

	Caratula para entrega de prácticas
Facultad de ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Pimentel

Asignatura: Fundamentos de Programación

Grupo: 3

No de Práctica(s): 13

Integrante(s): Arteaga Munguía Erick Alejandro

*No. de Equipo de
cómputo empleado:* Ruanda

No. de Lista o Brigada: 6294

Semestre: 2020-1

Fecha de entrega: 10/11/19

Observaciones:

CALIFICACIÓN: _____

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.

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 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 carácter nulo "\0". Pero deben recordar que ese carácter también ocupa un lugar en el arreglo.

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.

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.

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

```
fclose(archivo);
```

```
prac13.c x
#include <stdio.h>
#include <string.h>
int main (){
FILE *archivo;
archivo=fopen("calaverita.txt","r");
char linea [100];
char palabra [15];
fscanf(archivo, "%14s",palabra);
fgets(linea, 99, archivo);
while( ! feof(archivo)){
    fgets(linea, 99, archivo);
    printf("%s",linea );
}
fprintf(archivo, "%sn","texto" );
fclose(archivo);
}
```

Escritorio — -bash — 80x34

Ruanda37:Desktop fp03alu03\$ gcc prac13.c -o main

Ruanda37:Desktop fp03alu03\$./mian

-bash: ./mian: No such file or directory

Ruanda37:Desktop fp03alu03\$./main

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


```

#include <stdio.h>
#include <string.h>
int main (){
FILE *archivo,*archivosalida;
char linea [100];
char palabra [15];
printf("¿Cual es el nombre del archivo?\n");
char nombre [15];
scanf("%s",nombre);
char nombresalida[15];
printf("¿Cual es el archivo nuevo?\n");
scanf("%s",nombresalida);
archivo=fopen(nombre,"r");
int contadorlineas=0;
while(!feof(archivo)){
fgets(linea, 99, archivo);
printf("%s",linea);
contadorlineas++;
}
printf("\nnumero de lineas: %i\n",contadorlineas);
int contadorpalabras=0;
archivo=fopen(nombre,"r");
while(!feof(archivo)){
fscanf(archivo, "%8s",palabra);

```

```

printf("numero de palabras: %i\n",contadorpalabras);
archivo=fopen(nombre,"r");
int contadorcaracteres=0, npalabra;
while(!feof(archivo)){
fscanf(archivo,"%8s",palabra);
npalabra=strlen(palabra);
contadorcaracteres=contadorcaracteres+npalabra;
}
printf("el numero de caracteres %i\n",contadorcaracteres);
archivo=fopen(nombre,"r");
archivosalida=fopen(nombresalida,"w");
char listaarchivo[contadorlineas][99];
for(int i=contadorlineas-1;i!=-1;i--){
fgets(listaarchivo[i],99,archivo);
}
for(int i=0;i<contadorlineas;i++){
if(i==0){
fprintf(archivosalida,"%s\n",listaarchivo[i]);
}
else{
fprintf(archivosalida,"%s\n",listaarchivo[i]);
}
}
return 0;

```

[illegible]

CONCLUSIÓN:

En lo particular piensa que fue una práctica larga y un poco complicada porque había que realizar un programa con muchas variables y eso complicaba la práctica.

Pero creo que esta práctica es buena ya que aprendí a modificar o leer un archivo usando los comandos que aprendimos en la práctica.