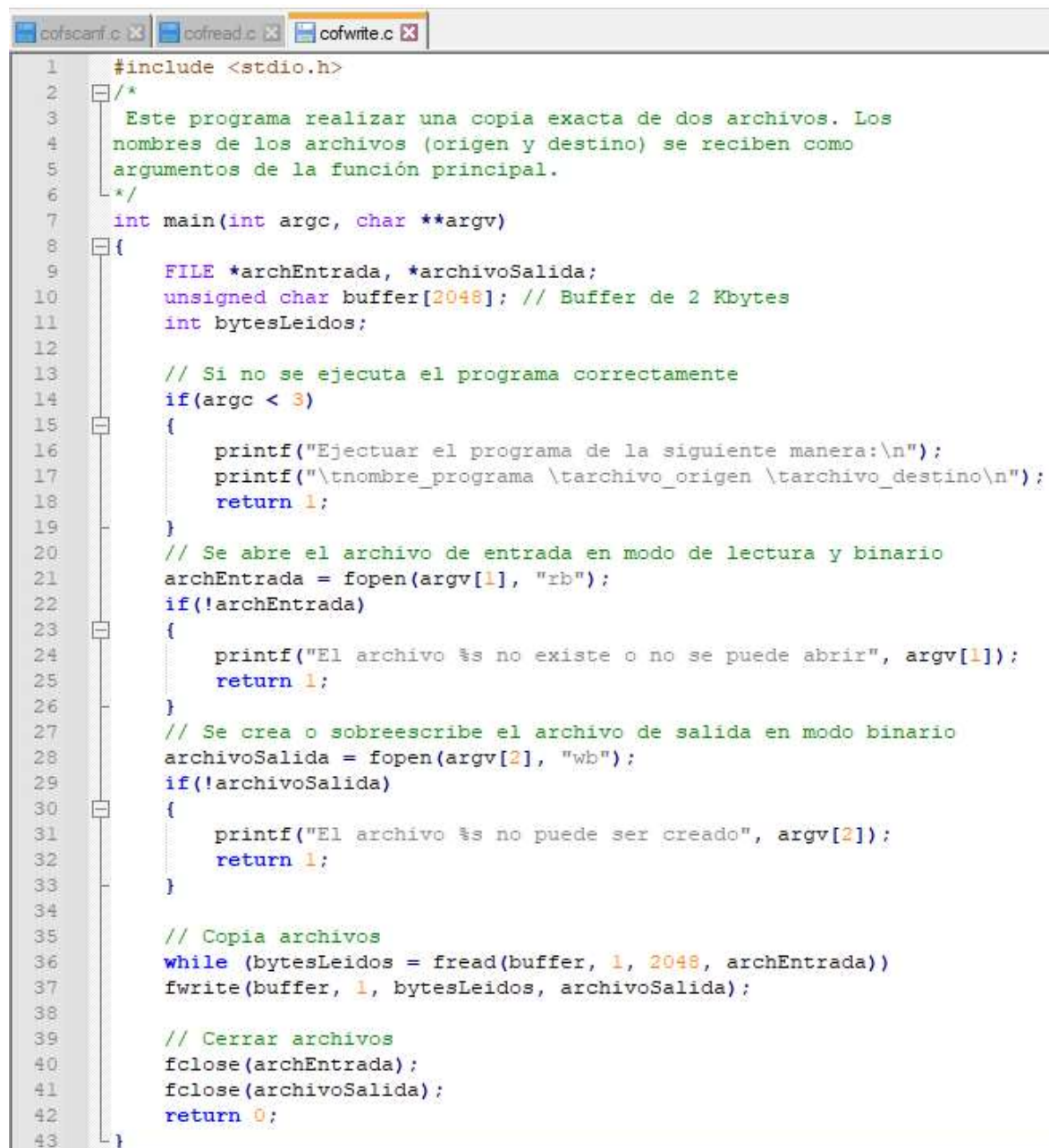
C:\Users\user\Desktop\Lenguaje C\Ejemplos>gcc cofread.c -o cofread.exe

C:\Users\user\Desktop\Lenguaje C\Ejemplos>cofread.exe
Ejecutar el programa de la siguiente manera:
    nombre_programa nombre_archivo

C:\Users\user\Desktop\Lenguaje C\Ejemplos>
C:\Users\user\Desktop\Lenguaje C\Ejemplos>cofread.exe fscanf.txt
Tarea de programacion
C:\Users\user\Desktop\Lenguaje C\Ejemplos>cofread.exe emiliano.txt
El archivo emiliano.txt no existe o no se puede abrir
C:\Users\user\Desktop\Lenguaje C\Ejemplos>

```

Código (fwrite)

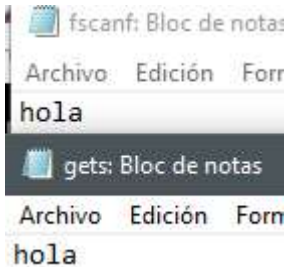


```

1  #include <stdio.h>
2  /*
3   Este programa realizar una copia exacta de dos archivos. Los
4   nombres de los archivos (origen y destino) se reciben como
5   argumentos de la función principal.
6  */
7  int main(int argc, char **argv)
8  {
9      FILE *archEntrada, *archivoSalida;
10     unsigned char buffer[2048]; // Buffer de 2 Kbytes
11     int bytesLeidos;
12
13     // Si no se ejecuta el programa correctamente
14     if(argc < 3)
15     {
16         printf("Ejecutar el programa de la siguiente manera:\n");
17         printf("\tnombre_programa \tarchivo_origen \tarchivo_destino\n");
18         return 1;
19     }
20     // Se abre el archivo de entrada en modo de lectura y binario
21     archEntrada = fopen(argv[1], "rb");
22     if(!archEntrada)
23     {
24         printf("El archivo %s no existe o no se puede abrir", argv[1]);
25         return 1;
26     }
27     // Se crea o sobrescribe el archivo de salida en modo binario
28     archivoSalida = fopen(argv[2], "wb");
29     if(!archivoSalida)
30     {
31         printf("El archivo %s no puede ser creado", argv[2]);
32         return 1;
33     }
34
35     // Copia archivos
36     while (bytesLeidos = fread(buffer, 1, 2048, archEntrada))
37         fwrite(buffer, 1, bytesLeidos, archivoSalida);
38
39     // Cerrar archivos
40     fclose(archEntrada);
41     fclose(archivoSalida);
42     return 0;
43 }

```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>gcc cofwrite.c -o cofwrite.exe  
C:\Users\user\Desktop\Lenguaje C\Ejemplos>cofwrite.exe gets.txt fscanf.txt  
C:\Users\user\Desktop\Lenguaje C\Ejemplos>
```



Conclusiones:

En conclusión, aprendimos que este tipo de códigos se ocupan para leer en nuestro código archivos aparte que se integran en la ejecución del código, como lo podría ser los archivos txt que ocupamos anteriormente. Además también aprendimos a identificar que todos estos son elementos de almacenamiento, y que estos, mediante el lenguaje C se pueden manipular su entrada y salida de datos para controlar el uso y lo que se haga con el mediante el código.