



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor:

Ingeniera Claudia Rodríguez Espino

Asignatura:

Fundamentos de Programación

Grupo:

1102

No de Práctica(s):

Práctica número 13

Integrante(s):

Chaveste Bermejo Carlos Alberto

Semestre:

2018-1

Fecha de entrega:

17/11/2017

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.

DESARROLLO:

Lectura y escritura de datos.-

En la práctica en cuestión, aprendimos sobre la lectura, creación, sobre-escritura y manipulación de datos dentro de archivos de texto plano, a través de lenguaje C, primero, se nos explica que es un archivo, que es, en esencia un conjunto de instrucciones y atributos que funcionan bajo un contexto, podemos usarlos mediante apuntadores archivos, cuya sintaxis es:

FILE *F;

Y existen comandos para el manejo de archivos a través de lenguaje c, como lo son:

Para abrir archivos.-

Comando	Función
r	Abrir 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.

Los comandos “r” sirven para abrir archivos, mientras que los comandos “w”, sirven para crear archivos; y por último, los comandos “a”, se usan para añadir.

Estos modos de abrir un programa se introducen en el comando:

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

En donde modo es la forma en la que se abrirá el archivo de texto plano en cuestión.

Cerrar archivos.-

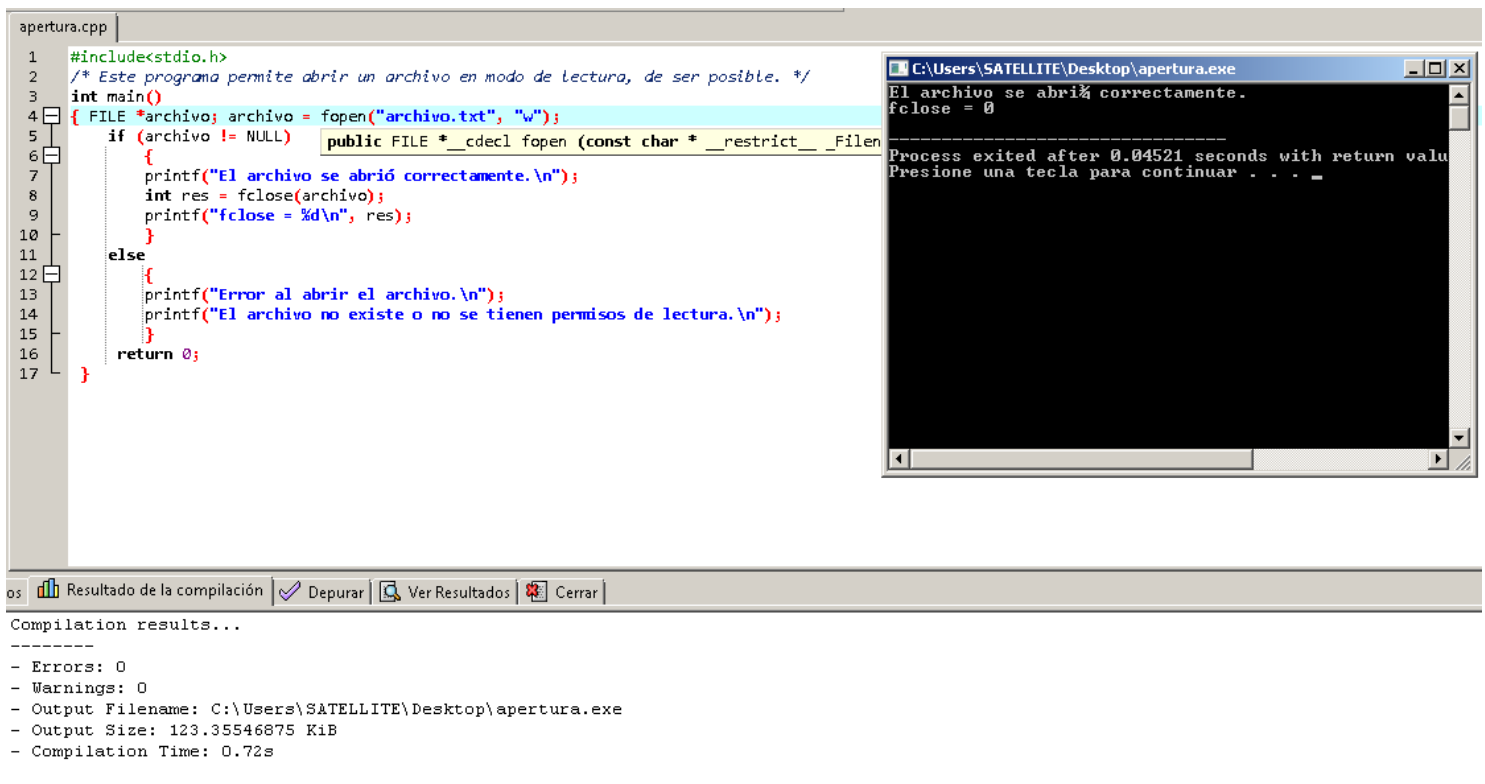
Para evitar la pérdida de datos, o la destrucción del archivo en su totalidad, el lenguaje c cuenta con el comando fclose, que sirve para darle un cierre formal al archivo, y de esta forma, evitar la pérdida de información valiosa. Su sintaxis es la siguiente:

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

En donde ***apArch** es el apuntador al archivo que se abrió con la función fopen.

ACTIVIDAD EN CLASE

Programa 1.Apertura de archivo.



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

```
C:\Users\SATELLITE\Desktop\apertura.exe
El archivo se abrió correctamente.
fclose = 0
-----
Process exited after 0.04521 seconds with return value 0
Presione una tecla para continuar . . . _
```

os Resultado de la compilación Depurar Ver Resultados Cerrar

Compilation results...

```
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\SATELLITE\Desktop\apertura.exe
- Output Size: 123.35546875 KiB
- Compilation Time: 0.72s
```

Programa 2.Fgets.

The screenshot shows a C++ IDE with two windows. The left window displays the source code for `fgets.cpp`, which reads the content of `archivo.txt` line by line using `fgetc` and prints it. The right window shows the program's output, indicating successful file opening and the content of the file: `FI UNAM`. Below the output window, a Notepad application is open, showing the same text: `FI UNAM`.

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

El archivo se abrió correctamente.
Contenido del archivo:
FI UNAM

Process exited after 0.05956 seconds with return value 0
Presione una tecla para continuar . . .

archivo: Bloc de notas

Archivo Edición Formato Ver Ayuda

FI UNAM

Programa 3. Fputs.

The screenshot shows a C++ IDE with two windows. The left window displays the source code for `fputs.cpp`, which writes the string `"Escribir cadena en archivo mediante fputs."` to `archivo.txt` using `fputs`. The right window shows the program's output, indicating successful file opening and the content of the file: `Escribir cadena en archivo mediante fputs. Facultadde Ingeniería.`. Below the output window, a Notepad application is open, showing the same text: `Escribir cadena en archivo mediante fputs. Facultadde Ingeniería.`

```
1 #include<stdio.h>
2 /*
3  Este programa permite escribir una cadena dentro de un archivo, de
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\t";
9     archivo = fopen("archivo.txt", "r+");
10    if (archivo != NULL) {
11        printf("El archivo se abrió correctamente.\n");
12        fputs (escribir, archivo);
13        fclose(archivo);
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 }
```

El archivo se abrió correctamente.

Process exited after 0.1249 seconds with return value 0
Presione una tecla para continuar . . .

archivo: Bloc de notas

Archivo Edición Formato Ver Ayuda

Escribir cadena en archivo mediante fputs.
Facultadde Ingeniería.

Programa 4.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     FILE *archivo;
8     char caracteres[50];
9     archivo = fopen("archivo.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 }
```

Process exited after 0.03665 seconds with return value 0
Presione una tecla para continuar . . .

archivo: Bloc de notas

Archivo Edición Formato Ver Ayuda

Escribir cadena en archivo mediante fputs.
Facultadde Ingenieria.
Ingenierya.

Programa 5.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     FILE *archivo;
8     char escribir[] = "Escribir cadena en archivo mediante fprintf. \nFacu";
9     archivo = fopen("archivo.txt", "r+");
10    if (archivo != NULL) {
11        fprintf(archivo, escribir);
12        fprintf(archivo, "%s", "UNAM\n");
13        fclose(archivo);
14    } else {
15        {
16            printf("El archivo no existe o no se tiene permisos de lectura /escrit");
17        }
18    }
19    return 0;
20 }
```

Process exited after 0.03627 seconds with return value 0
Presione una tecla para continuar . . .

archivo: Bloc de notas

Archivo Edición Formato Ver Ayuda

Escribir cadena en archivo mediante fprintf.
Facultadde Ingenieria.
UNAM

CONCLUSIÓN

Los comandos de apertura y escritura, pueden resultar bastante útiles para el manejo de información en cuanto a lenguaje c se refiere, es más fácil hacerlo mediante una interfaz gráfica, pero si se quiere manipular la información a nivel de lenguaje, es mejor hacerlo de esta manera. Es importante usar la función fclose para evitar la pérdida de información.

BIBLIOGRAFÍA

<http://lcp02.fi-b.unam.mx/#>