



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Karina García Morales

Asignatura: Fundamentos de programación

Grupo: 20

No. de práctica(s): 12

Integrante(s): Santiago Maya Luis Angel

No. de lista o brigada: 45

Semestre: 2023-1

Fecha de entrega: 10-Enero-23

Observaciones:

CALIFICACIÓN: _____

Objetivo:

El alumno elaborará 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.

Desarrollo:

En esta práctica llevamos más allá nuestros conocimientos adquiridos y vamos ahora a prender a hacer el manejo de archivos, que como su nombre lo indica, realizaremos programas que nos ayuden a abrir archivos de texto, a imprimir su contenido o a copiar su contenido de unos a otros.

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.

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. La firma de esta función es:

```
int fclose (FILE *apArch);
```

Así llegamos a realizar nuestro primer programa de la práctica en donde el programa nos permite abrir un archivo de texto y ver su contenido.

```
6 // Copyright © 2022 Santiago Maya Luis Angel. All rights reserved.
7 //
8
9 #include<stdio.h>
10 int main() {
11 FILE *archivo1;
12 archivo1 = fopen("/Users/fp20alu45/Documents/archivo1.txt", "w");
13 if (archivo1 != NULL) {
14 printf("El archivo se abrió correctamente.\n"); int res = fclose(archivo1);
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 return 0; }
22
```

```
El archivo se abrió correctamente.
fclose = 0
Program ended with exit code: 0
```



Como lo podemos ver en el código y en la ejecución, nuestro programa abrió el archivo correctamente y así mismo lo cerró, también podemos ver que nuestro archivo de texto tiene que estar previamente creado.

Funciones fgets y fputs

Las funciones fgets() y fputs() pueden leer y escribir, respectivamente, cadenas sobre los archivos. Las firmas de estas funciones son, respectivamente:

```
char *fgets(char *buffer, int tamaño, FILE *apArch);
```

```
char *fputs(char *buffer, FILE *apArch);
```

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. Ahora tenemos el segundo programa en el que este programa permite leer el contenido de un archivo, de ser posible, a través de la función fopen(), usamos esta función ya que es la que usamos en el programa que ocupamos.

```
6 // Copyright © 2022 Santiago Maya Luis Angel. All rights reserved.
7 //
8
9 #include<stdio.h>
10 int main() {
11     FILE *archivo;
12     char caracteres[50];
13     archivo = fopen("/Users/fp20alu45/Documents/puts.txt", "r");
14     if (archivo != NULL) {
15         printf("El archivo se abrió correctamente."); printf("\nContenido del archivo:\n");
16         while (feof(archivo) == 0)
17         {
18             fgets (caracteres, 50, archivo); //Lectura de contenido de un texto plano
19             printf("%s", caracteres); }
20         fclose(archivo); }
21     return 0; }
22
```

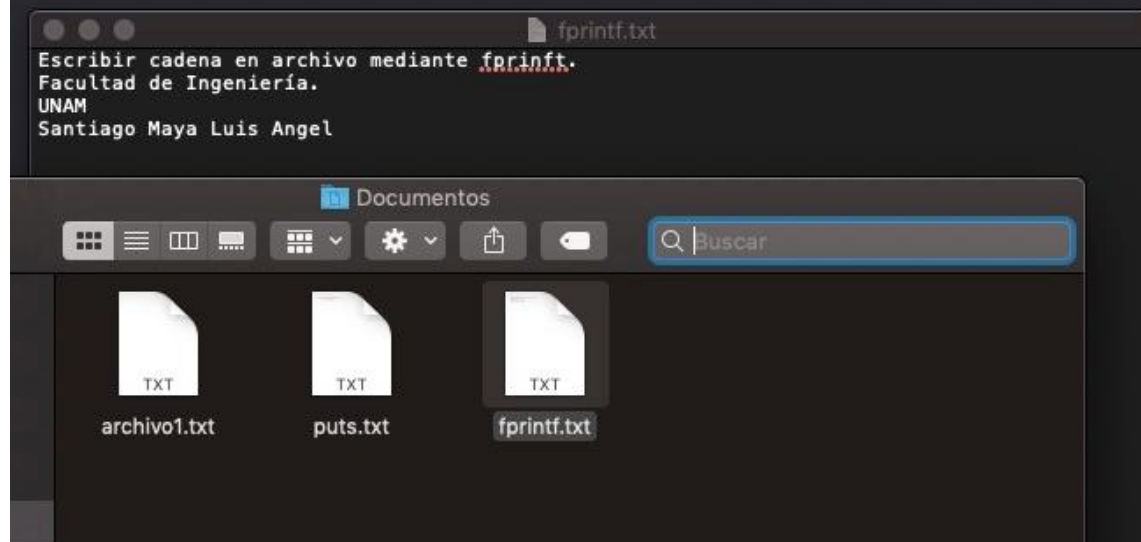


```
El archivo se abrió correctamente.
Contenido del archivo:
Escribir cadena en archivo mediante fputs.
    Facultad de Ingeniería.
    Facultad de Ingeniería.
Program ended with exit code: 0
```

Ahora como lo vemos en la ejecución nuestro programa abrió y mostro el contenido de nuestro archivo de texto, igualmente como en el programa anterior, nuestro archivo de texto tiene que estar previamente creado y con el contenido que queremos mostrar.

Este programa permite escribir una cadena dentro de un archivo, de ser posible, a través de la función `fprint`.

```
2 // main.c
3 // program
4 //
5 // Created by Santiago Maya Luis Angel on 12/14/22.
6 // Copyright © 2022 Santiago Maya Luis Angel. All rights reserved.
7 //
8
9 #include<stdio.h>
10 int main() {
11     FILE *archivo;
12     char escribir[] = "Escribir cadena en archivo mediante fprintf. \nFacultad de Ingeniería.\n";
13     char escribirNombre[] = "Santiago Maya Luis Angel \n";
14     archivo = fopen("/Users/fp20alu45/Documents/fprintf.txt", "w+");
15     if (archivo != NULL)
16     {
17         fprintf(archivo, escribir);
18         fprintf(archivo, "%s", "UNAM\n");
19         fprintf(archivo, escribirNombre);
20         fclose(archivo);
21     }
22     else
23     {
24         printf("El archivo no existe o no se tiene permisos de lectura/escritura.\n");
25     }
26     return 0; }
27
```



Como podemos ver, en la ejecución y en nuestros archivos de texto, este programa escribió en un archivo de texto que previamente también tenía que estar creado, y en el programa escribimos lo que queremos que el programa escriba en el archivo de texto.

Funciones fscanf y fprintf

Las funciones fprintf() y fscanf() se comportan similarmente a printf() (imprimir) y scanf() (leer), con la diferencia de que operan sobre un archivo. Sus estructuras son:

```
int fprintf(FILE *apArch, char *formato, ...);
```

```
int fscanf(FILE *apArch, char *formato, ...);
```

Donde apArch es un apuntador al archivo devuelto por una llamada a la función fopen(), es decir, fprintf() y fscanf() dirigen sus operaciones de E/S al archivo al que apunta apArch. formato es una cadena que puede incluir texto o especificadores de impresión de variables. En los puntos suspensivos se agregan las variables (si es que existen) cuyos valores se quieren escribir en el archivo.

consecuentemente tenemos el programa 4 en donde este programa permite leer el contenido de un archivo, de ser posible, a través de la función fscanf.

```
9  #include<stdio.h>
10 int main() {
11     FILE *archivo;
12     char caracteres[50];
13     archivo = fopen("/Users/fp20alu45/Documents/fprintf.txt", "r");
14     if (archivo != NULL)
15     {
16         while (feof(archivo)==0) {
17             fscanf(archivo, "%s", caracteres);
18             printf("%s\n", caracteres); }
19         fclose(archivo); }
20     else
21     {
22         printf("El archivo no existe.\n");
23     }
24     return 0;
25 }
26
27
```



```
Escribir
cadena
en
archivo
mediante
fprintf.
Facultad
de
Ingeniería.
UNAM
Santiago
Maya
Luis
Angel
Angel
```

Como vemos este programa abre un archivo de texto e imprime su contenido.

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. Su sintaxis es la siguiente:

```
int fread(void *ap, size_t tam, size_t nelem, FILE *archivo)
```

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. Su sintaxis es la siguiente:

```
int fwrite(void *ap, size_t tam, size_t nelem, FILE *archivo)
```

Ahora tenemos nuestro programa 6, este programa muestra el contenido de un archivo de texto. El nombre del archivo se recibe como argumento de la función principal.

```
1  #include <stdio.h>
2  int main(int argc, char **argv)
3  {
4      FILE *ap;
5      unsigned char buffer[2048]; // Buffer de 2 Kbytes
6      int bytesLeidos;
7      // Si no se ejecuta el programa correctamente
8      if(argc < 2)
9      {
10         printf("Ejecutar el programa de la siguiente manera: \n\tnombre_\tprograma nombre_archivo\n");
11         return 1;
12     }
13     // Se abre el archivo de entrada en modo lectura y binario
14     ap = fopen(argv[1], "rb");
15     if(!ap)
16     {
17         printf("El archivo %s no existe o no se puede abrir", argv[1]);
18         return 1;
19     }
20     while(bytesLeidos = fread(buffer, 1, 2048, ap))
21     {
22         printf("%s", buffer);
23     }
24     fclose(ap);
25     return 0;
26 }
```

C:\Users\luisr\Documents\prog612.exe

Hola

voy a sacar 10 :)H

Process exited after 0.03118 seconds with return value 0

Presione una tecla para continuar . . .




Este programa nos muestra también lo que ya hay en un archivo de texto.


Entonces finalizamos la practica con nuestro programa , 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.

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

C:\Users\luisr\Documents\prog712.exe

Process exited after 0.02519 seconds with return value 0
Presione una tecla para continuar . . .

 archivo12	10/01/2023 09:52 p. m.	Documento de texto
 nuevo	10/01/2023 11:31 a. m.	Documento de texto
 prog7prac	10/01/2023 11:31 a. m.	C++ Source File

 *nuevo: Bloc de notas

Archivo Edición Formato Ver Ayuda

hola

voy a sacar 10 :)

Así como lo decía al principio del programa, este copia el contenido de un archivo de texto y crea uno nuevo con el contenido de otro.

Conclusiones

Esta práctica estuvo sencilla y no fue tan compleja de entender, como podemos ver en la práctica, varios de los programas realizaban la misma función, solo que cambiaba la sintaxis de estos y las funciones que ocupaban, aun así nos dan varios ejemplos y podemos utilizar la que nosotros comprendamos de mejor manera, y si se obtuvieron los aprendizajes esperados, con esta práctica concluimos, en verdad yo aprendí mucho en el curso con estas prácticas y me ayudaran mucho para mi desarrollo dentro de la carrera .

Referencias

- El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.
- [Laboratorio Salas A y B \(unam.mx\)](https://laboratorio.salas.unam.mx/)