



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Ing. Karina García Morales

Asignatura: Fundamentos de la Programación

Grupo: 20

No. de práctica(s): 12

Integrante(s): Martínez Ordoñez Diego Tonatiah

No. de lista o brigada: 30

Semestre: 2023-1

Fecha de entrega: 09 / enero / 2023

Observaciones:

CALIFICACIÓN: _____

Práctica 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:

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.

- Apuntador a archivo:

Un apuntador a un archivo es un hilo común que unifica el sistema de Entrada/Salida (E/S) con un buffer donde se transportan los datos. Un apuntador a archivo señala a la información que contiene y define ciertas características sobre él, incluyendo el nombre, el estado y la posición actual del archivo.

FILE *F;

- 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);

Existen diferentes modos de apertura de archivos, los cuales se mencionan a continuación, además de que se pueden utilizar más de uno solo:

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:

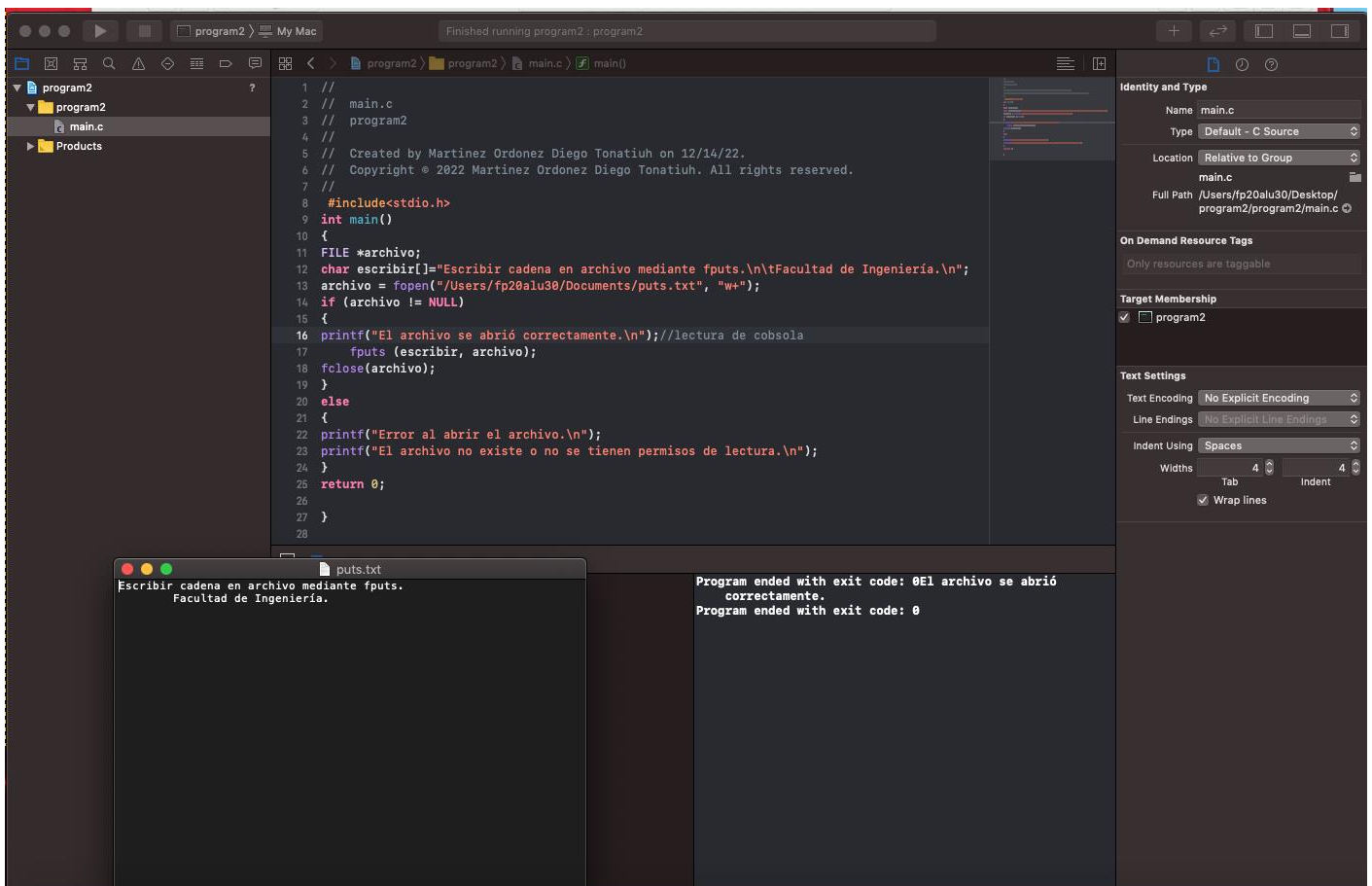
La función fclose() cierra una secuencia que fue abierta mediante una llamada a fopen(). Escribe la información que se encuentre en el buffer al disco y realiza un cierre formal del archivo a nivel del sistema operativo.

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:

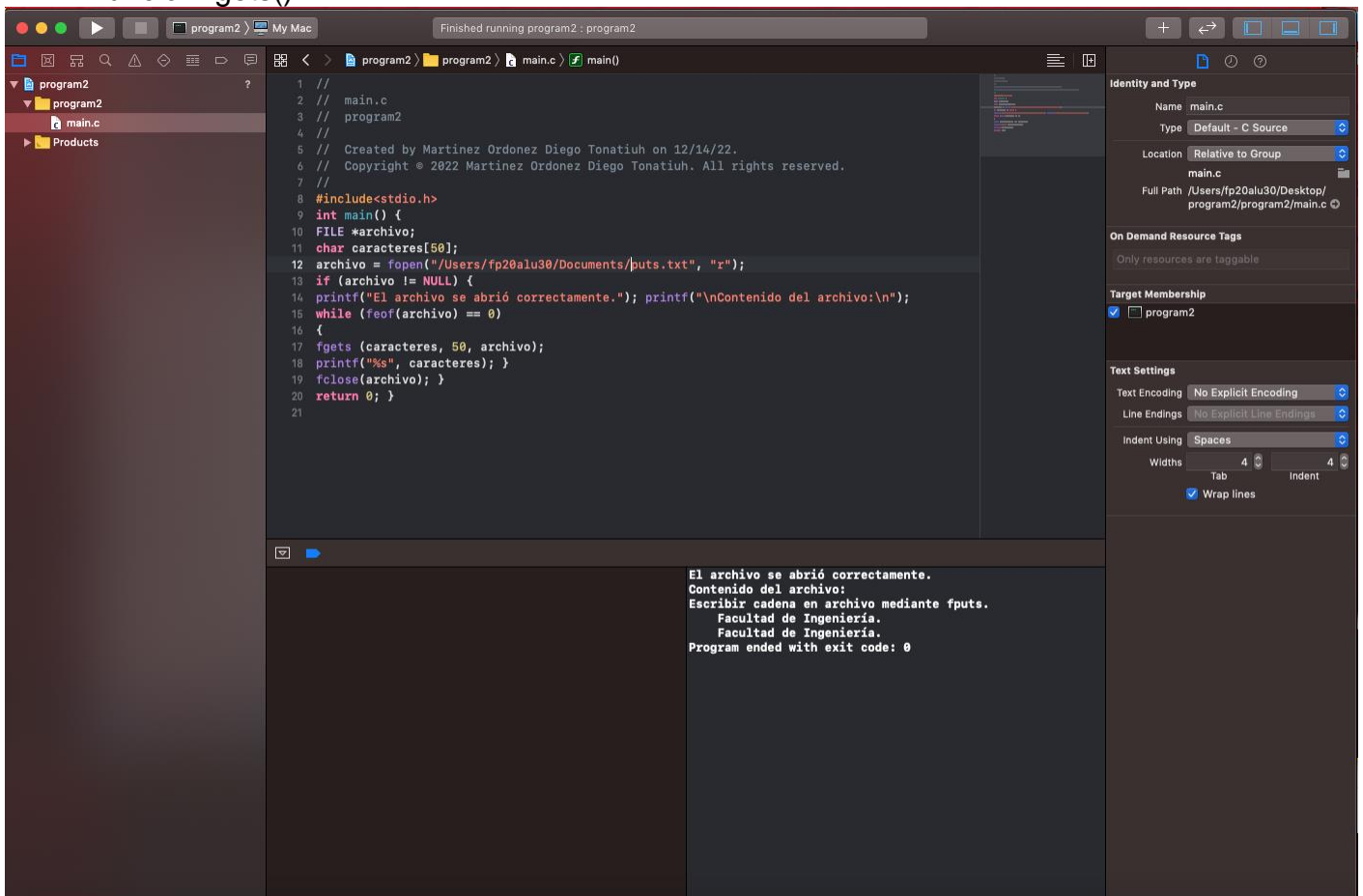
int fclose (FILE *apArch);

- Códigos en clase:

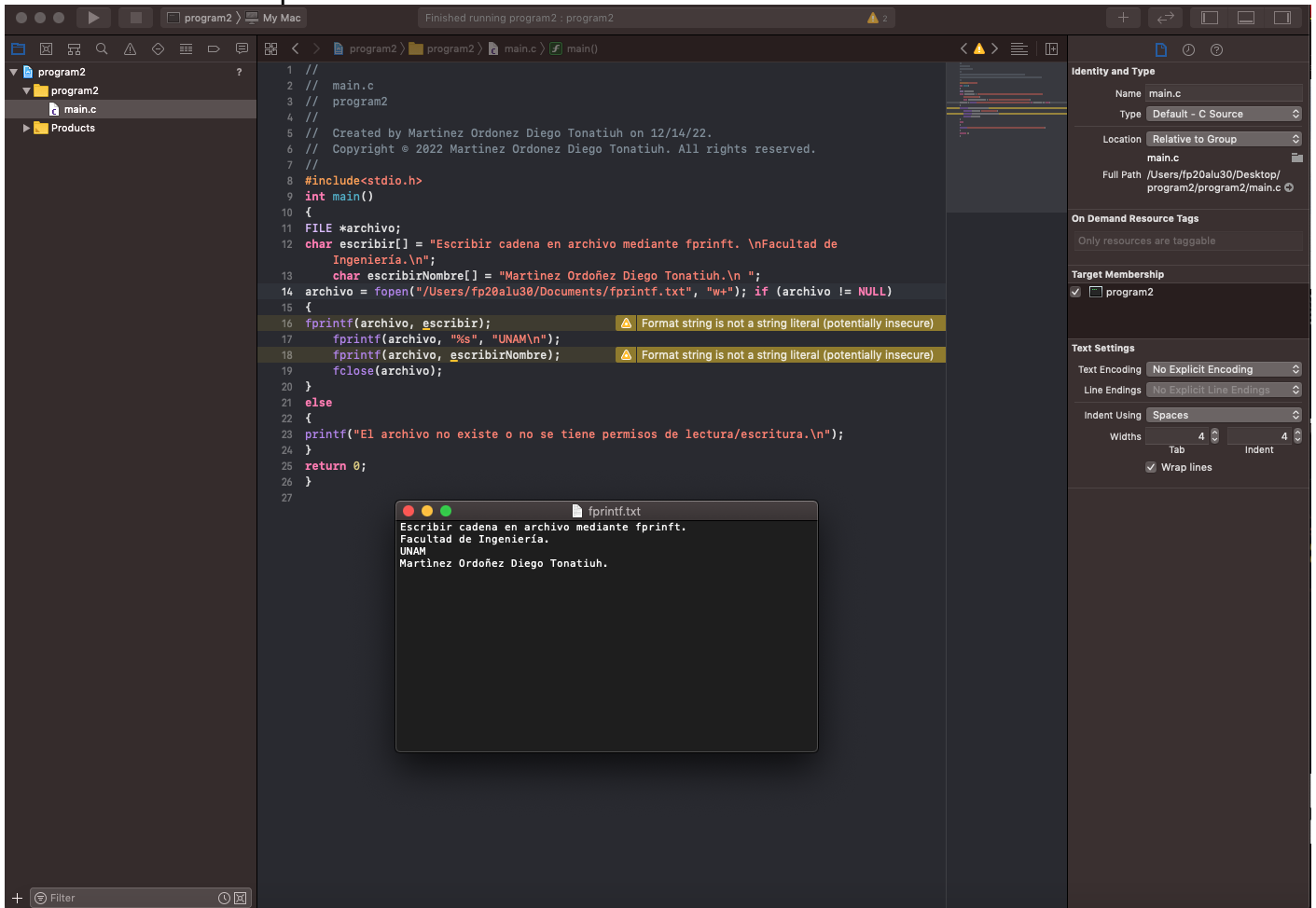
1. Este programa permite abrir un archivo en modo de lectura, de ser posible ya que el archivo debe existir o debe tener permiso de lectura.



2. Este programa permite leer el contenido de un archivo, de ser posible, a través de la función `fgets()`.



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



- Funciones `fscanf` y `fprintf`:

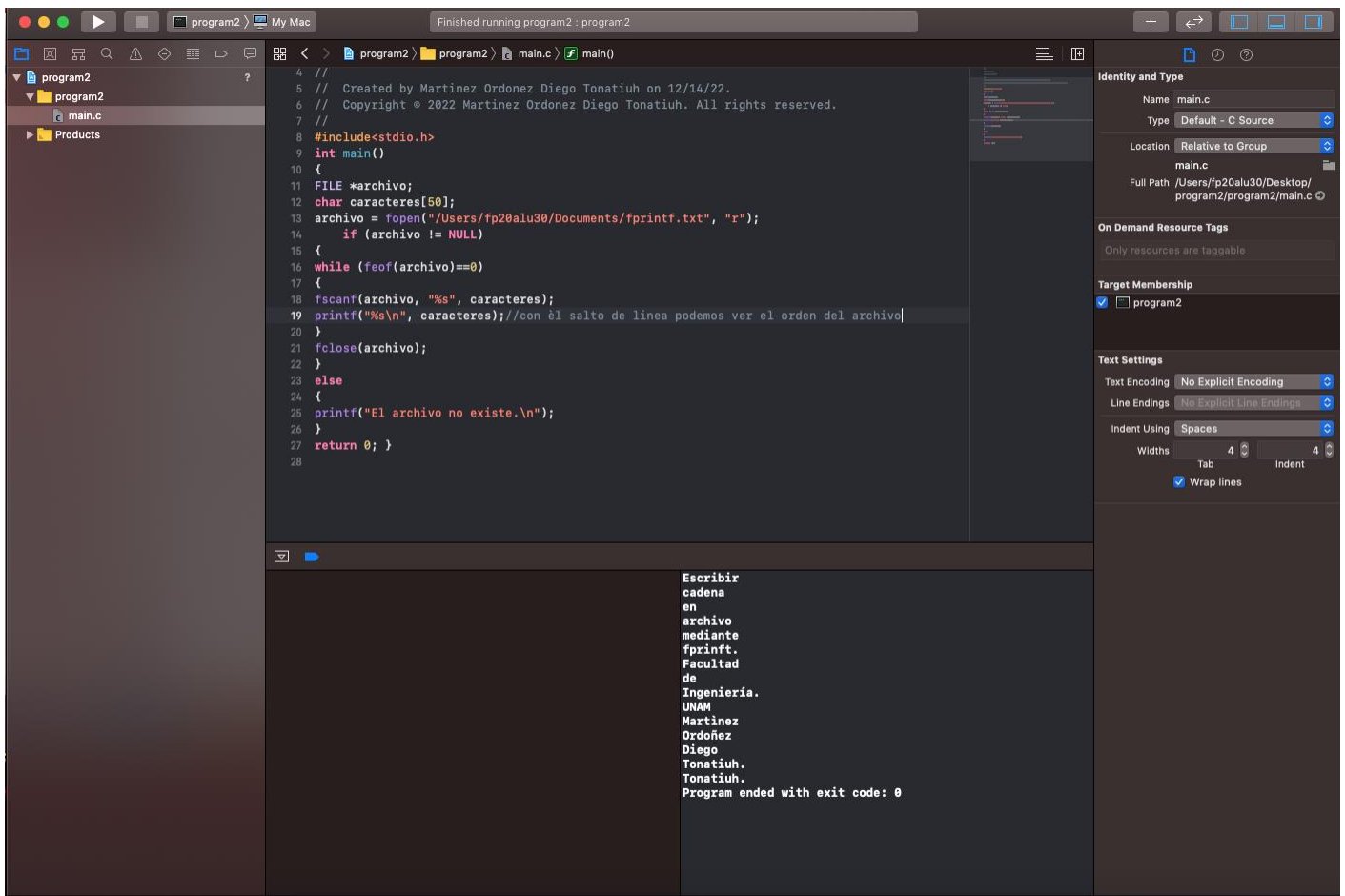
Las funciones `fprintf`() y `fscanf`() se comportan similarmente a `printf`() (imprimir) y `scanf`() (leer), con la diferencia de que operan sobre un archivo. Sus estructuras son:

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

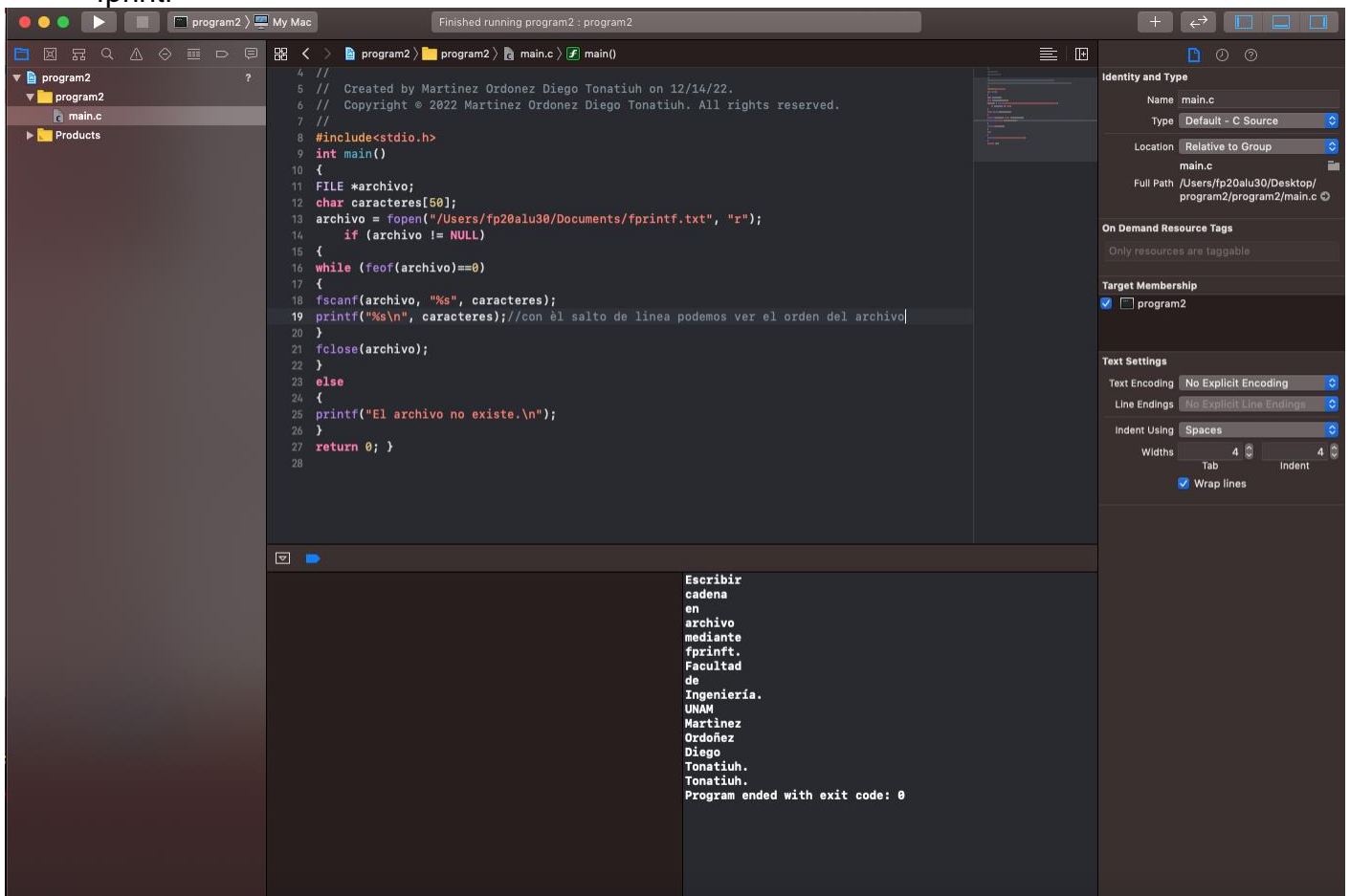
Donde `apArch` es un apuntador al archivo devuelto por una llamada a la función `fopen`(), es decir, `fprintf`() y `fscanf`() dirigen sus operaciones de E/S al archivo al que apunta `apArch`. `formato` es una cadena que puede incluir texto o especificadores de impresión de variables. En los puntos suspensivos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

- Códigos en clase:

4. Este programa permite leer el contenido de un archivo, de ser posible, a través de la función `fscanf`.



5. Este programa permite escribir dentro de un archivo, de ser posible, a través de la función `fprintf`



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

- Código en clase:

6. Este programa muestra el contenido de un archivo de texto. El nombre del archivo se recibe como argumento de la función principal.

```
#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\t\nombre_\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;
}
```

7. 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.

```
Documents — -bash — 103x20
Arabia06:~ fp20alu30$ ./program8.out
-bash: ./program8.out: No such file or directory
Arabia06:~ fp20alu30$ vi program8.c
Arabia06:~ fp20alu30$ gcc program8.c -o program0.out
Arabia06:~ fp20alu30$ ./program8.out
-bash: ./program8.out: No such file or directory
Arabia06:~ fp20alu30$ gcc program8.c -o program0.out
Arabia06:~ fp20alu30$ cd Documents/
Arabia06:Documents fp20alu30$ vi program87.c
Arabia06:Documents fp20alu30$ gcc program87.c -o program87.out
Arabia06:Documents fp20alu30$ ./program87.out fprintf.txt
Arabia06:Documents fp20alu30$ ./program87.out
Ejecutar el programa de la siguiente manera:
    nombre_programa nombre_archivo
Arabia06:Documents fp20alu30$ cat fprintf.txt
Escribir cadena en archivo mediante fprintf.
Facultad de Ingeniería.
UNAM
Martínez Ordoñez Diego Tonatiuh.
Arabia06:Documents fp20alu30$
```

- Conclusiones:

Ahora que ya conocemos nuevas funciones para poder imprimir en pantalla nuevas funciones y de igual manera para poder compilar estos, nos darán un nuevo refuerzo de lo que ya tenemos conocimiento, para poder reforzar las alternativas que tenemos para poder realizar algunos trabajos y de esta forma el ir progresando.

Estas prácticas nos ayudan inmensamente, ya que, nos ayudan a buscar alternativas para poder realizar trabajos, no solo quedarse con una opción, si no que tengamos las suficientes para poder sac adelante una situación de cualquier circunstancia, y de esta manera nos implique mejorar y adquirir experiencia a tal punto de saber el que hacer.

Durante este curso pudimos adquirir las habilidades necesarias para poder desarrollar programas y realizar la función que nosotros le asignamos, pudiendo desarrollar diferentes tipos de funciones.

Al igual, aprendimos a como hay funciones que te pueden ahorrar trabajo y hacer más fácil el poder trabajar.

- Referencia:

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