



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Ing. Claudia Rodriguez Espino

Asignatura: Fundamentos de Programación

Grupo: 1102

No de Práctica(s): Práctica número 13

Integrante(s): Alejandro Nava Cruz

Semestre: 2018-1

Fecha de entrega: 21 de Noviembre del 2017

Obervaciones:

CALIFICACIÓN: _____

Practica 13: Lectura y escritura de datos

Objetivo:

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

Desarrollo:

Un apuntador de archivo, al igual que otros apuntadores, funciona para llamar texto externo o especial, pero esta vez no solo de un programa simple, si no de archivos externos, algunos ya creados con anticipación y otros que se pueden crear al momento de ejecutar el programa. Se nos resume lo que es un archivo en sí, un conjunto de atributos que funcionan bajo un contexto, su sintaxis para usar archivos apuntadores es:

FILE *F

Existen diversos comandos para el manejo de archivos a través del lenguaje C, tales como:

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+: 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.

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

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

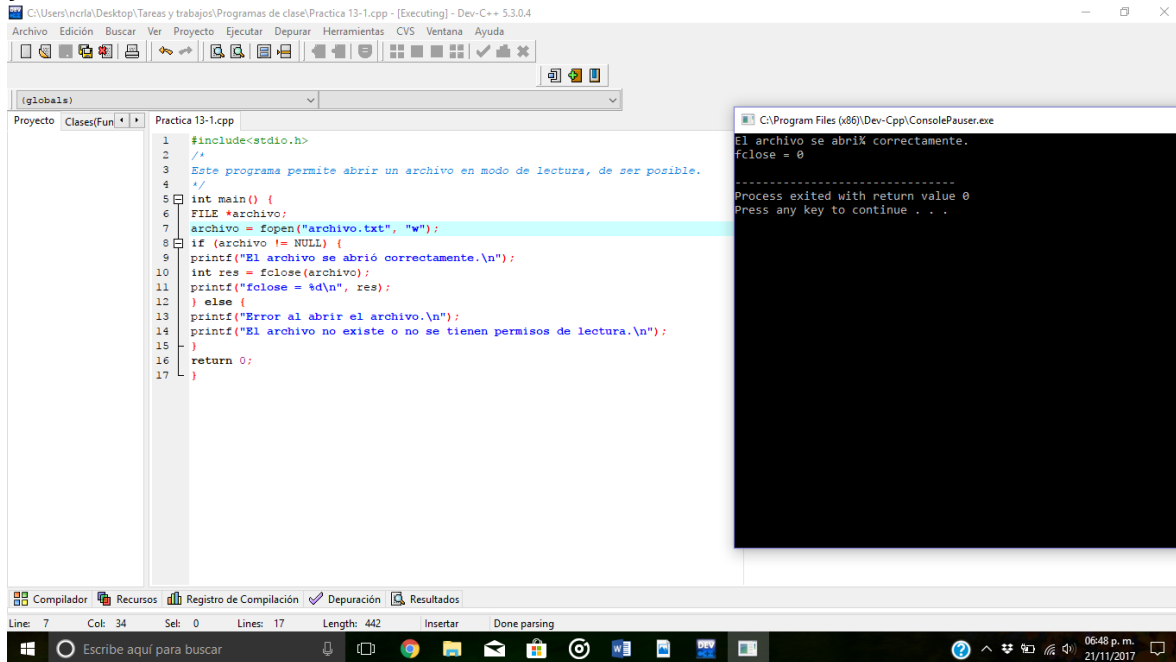
Para evitar la pérdida de datos, o la destrucción del archivo en su totalidad, el lenguaje c cuenta con el comando fclose, que sirve para darle un cierre formal al archivo:

int fclose(FILE *apArch);

Actividades hechas en casa

1. Apertura de un archivo con fopen

```
#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;
}
```



2. Uso de 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("archivo.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);
```

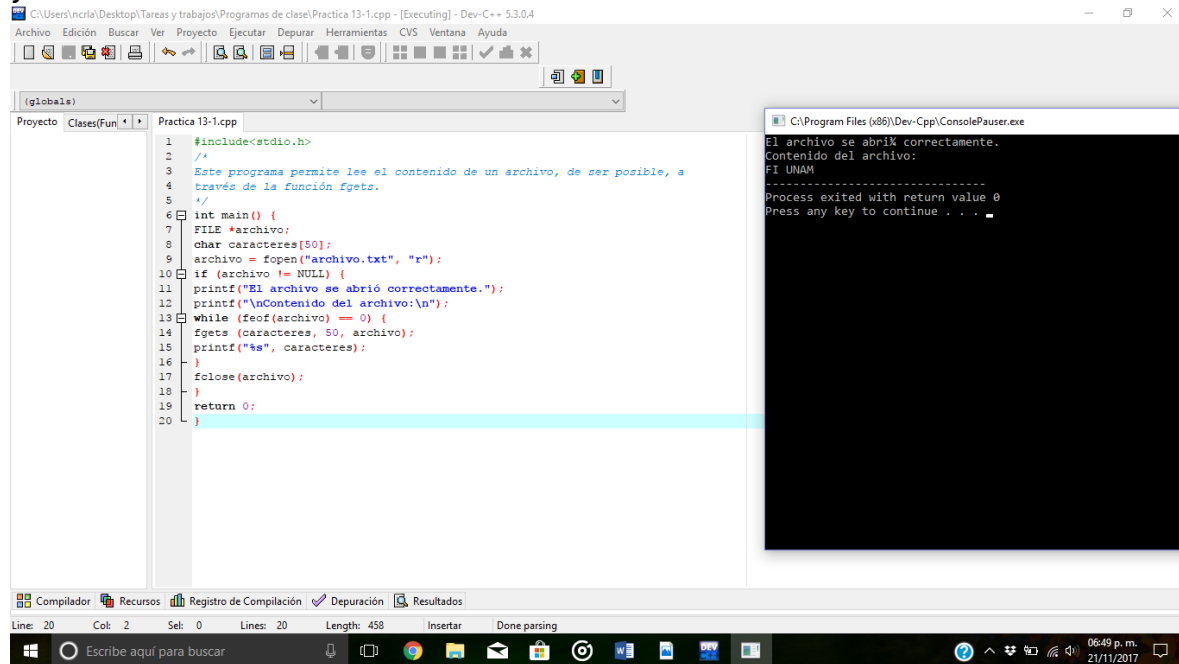
```
}
```

```
fclose(archivo);
```

```
}
```

```
return 0;
```

```
}
```



3. Uso de 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("archivo.txt", "r+");
```

```
if (archivo != NULL) {
```

```
printf("El archivo se abrió correctamente.\n");
```

```
fputs (escribir, archivo);
```

```
fclose(archivo);
```

```
} else {
```

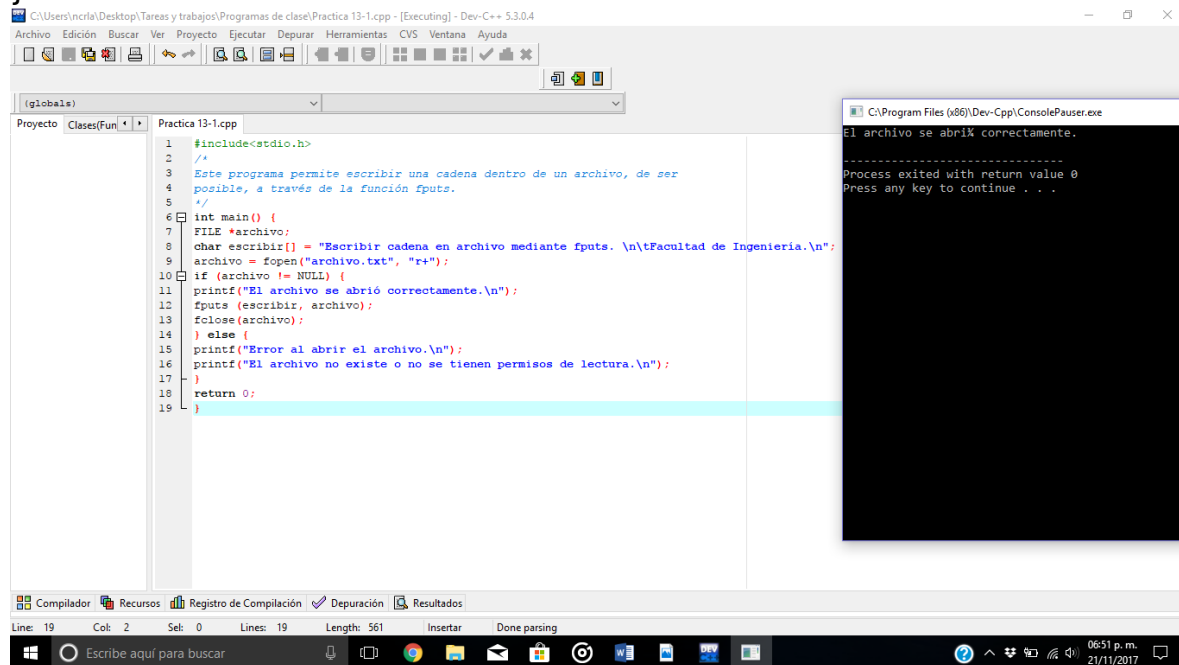
```
printf("Error al abrir el archivo.\n");
```

```
printf("El archivo no existe o no se tienen permisos de lectura.\n");
```

```
}
```

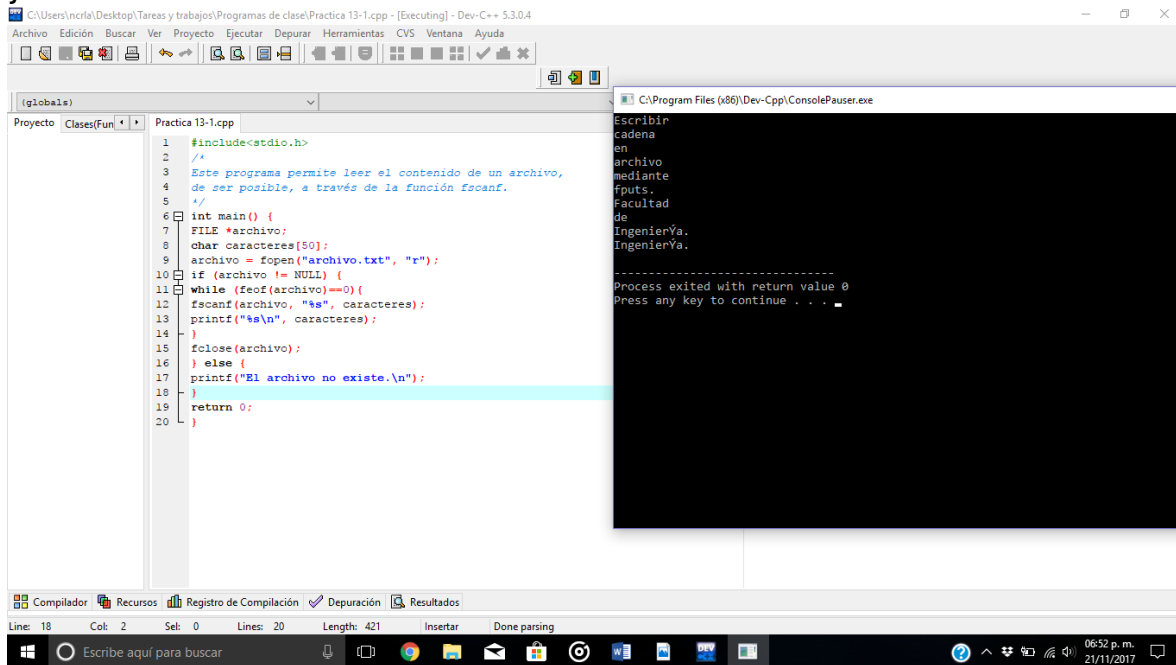
```
return 0;
```

```
}
```



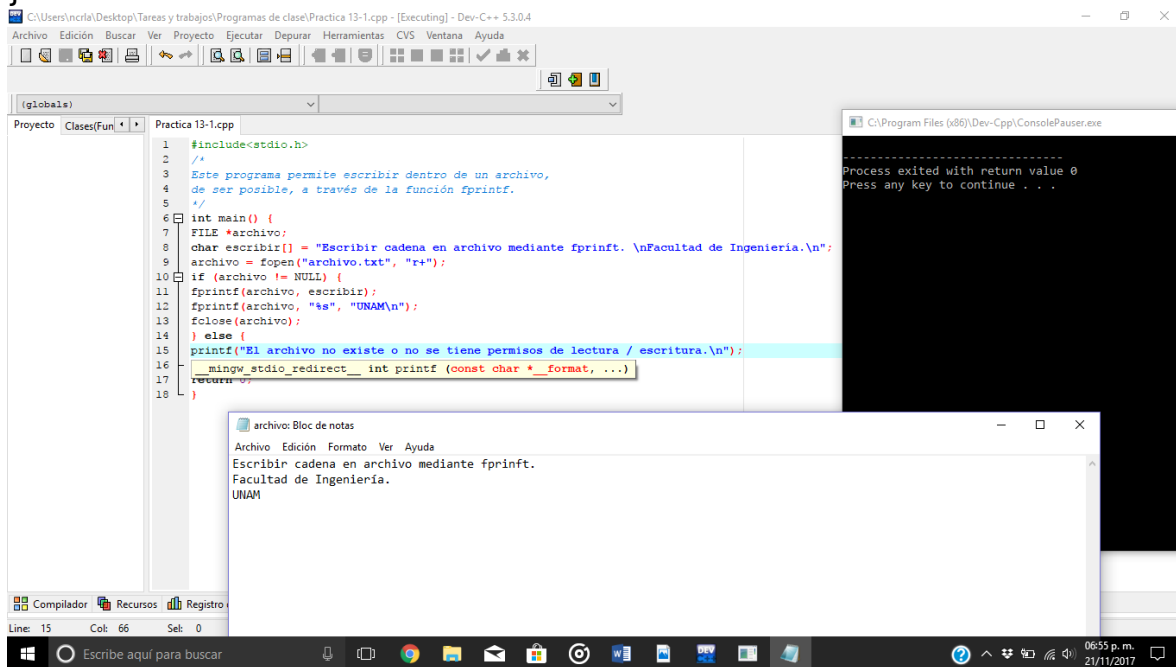
4. Uso de fscanf

```
#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;
}
```



5. Uso de 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;
}
```



Conclusiones:

Los comandos especiales para abrir y cerrar archivos en C quizás sean uno de los más útiles, ya que te permite crear, modificar o abrir archivos externos, de una manera “sencilla”, quizás no sea el mejor método para la realización de estas tareas, pero en cuanto a nivel de lenguaje se habla, esta es la mejor manera de hacerlo.

Bibliografía:

Práctica número 13, Facultad de Ingeniería, en Laboratorios A y B (2017), Sitio Web:
<http://lcp02.fib.unam.mx/>