Carátula para entrega de prácticas	
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcón

Asignatura: Fundamentos de Programación

Grupo: 03

No de Práctica(s): 13

Integrante(s): Cureño Arvizu Ameyalli Jocelyn

No. de Equipo de cómputo empleado: 0779

No. de Lista o

Semestre: 2020 - 1

Fecha de entrega: 14 de noviembre de 2019

Observaciones: Tarde entrega.
Y recuerda que los datos que yo escribo son solo ejemplos que ustedes deben cambiar, en tu fscanf, "8" son muy pocos caracteres para leer una palabra, debía coincidir con la longitud de tu variable (20)

CALIFICACIÓN: 8

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

C dispone de una colección de funciones de biblioteca para la entrada/lectura y salida/escritura entre el ordenador y los periféricos Entrada/Salida. Se encuentran declaradas en la librería del sistema stdio.h, para ello se coloca en la cabecera de los programas la directiva #include, se considera como unidad de entrada/lectura el teclado y como unidad de salida/escritura la pantalla del ordenador.

Debido a la naturaleza dinámica de los “string”, no hay que preocuparse de la cantidad de memoria que hay que reservar para el “string”. Simplemente hay que añadir cosas y el “string” irá expandiéndose para dar cabida a lo que le introduzca.

Una de las cosas agradables de poner el fichero entero en una cadena es que la clase “string” proporciona funciones para la búsqueda y manipulación que le permiten modificar el fichero como si fuera una simple línea. Sin embargo, tiene sus limitaciones. Por un lado, a menudo, es conveniente tratar un fichero como una colección de líneas en vez de un gran bloque de texto. Por ejemplo, si quiere añadir numeración de líneas es mucho más fácil si tiene un objeto “string” distinto para cada línea. Para realizarlo, necesitamos otro concepto.

1. **ACTIVIDAD**

   Crear un programa que pida el nombre de un archivo de entrada y un archivo de salida.
   Para un archivo de entrada mostrar:

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

   Para un archivo de salida:

   - Copiar el archivo de entrada con las líneas invertidas

```c
#include<stdio.h>
#include<string.h>
int main(){

    FILE *archivo, *archivosalida;

    char palabra [20], linea[100];
    printf("Nombre del archivo\n");
    char nombre[30];
    scanf("%s",nombre);
    char nombresalida[30];
    printf("Nombre del archivo nuevo\n");
    scanf("%s",nombresalida);

    archivo=fopen(nombre,"r");
    int contadorlineas=0;

    while(!feof(archivo)){
        fgets(linea,100,archivo);
        printf("%s",linea);
        contadorlineas++;
    }

```

Incluimos la librería string para poder usar algunas funciones especiales que nos permitan trabajar dentro de los archivos que deseamos, usamos dos apuntadores para los archivos, colocamos unas variables para poder guardar texto y utilizamos scanf para que el usuario pudiese colocar cualquier archivo. Después mandamos la función para que lo abra, y colocamos un ciclo para recibir el número de líneas.

```
24        printf("\n numero de lineas: %i\n",contadorlineas);
25        int contadorpalabras=0;
26
27        archivo=fopen(nombre,"r");
28        while(!feof(archivo)){
29            fscanf(archivo,"%8s",palabra);
30            contadorpalabras++;
31        }
32
33         printf("\n numero de palabras: %i\n",contadorpalabras);
34         archivo=fopen(nombre,"r");
35         int contadorcaracteres=0,npalabra;
36
37          while(!feof(archivo)){
38              fscanf(archivo,"%8s",palabra);
39              npalabra=strlen(palabra);
40              contadorcaracteres=contadorcaracteres+npalabra;
41           }
42           printf("\n numero de caracteres: %i\n",contadorcaracteres);
43         archivo=fopen(nombre,"r");
44        archivosalida=fopen(nombresalida,"w");
45
46        char listaarchivo[contadorlineas][100];
47
```

Después repetimos el mismo proceso para poder leer el número de palabras y al final el número de caracteres, la única diferencia se encuentra en la función strlen que nos sirve para leer caracteres unicamente.

```
47
48        for(int i=contadorlineas-1;i!=-1;i--){
49            fgets(listaarchivo[i],100,archivo);
50        }
51
52        for(int i=0;i<contadorlineas;i++){
53            if(i==0){
54                fprintf(archivosalida,"%s\n",listaarchivo[i]);
55            }
56            else{
57                fprintf(archivosalida,"%s",listaarchivo[i]);
58            }
59        }
60        return 0;
61
62    }
```

Al final, en el cierre del programa usamos dos for para invertir las lineas del texto de atrás hacia adelante, pero se usaron para retroceder sin invertir los caracteres, solo las líneas, al final se manda imprimir el archivo en el orden que deseamos.

Corrimos el programa y nos pidió dos archivos.

Nos mostró el número de líneas, palabras y caracteres.

Primero nos mostro el numero de palabras, caracteres y líneas, después mostró el texto original escrito al revés

CONCLUSIÓN:

Los archivos de texto plano para la resolución de problemas son de vital importancia, aprenderlos en C es importante porque así podemos editar textos planos, leerlos, y recibir datos de importancia e interpretarlos como sea nuestra conveniencia desde un programa en C, es una excelente herramienta para leer dentro de códigos fuentes ya que la mayoría de éstos se escriben en texto plano.

Con la librería string podemos leer dentro del archivo que abrimos en C, también es importante guardar el texto que ocupamos en el texto en otra variable para poder utilizarlo después dependiendo del uso que le queramos dar, lo importante al momento de usar estos programas es que al escribirlos usamos el mismo lenguaje en C con “if”, “for”, “while”, “do while”, etc. Gracias a los fundamentos que aprendimos podemos utilizar este tipo de herramientas de edición de archivos.