



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I. Marco Antonio Martínez Quintana

Asignatura: Fundamentos de Programación

Grupo: 3 Bloque: 136

No de Práctica(s): 13: Lectura y escritura de datos

Integrante(s): Carranza Ochoa José David

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

No. de Lista o Brigada: 6

Semestre: 2021-1

Fecha de entrega: 25/01/2021

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.

Actividades:

- A través de programas en C, emplear las funciones para crear, leer, escribir y sobrescribir archivos de texto plano.
- Manipular archivos empleando los diferentes tipos de acceso a ellos.

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 es una forma de interacción entre la entrada y salida de datos, dicho apuntador define las características de un archivo, así como nombre, estado y posición actual del archivo su sintaxis es la siguiente:

`FILE *F;`

Abrir archivo

Para abrir algún archivo se utiliza “fopen” cuya sintaxis es la siguiente:

`*FILE fopen(char *nombre_archivo, char *modo);`

Donde se tiene que tener en cuenta los modos de operación, siendo aquellos:

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

Para cancelar la apertura de archivos realizada con “fopen”, se suele emplear “fclose” cuya operación es la opuesta, se debe tomar en cuenta la trascendencia de la orden ya que puede generar daños al archivo, así como la pérdida de datos:

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

Ejemplo: Apertura y cierre de archivos

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

D:\USUARIO\DOCUMENTOS_MECANICO\CompartidaU-W\trece.exe

```
Error al abrir el archivo.
El archivo no existe o no se tienen permisos de lectura.
```

```
-----
Process exited after 0.02676 seconds with return value 0
Presione una tecla para continuar . . .
```

Funciones fgets y fputs

Las funciones “fgets()” y “fputs()” permiten la lectura y escritura de información en algún archivo, tomando en consideración la operación por renglones, donde:

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

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

Ejemplo: Codificación de lectura de un archivo

```
1  #include<stdio.h>
2  /*
3   * Este programa permite Lee 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 }
```

D:\USUARIO\DOCUMENTOS_MECANICO\CompartidaU-W\trece.exe

```
-----
Process exited after 0.02779 seconds with return value 0
Presione una tecla para continuar . . .
```

Ejemplo: Escritura en un archivo definido

```
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.\n";
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 }
```

D:\USUARIO\DOCUMENTOS_MECANICO\CompartidaU-W\trece.exe

```
Error al abrir el archivo.
El archivo no existe o no se tienen permisos de lectura.
-----
Process exited after 0.02747 seconds with return value 0
Presione una tecla para continuar . . .
```

Funciones fscanf y fprintf

Ambas funciones se comportan como “printf” y “scanf” solo que en vez de operar con bloques de código trabajan con archivos teniendo la siguiente sintaxis:

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

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

Ejemplo: lectura del contenido de un archivo

```
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 }
```

D:\USUARIO\DOCUMENTOS_MECANICO\CompartidaU-W\trece.exe

El archivo no existe.

Process exited after 0.02628 seconds with return value 0

Presione una tecla para continuar . . .

Ejemplo: Escritura en el contenido de un archivo

```
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      FILE *archivo;
8      char escribir[] = "Escribir cadena en archivo mediante fprintf. \nFacultad de Ingeniería.\n";
9      archivo = fopen("fprintf.txt", "r+");
10     if (archivo != NULL) {
11         fprintf(archivo, escribir);
12         fprintf(archivo, "%s", "UNAM\n");
13         fclose(archivo);
14     } else {
15         printf("El archivo no existe o no se tiene permisos de lectura /escritura.\n");
16     }
17     return 0;
18 }
```

D:\USUARIO\DOCUMENTOS_MECANICO\CompartidaU-W\trece.exe

El archivo no existe o no se tiene permisos de lectura /escritura.

Process exited after 0.02831 seconds with return value 0

Presione una tecla para continuar . . .

Funciones fread y fwrite

Ambas funciones trabajan con elementos de la misma longitud, permitiendo leer uno o diferentes elementos de la misma longitud por medio de un apuntador.

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

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

Ejemplo: Código donde se expone el contenido del archivo (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     FILE *ap;
9     unsigned char buffer[2048]; // Buffer de 2 Kbytes
10    int bytesLeidos;
11
12    // Si no se ejecuta el programa correctamente
13    if(argc < 2) {
14        printf("Ejecutar el programa de la siguiente manera:\n\tnombre_programa nombre_archivo\n");
15        return 1;
16    }
17
18    // Se abre el archivo de entrada en modo Lectura y binario
19    ap = fopen(argv[1], "rb");
20    if(!ap) {
21        printf("El archivo %s no existe o no se puede abrir", argv[1]);
22        return 1;
23    }
24
25    while(bytesLeidos = fread(buffer, 1, 2048, ap))
26        printf("%s", buffer);
27
28    fclose(ap);
29    return 0;
30 }
```

D:\USUARIO\DOCUMENTOS_MECANICO\CompartidaU-W\trece.exe

Ejecutar el programa de la siguiente manera:
nombre_programa nombre_archivo

Process exited after 0.02668 seconds with return value 1
Presione una tecla para continuar . . .

Ejemplo: Código donde se realiza una copia exacta del documento

```
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     FILE *archEntrada, *archivoSalida;
9     unsigned char buffer[2048]; // Buffer de 2 Kbytes
10    int bytesLeidos;
11
12    // Si no se ejecuta el programa correctamente
13    if(argc < 3) {
14        printf("Ejecutar el programa de la siguiente manera:\n");
15        printf("\tnombre_programa \tarchivo_origen \tarchivo_destino\n");
16        return 1;
17    }
18
19    // Se abre el archivo de entrada en modo de Lectura y binario
20    archEntrada = fopen(argv[1], "rb");
21    if(!archEntrada) {
22        printf("El archivo %s no existe o no se puede abrir", argv[1]);
23        return 1;
24    }
25
26    // Se crea o sobrescribe el archivo de salida en modo binario
27    archivoSalida = fopen(argv[2], "wb");
28    if(!archivoSalida) {
29        printf("El archivo %s no puede ser creado", argv[2]);
30        return 1;
31    }
32
33    // Copia archivos
34    while(bytesLeidos = fread(buffer, 1, 2048, archEntrada))
35        fwrite(buffer, 1, bytesLeidos, archivoSalida);
36
37    // Cerrar archivos
38    fclose(archEntrada);
39    fclose(archivoSalida);
40    return 0;
41 }
```

D:\USUARIO\DOCUMENTOS_MECANICO\CompartidaU-W\trece.exe

Ejecutar el programa de la siguiente manera:
nombre_programa archivo_origen archivo_destino

Process exited after 0.02642 seconds with return value 1
Presione una tecla para continuar . . .

Conclusión

El uso de herramientas que solucionen problemas como la escritura, lectura y la edición de textos en archivos, facilita y crea nuevas alternativas donde se llegan a nuevas posibilidades donde hasta ahora no conocía. Ahora la manipulación de archivos, así como su contenido resultó agradable en cuanto al desarrollo de la práctica.

Conforme avanzaba en la práctica, tuve ligeros problemas al momento de generar los archivos de texto con el fin de analizar su contenido, es por ellos que decidí investigar los factores generadores encontrando soluciones que en lo personal no me favorecieron; sin embargo, pude ahondar más en la edición de textos llegando a visualizar por medio de vídeos el cómo debería obtener mi resultado, es por ello que añadí las resultantes “archivo no encontrado”

Bibliografía

El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.