



Carátula 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): Meraz Dionicio Israel

*No. de Equipo de
cómputo empleado:* 1

No. de Lista o Brigada: 8875

Semestre: 1

Fecha de entrega: 11 de Noviembre de 2019

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.

INTRODUCCION

Los archivos son listas largas de datos que se guardan en un sistema de archivo dentro de un dispositivo de almacenamiento masivo; disco duro, pendrive, cdrom, dvdrom, cinta magnética, etc.

Si bien todos los archivos son información binaria, para los efectos de su uso en C se reconocen dos tipos de archivos, los archivos de texto y los archivos binarios.

C contiene funciones para la edición de ficheros, están definidas en `stdio.h` y por lo general empieza con la letra `f`, se muestran en la tabla siguiente:

| Nombre | Función |
|------------------------|---|
| <code>fopen()</code> | Abre un archivo. |
| <code>fclose()</code> | Cierra un archivo. |
| <code>fgets()</code> | Lee una cadena de un archivo. |
| <code>fputs()</code> | Escribe una cadena en un archivo. |
| <code>fseek()</code> | Busca un byte específico de un archivo. |
| <code>fprintf()</code> | Escribe una salida con formato en el archivo. |
| <code>fscanf()</code> | Lee una entrada con formato desde el archivo. |
| <code>feof()</code> | Devuelve cierto si se llega al final del archivo. |
| <code>ferror()</code> | Devuelve cierto si se produce un error. |
| <code>rewind()</code> | Coloca el localizador de posición del archivo al principio del mismo. |
| <code>remove()</code> | Borra un archivo. |
| <code>fflush()</code> | Vacía un archivo. |

Un puntero a un archivo es un puntero a una información que define varias cosas sobre él, incluyendo el nombre, el estado y la posición actual del archivo. Un puntero a un archivo es una variable de tipo puntero a la estructura `FILE` que se define en `stdio.h` para el manejo de ficheros.

Las formas de abrir el archivo son estas:

`r` — abre el archivo en modo lectura.

`w` — abre el archivo para escritura (si no existe lo crea, si existe lo destruye).

`a` — abre el archivo para agregar información (si no existe lo crea).

`r+` — abre el archivo para lectura/escritura (comienza al principio del archivo).

`w+` — abre el archivo para lectura/escritura, sobre-escribe el archivo si este ya existe o lo crea si no).

`a+` — abre el archivo para lectura/escritura (se sitúa al final del archivo).

- Crear un programa que pida el nombre de un archivo de entrada y un archivo de salida
Para el archivo de entrada mostrar: Texto, Numero de lineas, Numero de palabras, Numero de caracteres.
Para el archivo de salida, copiar el archivo de entrada con las lineas invertidas.

```

1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5
6  int contador=0;
7  int contarpalabras=0;
8  int contarcaracteres=0;
9  FILE *archivo, *archivoS;
10 char cadena[350], palabra[35];
11 printf("Nombre del archivo:\n");
12 char documento[25];
13 scanf("%s", documento);
14 printf("Nombre del archivo de salida:\n");
15 char documentoSa[25];
16 scanf("%s", documentoSa);
17 printf("\n");
18 archivo = fopen(documento, "r");
19 while (!feof(archivo)) {
20     fgets(cadena, 350, archivo);
21     printf("%s", cadena);
22     contador++;
23 }
24 printf("\nNumero de lineas: %i\n", contador);
25
26 archivo = fopen(documento, "r");
27 while (!feof(archivo)) {
28     fscanf(archivo, "%8s", palabra);
29     fgets(cadena, 350, archivo);
30     contarpalabras++;
31 }
32 printf("\nNumero de palabras: %i\n", contarpalabras);
33
34 archivo = fopen(documento, "r");
35 while (!feof(archivo)) {
36     fscanf(archivo, "%8s", palabra);
37     int caracter = strlen(palabra);
38     contarcaracteres=contarcaraacteres+caracter;
39 }
40 printf("\nNumero de caracteres: %i\n", contarcaracteres);
41
42 archivo= fopen(documento, "r");
43 archivoS= fopen(documentoSa, "a");
44
45 char cont[contador][350];
46
47 for(int i=contador-1; i != -1; i--){
48     fgets(cont[i], 350, archivo);
49 }
50 for(int i=0; i<contador; i++){
51     if(i==0){
52         fprintf(archivoS, "%s\n", cont[i]);
53     } else {
54         fprintf(archivoS, "%s\n", cont[i]);
55     }
56 }
57 return 0;|
58 }
59

```

Para invertir las lineas del texto, primero se abren los archivos, de entrada y salida, con for.
El primer for hace que la ultima linea se mueva a la primera y asi sucesivamente
Con el segundo for hace que imprima el texto, con el if realiza que la ultima linea empiece, el else ya lo hace con el resto del texto.
A mi se me guardo en el archivo de salida

Se utiliza la librería #include <string.h> para la utilidad de manejar texto

Primero se declaran las variables contador, contarpalabras y contarcaracteres y se inician para utilizarlos después como contadores

Después los apuntadores para que lean el contenido del documento, después, con scanf ingrese el nombre de los documentos, una de entrada y otra de salida.

El primer while, en la conducción, el feof nos ayuda a saber si ya llegamos al final del texto y tiene que ser diferente para que se repita, aproveche a que imprima las cadenas de lineas que tiene el texto y que cuente las lineas y luego se muestra la cantidad de lineas

En el segundo y tercer while es casi el mismo proceso como el primer while, el segundo cuenta las palabras que tiene cada linea, el tercero es que esas palabras con strlen, mida la longitud y con un acumulador vaya guardando y sumando

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,

```

/cygdrive/c/Users/ACER/Documents
ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$ ./tarea
Nombre del archivo:
calaverita.txt
Nombre del archivo de salida:
borrador.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

```

```

/cygdrive/c/Users/ACER/Documents
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!

```

```

/cygdrive/c/Users/ACER/Documents
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: 42
Numero de caracteres: 849
ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$

```

Conclusion:

La utilidad de este funcionamiento en C radica en que una situación x, llegue a que un archivo en word se pueda borrar por el paso del tiempo que lleva la RAM, o solamente queremos escribir solo texto. Es complicado si apenas va adentrando al lenguaje en C y además si no tienes desarrollada una habilidad de organización, saber que requiere el problema y conocer el funcionamiento de la mayoría los códigos de la librería.