	<b>Lectura y escritura de datos</b>	
Facultad de Ingeniería	Laboratorio de docencia	

Laboratorios de computación  
salas A y B

*Profesor:* Alejandro Pimentel

*Asignatura:* Fundamentos de Programación

*Grupo:* 135

*No de Práctica(s):* 13

*Integrante(s):* Lorena Basurto Amezcu

*No. de Equipo de  
cómputo empleado:*

*No. de Lista o Brigada:* 2858

*Semestre:* 2020-1

*Fecha de entrega:* Noviembre 11, 2019.

*Observaciones:*

CALIFICACIÓN: \_\_\_\_\_

## Objetivo:

Elaborar programas en 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:

- **Archivos**

Para trabajar con archivos en C, es necesario tener un apuntador hacia un archivo:

```
FILE *archivo;
```

Para asignar el apuntador a su lugar correspondiente, podemos contar con una función para abrir el archivo por nombre:

```
archivo = fopen("archivo.txt", "r");
```

A partir de este punto, ya podemos utilizar nuestro apuntador de archivo. Pero para leer, necesitaremos una variable en dónde guardar el texto:

```
char linea[90];  
char palabra[9];
```

- **Formas para abrir**

Al momento de abrir un archivo (*fopen*) se puede elegir una entre varias opciones:

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.

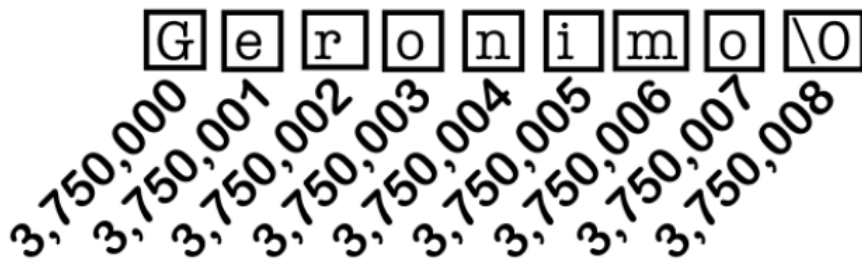
- **String.h**

String es una librería que será de utilidad siempre que tengamos que manejar texto (cadenas de caracteres).

```
#include <string.h>
```

```
strlen(char texto[]);  
strcpy(char destino[], char origen[]);  
strcmp(char texto1[], char texto2[]);  
strcat(char destino[], char origen[]);  
strstr(char texto[], char buscado[]);  
strchr(char texto[], char buscado);
```

El secreto de las cadenas de caracteres es que indican su final con el caracter nulo "\0". Pero deben recordar que ese caracter también ocupa un lugar en el arreglo.



- **Leer**

Tenemos dos formas sencillas de leer texto desde un archivo:

```
fscanf(archivo,"%8s",palabra);
```

Que funciona igual que *scanf()* con la diferencia de que como primer parámetro recibe el apuntador hacia el archivo. Recuerden que esto solo lee una palabra a la vez. El número después del porcentaje, es para limitar la cantidad de caracteres máximos que toma, esto es útil para no sobrepasar la longitud del arreglo de caracteres.

Y la otra manera es:

```
fgets(linea,89,archivo);
```

Que lee una línea completa, con un número máximo de caracteres que recibe como segundo argumento.

Observen que el orden del apuntador y la variable se invierten.

Ambas formas irán haciendo que el archivo "avance". Eso quiere decir que si las usan repetidas veces, irán leyendo nuevas palabras/líneas del archivo.

```
while( ! feof(archivo) ){  
    fgets(linea,89,archivo);  
    printf("%s",linea);  
}
```

La función *feof* nos ayuda a darnos cuenta si ya llegamos al final del archivo. La función recibe como único argumento el apuntador al archivo, y devuelve 0 mientras no sea el final del archivo.

- **Escribir**

También hay varias formas de escribir en un archivo, pero por familiaridad, la más sencilla es *fprintf*.

```
fprintf(archivo,"%sn","texto");
```

Al igual que con *fscanf*, lo único que cambia es que el primer parámetro es el apuntador del archivo.

- **Cerrar**

Por último, hay que cerrar el archivo que abrimos con *fopen*. Se cierra con *fclose*.

```
fclose(archivo);
```

- **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
4  int main() {
5
6      FILE *archivoEntrada; // Se utiliza un apuntador hacia el archivo de entrada
7      FILE *archivoSalida; // Se utiliza un apuntador hacia el archivo de salida
8
9      char palabra[21]; // Se declara una variable que podra contener una palabra de hasta 20 caracteres
10     char linea[101]; // Se declara una variable que podra contener
11
12     printf("Nombre del archivo:\n");
13     char nombreArchivo[21];
14     scanf("%s", nombreArchivo);
15
16     char nombreArchivo2[21];
17     printf("Nombre del archivo nuevo:\n");
18     scanf("%s", nombreArchivo2);
19
20     archivoEntrada = fopen(nombreArchivo, "r"); // Se abre el archivo para lectura
21
22     int contadorLineas = 0; // Se declara una variable que servira como contador de lineas
23
24     while(!feof(archivoEntrada)) {
25         fgets(linea, 100, archivoEntrada); // La funcion 'fgets' servira para leer el texto desde el archivo
26         printf("%s", linea); // 'fgets' lee lineas completas y 'printf' las imprime una por una
27         contadorLineas++; // Cada vez que se imprima una linea el contador suma 1
28     }
29
30     printf("\nNumero de lineas en el texto: %i\n", contadorLineas); // 'contadorLineas' contiene el numero total de lineas en el texto
31
32     int contadorPalabras = 0; // Se declara una variable para contar en numero de palabras en el texto
33
34     archivoEntrada = fopen(nombreArchivo, "r"); // Se abre de nuevo el archivo para iniciar desde el principio
35
36     while(!feof(archivoEntrada)) {
37         fscanf(archivoEntrada, "%s", palabra); // En este caso se utiliza la funcion 'fscanf' pues leera el archivo palabra por palabra
38         contadorPalabras++; // El contador suma 1 cada vez que se lee una palabra
39     }
40
41     printf("Numero de palabras en el texto: %i\n", contadorPalabras); // Se imprime el numero de palabras en el contador
42
43     archivoEntrada = fopen(nombreArchivo, "r"); // Se vuelve a abrir el archivo
44
45     int contadorCaracteres = 0; // Se declara una variable para contar el numero de caracteres
46     int charsPalabra; // Se declara una variable para guardar el numero de caracteres en una palabra dada
47
48     while(!feof(archivoEntrada)) {
49         fscanf(archivoEntrada, "%s", palabra); // Escanea una palabra y la guarda en la variable 'palabra'
50         charsPalabra = strlen(palabra); // El numero de caracteres de la palabra se guarda en la variable 'charsPalabra'
51         contadorCaracteres += charsPalabra; // Se hace la sumatoria
52     }
53
54     printf("Numero de caracteres en el texto: %i\n", contadorCaracteres); // Se imprime el numero de caracteres en el texto
55
56     archivoEntrada = fopen(nombreArchivo, "r"); // Se abre el archivo para lectura de nuevo
57     archivoSalida = fopen(nombreArchivo2, "w"); // Con 'fopen' y la opcion 'w' se crea un archivo de texto para escritura
58
59     char listaArchivo[contadorLineas][100]; // Se crea una matriz para guardar las lineas
60
61     for(int i=contadorLineas-1; i>=0; i--) { // Se recorre el texto de atras hacia adelante
62         fgets(listaArchivo[i], 100, archivoEntrada);
63     }
64
65     for(int i=0; i<contadorLineas; i++) {
66         if(i==0) {
67             fprintf(archivoSalida, "%s\n", listaArchivo[i]);
68         }
69         else {
70             fprintf(archivoSalida, "%s", listaArchivo[i]);
71         }
72     }
73
74
75     return 0;
76 }
```

/cygdrive/c/Users/lbasurtoa/FP\_2020-1\_2858

```
lbasurtoa@DESKTOP-UVTH5Q4 /cygdrive/c/Users/lbasurtoa/FP_2020-1_2858
$ gcc P13.c -o P13
```

```
lbasurtoa@DESKTOP-UVTH5Q4 /cygdrive/c/Users/lbasurtoa/FP_2020-1_2858
$ ./P13
```

Nombre del archivo:  
calaverita.txt  
Nombre del archivo nuevo:  
calaverita2.txt  
En este mes singular,  
En que todo es fiesta y danza,  
Recordamos con amor  
Tradiciones y alabanzas.

Es menester recordar  
Que aA^n 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!

AsA- que los festejamos  
Con grandes piezas de pan,  
Colocados en altares  
Con flores velas, mezcalt!

Estos panes primorosos,  
Que de Colores estA^n,  
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 mA- es un gran regalo  
El poderlos encontrar,  
En fechas tan especiales  
Y poderlos disfrutar.

Si supieran mis muertitos  
CuA^nto los echo de menos,  
VendrA-an todos los dA-as  
Y se harA-an muy amenos!

Yo los espero sentada  
Pues sA@ 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 lineas en el texto: 55  
Numero de palabras en el texto: 198  
Numero de caracteres en el texto: 849

```
lbasurtoa@DESKTOP-UVTH5Q4 /cygdrive/c/Users/lbasurtoa/FP_2020-1_2858
$ |
```

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 sÃ© que departiremos  
Yo los espero sentada

Y se harÃ¡n muy amenos!  
VendrÃ¡n todos los dÃ¡-as  
CuÃ¡nto los echo de menos,  
Si supieran mis muertitos

Y poderlos disfrutar.  
En fechas tan especiales  
El poderlos encontrar,  
Para mÃ¡- 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 estÃ¡n,  
Estos panes primorosos,

---

Con flores velas, mezcal!  
Colocados en altares  
Con grandes piezas de pan,  
AsÃ¡- 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 aÃ©n en la confianza,  
Es menester recordar

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



**Conclusiones:**

La práctica fue muy útil para comprender claramente cómo se utilizan los apuntadores, así como funciones tales como *fopen*, *fscanf*, *fgets* y *fprintf*, para la manipulación y modificación de textos planos.