



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Alejandro Pimentel Alarcón

*Asignatura:* Fundamentos de Programación

*Grupo:* Bloque 35

*No de Práctica(s):* Práctica 13.

*Integrante(s):* Partida Arias Emily Rachel

*No. de Lista o Brigada:* 41

*Semestre:* 2020-1

*Fecha de entrega:* Lunes 11 de noviembre

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## Práctica 13. Lectura y escritura de datos.

### 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.

### Introducción:

#### FICHEROS

El estándar de C contiene varias funciones para la edición de ficheros, éstas están definidas en la cabecera `stdio.h` y por lo general empiezan con la letra `f`, haciendo referencia a `file`. Adicionalmente se agrega un tipo **FILE**, el cual se usará como apuntador a la información del fichero. La secuencia que usaremos para realizar operaciones será la siguiente:

- Crear un apuntador del tipo **FILE**
- Abrir el archivo utilizando la función **fopen** y asignándole el resultado de la llamada a nuestro apuntador.
- Hacer las diversas operaciones (lectura, escritura, etc).
- Cerrar el archivo utilizando la función **fclose**.

### Lista de parámetros `opentype` para la función `fopen` son:

- `"r"` : abrir un archivo para lectura, el fichero debe existir.
- `"w"` : abrir un archivo para escritura, se crea si no existe o se sobrescribe si existe.
- `"a"` : abrir un archivo para escritura al final del contenido, si no existe se crea.
- `"r+"` : abrir un archivo para lectura y escritura, el fichero debe existir.
- `"w+"` : crear un archivo para lectura y escritura, se crea si no existe o se sobrescribe si existe.
- `"r+b` ó `rb+"` : Abre un archivo en modo binario para actualización (lectura y escritura).
- `"rb"` : Abre un archivo en modo binario para lectura.

#### LECTURA

Un archivo generalmente debe verse como un string (una cadena de caracteres) que está guardado en el disco duro. Para trabajar con los archivos existen diferentes formas y diferentes funciones. Las funciones que podríamos usar para leer un archivo son:

- `char *fgets(char *buffer, int tamano, FILE *archivo)`
- `size_t fgetc(FILE *archivo)`
- `_t fread(void *puntero, size_t tamano, size_t cantidad, FILE *archivo);`
- `int fscanf(FILE *fichero, const char *formato, argumento, ...);`

## fgets

Esta función está diseñada para leer cadenas de caracteres. Leerá hasta n-1 caracteres o hasta que lea un cambio de línea '\n' o un final de archivo EOF. En este último caso, el carácter de cambio de línea '\n' también es leído.

### Desarrollo:

**Actividad. Crear un programa que pida el nombre de un archivo de entrada y un archivo de salida.**

**Para el archivo de entrada, mostrar:**

- **Texto.**
- **Número de líneas.**
- **Número de palabras (cualquier cosa entre espacios).**
- **Número de caracteres.**

**Para el archivo de salida:**

- **Copiar el archivo de entrada con las líneas invertidas.**

```
1 #include <stdio.h>
2 #include <string.h>
3 int main(){
4     FILE *archivo,*archivosalida;
5     char palabra [21], linea[101];
6     printf("Nombre del archivo\n");
7     char nombre[21];
8     scanf("%s",nombre);
9     char nombresalida[21];
10    printf("Nombre del nuevo archivo\n");
11    scanf("%s",nombresalida);
12    archivo=fopen(nombre,"r");
13    int contadorlineas=0;
14    while(!feof(archivo)){
15        fgets(linea,100,archivo);
16        printf("%s",linea);
17        contadorlineas++;
18    }
19    printf("\nnumero de lineas: %i\n",contadorlineas);
20    int contadorpalabras=0;
21    archivo=fopen(nombre,"r");
22    while(!feof(archivo)){
23        fscanf(archivo,"%8s",palabra);
24        contadorpalabras++;
25    }
26    printf("numero de palabras:%i\n",contadorpalabras);
27    archivo=fopen(nombre,"r");
28    int contadorcaracteres=0,npalabra;
29    while(!feof(archivo)){
30        fscanf(archivo,"%8s",palabra);
31        npalabra=strlen(palabra);
32        contadorcaracteres=contadorcaracteres+npalabra;
33    }
34    printf("numero de caracteres %i\n",contadorcaracteres);
35    archivo=fopen(nombre,"r");
36    archivosalida=fopen(nombresalida,"w");
37    char listaarchivo[contadorlineas][100];
38    for(int i=contadorlineas-1;i!=-1;i--){
39        fgets(listaarchivo[i],100,archivo);
40    }
41    for(int i=0;i<contadorlineas;i++){
42        if(i==0){
43            fprintf(archivosalida,"%s",listaarchivo[i]);
44        }
45    }
46    return 0;
47 }
48
```

Para este programa se usó otra librería para usar nuevas funciones durante el programa, declaramos los apuntadores y con estos se va a poder usar el texto, abrimos un archivo para leer y declaramos en Contador de líneas.

Luego se Vuelve a abrir el archivo pero ahora para que se invierta y comience en el inicio, con fscanf se aumenta el Contador cada que pasa una palabra, se imprimió y Vuelve a abrir el archivo desde el principio

Luego se hace una matriz para guardar las líneas y ahora si el texto va a empezar de el principio y se imprimen las líneas.

## Al compilar en la terminal:

```
plano 3.pdf
plano 4.pdf
plano 5.pdf
plano1.pdf
practica microscopio 510.docx
practicass
principio-de-direccion-comun.jpg
zacatlan281216.docx
iMac-de-Mauro:~ mauropartida$ gcc main13.cpp -o main
clang: error: no such file or directory: 'main13.cpp'
clang: error: no input files
iMac-de-Mauro:~ mauropartida$ cd desktop
iMac-de-Mauro:desktop mauropartida$ gcc main13.cpp -o main
main13.cpp:23:25: warning: incomplete format specifier [-Wformat]
    fscanf(archivo,"%8",palabra);
                        ^~
1 warning generated.
iMac-de-Mauro:desktop mauropartida$ ./main
Nombre del archivo
calaverita.txt
```

En este mes singular,  
En que todo es fiesta y danza,  
Recordamos con amor  
Tradiciones y alabanzas.

Es menester recordar  
Que aún en la confianza,  
De volvernos a encontrar,  
Recordamos sus andanzas.

A nuestros seres queridos  
Les ponemos un altar,  
Pues sus almas y latidos  
Los sentimos regresar!

Así que los festejamos  
Con grandes piezas de pan,  
Colocados en altares  
Con flores velas, mezcal!

Estos panes primorosos,  
Que de Colores están,  
Con formas de cuerno y hueso  
Se los hemos de dejar.

Para que pasen contentos  
En su visita fugaz,  
En esta tierra de amores  
Los recuerdos que se van.

Para mí es un gran regalo  
El poderlos encontrar,  
En fechas tan especiales  
Y poderlos disfrutar.

Si supieran mis muertitos  
Cuánto los echo de menos,  
Vendrían todos los días  
Y se harían muy amenos!

Yo los espero sentada  
Pues sé que departiremos  
Estas dos noches completas  
Con atoles y galletas.

Queridas almas contentas,  
De saber que las queremos,  
Recordamos sus amores,  
Y esperamos su regreso!

Autora: Davina Gpe. Ponce Mtz.

numero de líneas: 55  
numero de palabras: 198  
numero de caracteres 849

## Calaverita al revés:

Autora: Davina Gpe. Ponce ~~Mtz.~~

Y esperamos su regreso!  
Recordamos sus amores,  
De saber que las queremos,  
Queridas almas contentas,

Con atoles y galletas.  
Estas dos noches completas  
Pues sé que departiremos  
Yo los espero sentada

Y se harían muy amenos!  
Vendrían todos los días  
Cuánto los echo de menos,  
Si supieran mis ~~muertitos~~

Y poderlos disfrutar.  
En fechas tan especiales  
El poderlos encontrar,  
Para mí es un gran regalo

Los recuerdos que se van.  
En esta tierra de amores  
En su visita fugaz,  
Para que pasen contentos

Se los hemos de dejar.  
Con formas de cuerno y hueso  
Que de Colores están,  
Estos panes primorosos,

Con flores velas, mezcal!  
Colocados en altares  
Con grandes piezas de pan,  
Así que los festejamos

Los sentimos regresar!  
Pues sus almas y latidos  
Les ponemos un altar,  
A nuestros seres queridos

Recordamos sus andanzas.  
De volvernos a encontrar,  
Que aún en la confianza,  
Es menester recordar

Tradiciones y alabanzas.  
Recordamos con amor  
En que todo es fiesta y danza,  
En este mes singular,

**Conclusión:**

Es importante saber cómo trabajar con archivos en C, y saber para qué sirven los punteros, que sirven para almacenar las direcciones de las variables, cada variable tiene un espacio diferente por esto cuando declaramos un puntero hay que decir el tipo de dato que almacenará.

**Referencias:**

[https://es.wikibooks.org/wiki/Programaci%C3%B3n\\_en\\_C/Manejo\\_de\\_archivos](https://es.wikibooks.org/wiki/Programaci%C3%B3n_en_C/Manejo_de_archivos)