	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: 22

No. de práctica(s): Práctica No.12

Integrante(s): Freddy Beckham Cedillo Arias

No. de lista o brigada: No.11

Semestre: 1er Semestre

Fecha de entrega: 6 de noviembre del 2024

Observaciones:

CALIFICACIÓN: _____

Practica 12

Lectura y escritura de datos

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:

Programa1.c

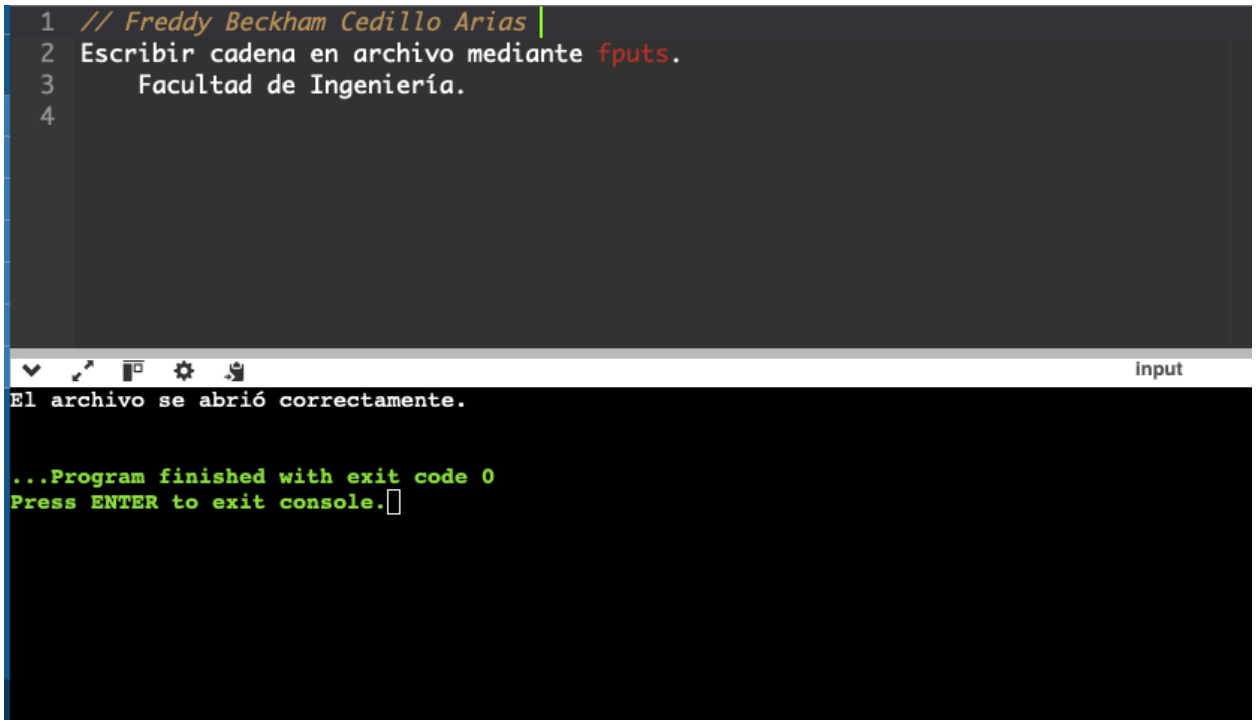
Se hizo el cambio de “r” por “w” para poder crear el archivo, además de utilizar fopen y fclose para crear el archivo y cerrar el archivo creado respectivamente.

```c
#include<stdio.h>
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;
}
```

```
input
El archivo se abrió correctamente.
fclose = 0


...Program finished with exit code 0
Press ENTER to exit console.
```

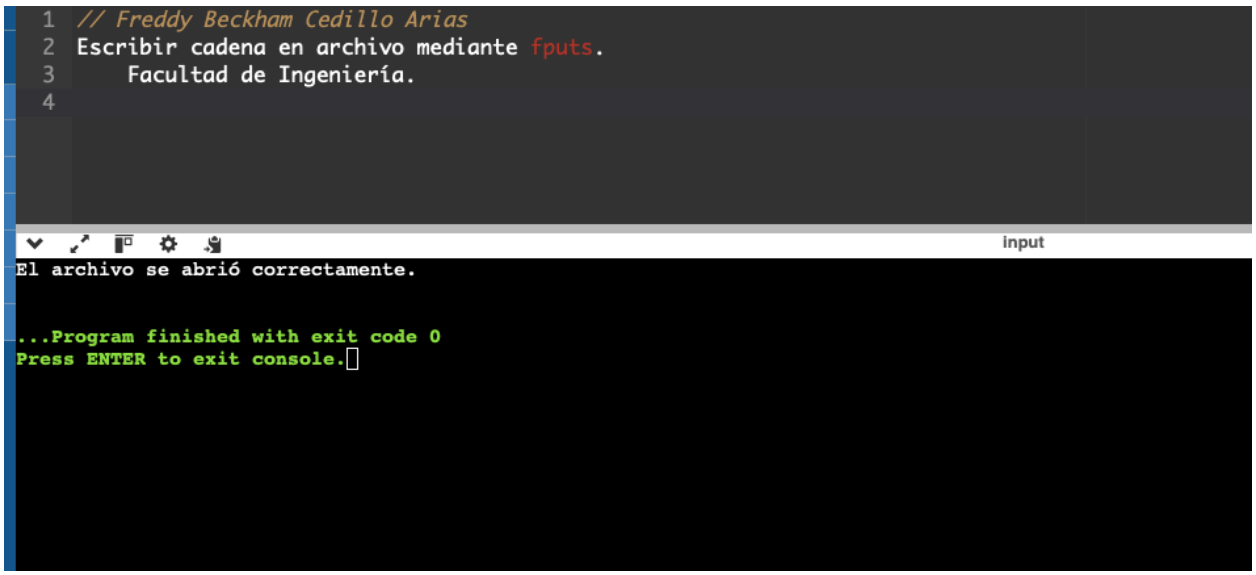
```
Last login: Wed Oct 30 19:34:23 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[Eslovaquia50:~ fp22alu11$ pwd
/Users/fp22alu11
[Eslovaquia50:~ fp22alu11$ cd Desktop/
[Eslovaquia50:Desktop fp22alu11$ pwd
/Users/fp22alu11/Desktop
Eslovaquia50:Desktop fp22alu11$
```

Programa 3.c
Se utilizaron las funciones fopen y fclose para abrir y cerrar el archivo. Además de que se utilizo fputs para poder escribir en el archivo.



Programa 2.c

Se emplearon fopen y fclose para crear un archivo y cerrarlo respectivamente y fgets para leerlo.



Programa 5.c
Se utilizaron fopen y fclose para abrir y cerrar el archivo, para después usar fprintf para imprimir en el archivo.

```
1  Escribir cadena en archivo mediante fprinft.
2  Facultad de Ingeniería.
3  UNAM
4  Freddy Beckham
5
```

input

```
main.c: In function 'main':
main.c:7:18: warning: format not a string literal and no format arguments [-Wformat-security]


...Program finished with exit code 0
Press ENTER to exit console.
```

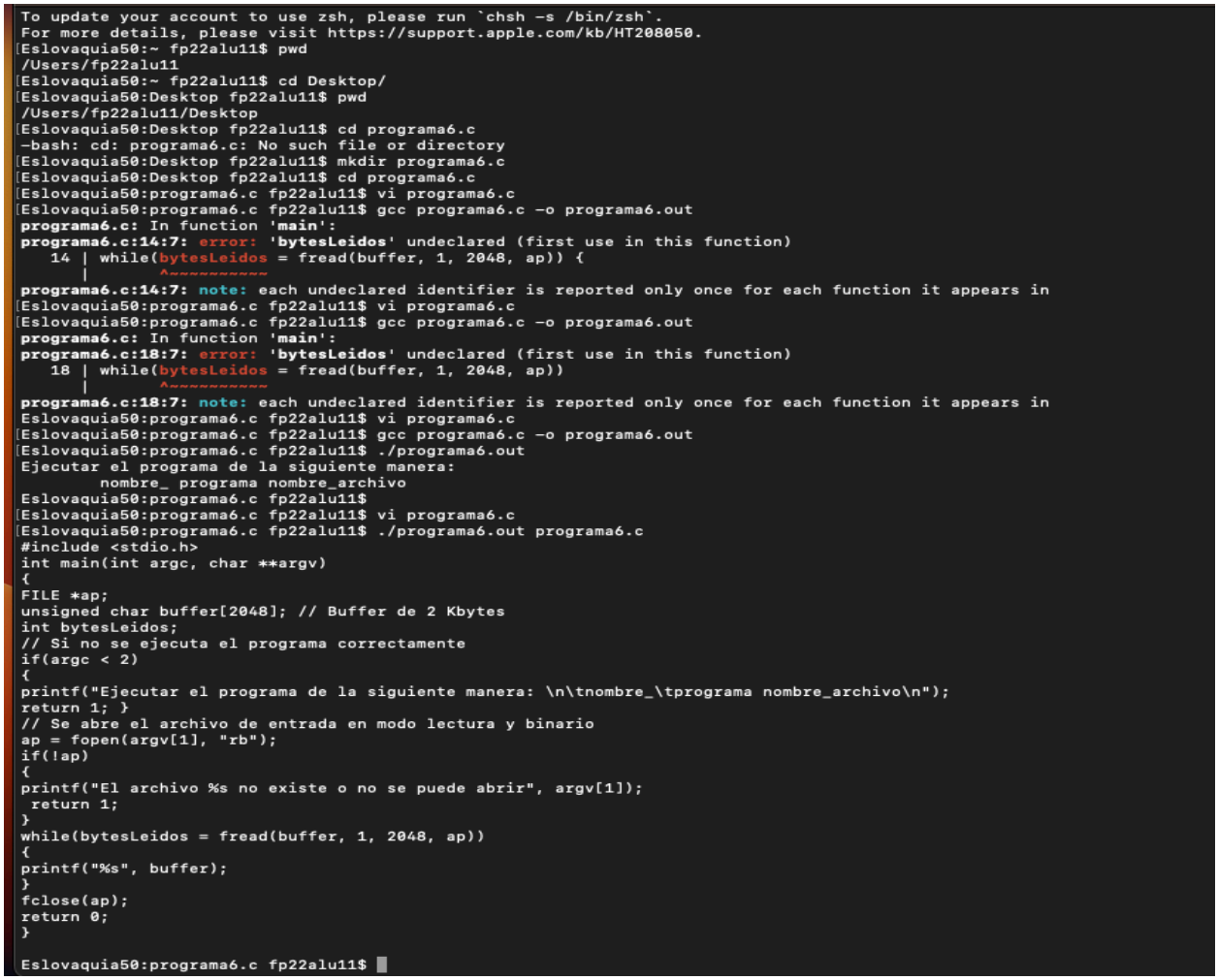
Programa 4.c
Se emplearon fopen y fclose para abrir y cerrar el archivo, y posteriormente fscanf para leerlo.

```
1  // Freddy Beckham Cedillo Arias
2  #include<stdio.h>
3  int main() {
4  FILE *archivo;
5  char caracteres[50];
6  archivo = fopen("fprintf.txt", "r");
7  if (archivo != NULL)
8  {
9  while (feof(archivo)==0) {
10 fscanf(archivo, "%s", caracteres);
11 printf("%s\n", caracteres); }
12 fclose(archivo); }
13 else
14 {
15 printf("El archivo no existe.\n");
16 }
17 return 0; }
```

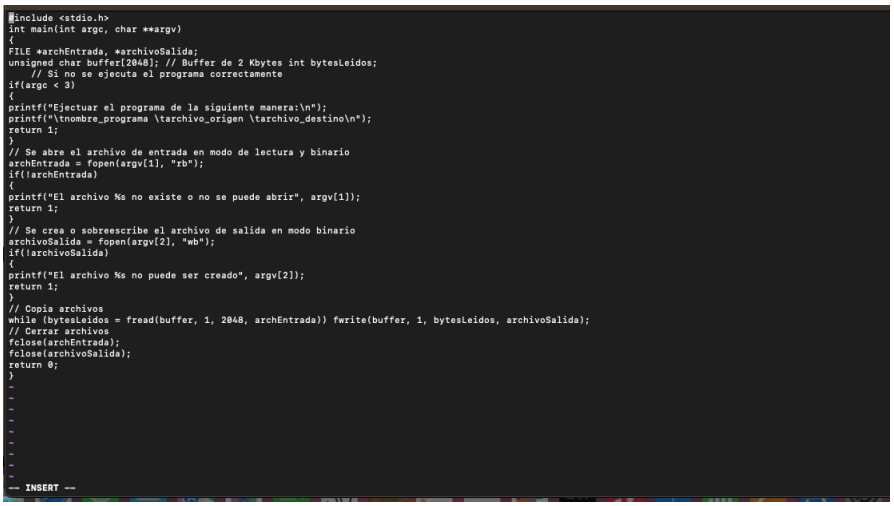
input

```
Escribir
cadena
en
archivo
mediante
fprinft.
Facultad
de
Ingeniería.
UNAM
"fprintf.txt"


...Program finished with exit code 0
Press ENTER to exit console.
```

Se ejecutaron los programas 6.c y 7.c en la terminal



```
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[Eslovaquia50:~ fp22alu11$ pwd
/Users/fp22alu11
[Eslovaquia50:~ fp22alu11$ cd Desktop/
[Eslovaquia50:Desktop fp22alu11$ pwd
/Users/fp22alu11/Desktop
[Eslovaquia50:Desktop fp22alu11$ cd programa6.c
-bash: cd: programa6.c: No such file or directory
[Eslovaquia50:Desktop fp22alu11$ mkdir programa6.c
[Eslovaquia50:Desktop fp22alu11$ cd programa6.c
[Eslovaquia50:programa6.c fp22alu11$ vi programa6.c
[Eslovaquia50:programa6.c fp22alu11$ gcc programa6.c -o programa6.out
programa6.c: In function 'main':
programa6.c:14:7: error: 'bytesLeidos' undeclared (first use in this function)
   14 | while(bytesLeidos = fread(buffer, 1, 2048, ap)) {
      |       ^~~~~~~~~~~
programa6.c:14:7: note: each undeclared identifier is reported only once for each function it appears in
[Eslovaquia50:programa6.c fp22alu11$ vi programa6.c
[Eslovaquia50:programa6.c fp22alu11$ gcc programa6.c -o programa6.out
programa6.c: In function 'main':
programa6.c:18:7: error: 'bytesLeidos' undeclared (first use in this function)
   18 | while(bytesLeidos = fread(buffer, 1, 2048, ap))
      |       ^~~~~~~~~~~
programa6.c:18:7: note: each undeclared identifier is reported only once for each function it appears in
Eslovaquia50:programa6.c fp22alu11$ vi programa6.c
[Eslovaquia50:programa6.c fp22alu11$ gcc programa6.c -o programa6.out
[Eslovaquia50:programa6.c fp22alu11$ ./programa6.out
Ejecutar el programa de la siguiente manera:
        nombre_ programa nombre_archivo
Eslovaquia50:programa6.c fp22alu11$
[Eslovaquia50:programa6.c fp22alu11$ vi programa6.c
[Eslovaquia50:programa6.c fp22alu11$ ./programa6.out programa6.c
#include <stdio.h>
int main(int argc, char **argv)
{
FILE *ap;
unsigned char buffer[2048]; // Buffer de 2 Kbytes
int bytesLeidos;
// Si no se ejecuta el programa correctamente
if(argc < 2)
{
printf("Ejecutar el programa de la siguiente manera: \n\tnombre_\tprograma nombre_archivo\n");
return 1; }
// Se abre el archivo de entrada en modo lectura y binario
ap = fopen(argv[1], "rb");
if(!ap)
{
printf("El archivo %s no existe o no se puede abrir", argv[1]);
 return 1;
}
while(bytesLeidos = fread(buffer, 1, 2048, ap))
{
printf("%s", buffer);
}
fclose(ap);
return 0;
}

Eslovaquia50:programa6.c fp22alu11$
```



```
#include <stdio.h>
int main(int argc, char **argv)
{
FILE *archEntrada, *archivoSalida;
unsigned char buffer[2048]; // Buffer de 2 Kbytes int bytesLeidos;
    // Si no se ejecuta el programa correctamente
if(argc < 3)
{
printf("Ejectuar el programa de la siguiente manera:\n");
printf("\tnombre_programa \tarchivo_origen \tarchivo_destino\n");
return 1;
}
// Se abre el archivo de entrada en modo de lectura y binario
archEntrada = fopen(argv[1], "rb");
if(!archEntrada)
{
printf("El archivo %s no existe o no se puede abrir", argv[1]);
return 1;
}
// Se crea o sobreescribe el archivo de salida en modo binario
archivoSalida = fopen(argv[2], "wb");
if(!archivoSalida)
{
printf("El archivo %s no puede ser creado", argv[2]);
return 1;
}
// Copia archivos
while (bytesLeidos = fread(buffer, 1, 2048, archEntrada)) fwrite(buffer, 1, bytesLeidos, archivoSalida);
// Cerrar archivos
fclose(archEntrada);
fclose(archivoSalida);
return 0;
}
~
~
~
~
~
~
~
~
~
-- INSERT --
```

Sintaxis de funciones.:

int fnNombredelprograma()

{

//Bloque de código de la función

}

Funciones de lectura y escritura en un archivo:

La función fgets() permite leer un un archivo y la función fputs() permite escribir en un archivo y la sintaxis de ambas aparece en la tabla

Tarea:

1.- Completa el cuadro referente a funciones, sintaxis, ejemplo y características de cada una de las funcionesvistas en el laboratorio.

Función	Sintaxis	Características	Ejemplo de sintaxis

fputs()	fputs char *fputs(char *buffer, FILE *apArch);	La función fputs() permite escribir una cadena en un archivo especifico	fputs (listaNombre, archivo);
fgets()	char*fputs(char*buffer,FILE *apArch)	Permite leer una cadena de código del archivo especificado y solo lee un renglón a la vez.	Fputs (escribir, archivo);
fopen()	*FILEfopen(char *nombre_archivo,char *modo);	Existen diferentes modos en los que se puede abrir el archivo y se puede usar más de uno.	=fopen(“archivo.txt” , “r”);
fclose()	int fclose(FILE*apArch);	Finaliza el archivo que se abrió, si se cierra de forma incorrecta se puede dañar el archivo o incluso destruirse.	Int res = fclose(archivo);
fprintf()	int fprintf(FILE*apArch,char *formato, ...);	apArch es un apuntador el cual al archivo devuelto por una función llamada fopen(), que funciona de la misma manera que printf.	Fprintf(archivo, escribir);
fscanf()	int fscanf(FILE*apArch,char *formato, ...);	Del mismo modo aquí apArch es un apuntador, el cual sirve de la misma manera que scanf.	Fscanf(archivo, “%s” , caracteres);
fread()	int fread(void*ap,size_t tam,size_tnelem,FILE*archivo)	Permite leer a partir de un apuntador uno o varios elementos de la misma longitud.	Fread(buffer, 1, 2048, ap)
fwrite()	int fwrite(void*ap,size_t tam, size_tnelem,FILE*archivo)	Permite escribir hacia un archivo uno o varios elementos de la misma magnitud que se encuentran almacenados en un apuntador.	Fwrite(buffer, 1, bytesLeidos, archivoSalida);

Conclucion:

Las funciones son muy útiles porque permiten dividir un programa en partes más pequeñas y fáciles de manejar. Nos ayudan a organizar nuestro código, evitar repetirlo y resolver problemas de manera más fácil. Las funciones nos será útil en el futuro ya que esta herramienta será útil para hacer programas más claras y fáciles al desarrollar.

Bibliografia:

(No se hizo uso de investigación en la web)