



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcon

Asignatura: Fundamentos de programación

Grupo: 3

No de Práctica(s): 10

Integrante(s): Rivera Sosa Arlethe

Yuan Xiaojing/

No. de Equipo de cómputo empleado:

No. de Lista o Brigada: 317083033

317693612

Semestre: 2020-1

Fecha de entrega: 11/noviembre/19

Observaciones:

Recuerda que los números en las prácticas son solo ejemplos, en tus programas siempre debes cambiar las cosas según los casos, el fscanf necesitaba poder leer más que "8" caracteres, eso es muy poco para una palabra, debía corresponder con tu longitud de lista (20)

CALIFICACIÓN: 10

Textos planos

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 se utiliza para realizar operaciones es 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`.

Fopen: esta función sirve para abrir y crear ficheros en disco. El prototipo correspondiente de `fopen` es:

```
FILE * fopen (const char *filename, const char *opentype);
```

Los parámetros de entrada de `fopen` son:

- `filename`: una cadena que contiene un nombre de fichero válido.
- `opentype`: especifica el tipo de fichero que se abrirá o se creará.

Una 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 sobre escribe 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 sobre escribe 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.

Feof: esta función sirve para determinar si el cursor dentro del archivo encontró el final (end of file). Existe otra forma de verificar el final del archivo que es comparar el caracter que trae `fgetc` del archivo con el macro `EOF` declarado dentro de `stdio.h`, pero este método no ofrece la misma seguridad (en especial al tratar con los archivos "binarios"). La función `feof` siempre devolverá cero (Falso) si no es encontrado `EOF` en el archivo, de lo contrario regresará un valor distinto de cero (Verdadero).

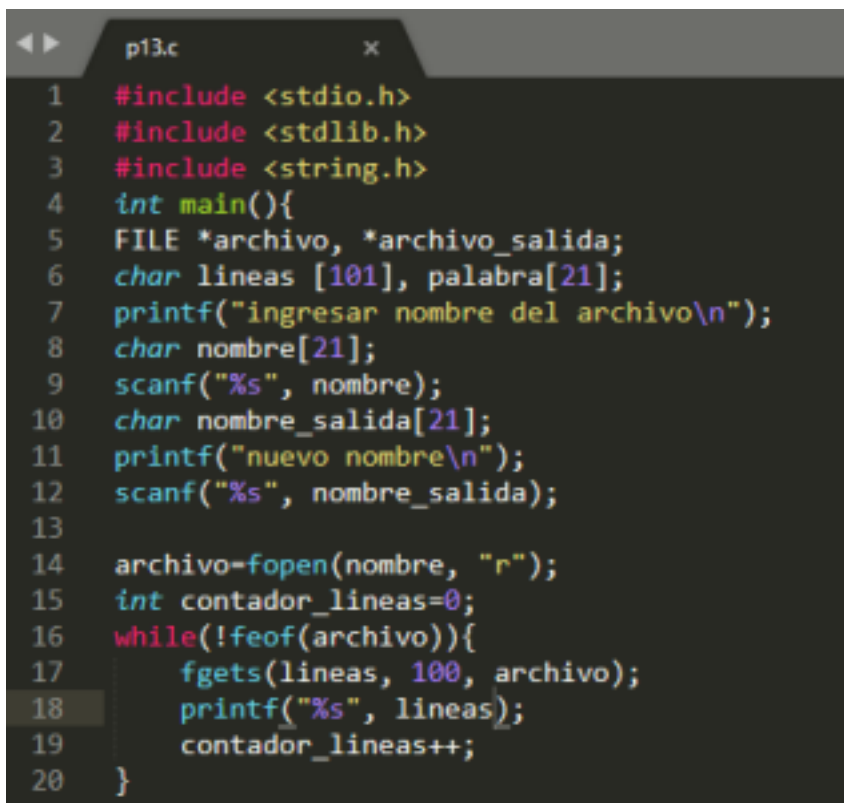
Objetivo: Elaborar programas en lenguaje C que requieran el uso de archivos de texto plano en la resolución de problemas, entendiendo a los archivos como un elemento de almacenamiento secundario.

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 <stdlib.h>
3 #include <string.h>
4 int main(){
5 FILE *archivo, *archivo_salida;
6 char lineas [101], palabra[21];
7 printf("ingresar nombre del archivo\n");
8 char nombre[21];
9 scanf("%s", nombre);
10 char nombre_salida[21];
11 printf("nuevo nombre\n");
12 scanf("%s", nombre_salida);
13
14 archivo=fopen(nombre, "r");
15 int contador_lineas=0;
16 while(!feof(archivo)){
17 fgets(lineas, 100, archivo);
18 printf("%s", lineas);
19 contador_lineas++;
20 }
```

Al hacer nuestro programa, vamos a usar los apuntadores, para transferir información directa y nos pueda abrir un archivo.

Primero hacemos que lea el número de líneas que hay en el texto, para que al finalizar no lo imprima.

```

printf("\ncantidad de líneas: %i\n", contador_lineas);
int contador_palabras=0;
archivo=fopen(nombre, "r");
while(!feof(archivo)){
    fscanf(archivo, "%8s", palabra);
    contador_palabras++;
}

printf("numero de palabras: %i\n", contador_palabras);
archivo=fopen(nombre, "r");
int contadores=0, letras;
while(!feof(archivo)){
    fscanf(archivo, "%8s", palabra);
    letras=strlen(palabra);
    contadores=contadores+letras;
}

```

Ahor en lugar de líneas nos va a contar las palabras y le ponemos "%8s" para que no nos cuente el \0, así nos limita la cantidad de caracteres al leer.

```

printf("numero de caracateres %i\n", contadores);
archivo=fopen(nombre, "r");
archivo_salida=fopen(nombre_salida, "w");
char lista[contador_lineas][100];
for(int i=contador_lineas-1; i!=-1; i--){
    fgets(lista[i],100,archivo);
}

for(int i=0; i<contador_lineas; i++){
    if(i==0){
        fprintf(archivo_salida, "%s\n", lista[i]);
    }
    else{
        fprintf(archivo_salida, "%s", lista[i]);
    }
}
return 0;
}

```

Ahora como va a contar caracteres, quitamos cualquier cosa que lo pueda limitar, contando el "\0" pues ahí es en donde acaba el texto.

Después hacemos que nos imprima los valores. Al momento de compilarlo hay que tener el archivo en donde se va a trabajar y el programa en el mismo lugar o en la misma carpeta, pues de no ser así este no correrá exitosamente.

```

Familia@DESKTOP-NE4QAFJ ~
$ gcc pl3.c -o t

Familia@DESKTOP-NE4QAFJ ~
$ ./t
ingresar nombre del archivo
calaverita.txt
nuevo nombre
nuevo.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.

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

```

ingresar nombre del archivo
nuevo.txt
nuevo nombre
nuevo.txt

```

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,
En este mes singular,

```

cantidad de líneas: 56
numero de palabras: 198
numero de caracteres 846

Al compilarlo, nos va a preguntar sobre el nombre del archivo y el nuevo nombre que le queremos poner. Finalmente nos aparecerá el texto tal cual esta en el archivo descargado.

Después al momento de hacer nuevo nuestro archivo esté va a cambiar el orden, haciendo que el texto sea diferente pero teniendo el mismo contenido.

Conclusión: los textos planos como almacenamiento, nos sirven para almacenar una serie de datos, pero poderlas cambiar, ya sea editando sin borrar, usarla solo como lectura y bórrala toda entre otras. En esta práctica aprendimos a como editar sin borrar, ocuparla como lectura y almacenando la información en un nuevo archivo sin borra el anterior.