



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Rodríguez Espino Claudia Ing.

Asignatura: Fundamentos de Programación

Grupo: 1104

No de Práctica(s): 13

Integrante(s): Santa Rosa Ortiz Thelma Jazmín.

*No. de Equipo de
cómputo empleado* 50

Semestre: 1°

Fecha de entrega: 11 de noviembre del 2018

Observaciones:

CALIFICACIÓN: _____

Práctica 13. 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.

Desarrollo:

Un archivo es un conjunto de datos estructurados en una colección de entidades elementales o básicas denominadas registros que son del mismo tipo, pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Lenguaje C permite manejar la entrada y la salida de datos desde o hacia un archivo, respectivamente, a través del uso de la biblioteca de funciones de la cabecera stdio.h.

Apuntador a archivo

Un apuntador a un archivo es un hilo común que unifica el sistema de Entrada/Salida (E/S) con un buffer donde se transportan los datos.

Un apuntador a archivo señala a la información que contiene y define ciertas características sobre él, incluyendo el nombre, el estado y la posición actual del archivo.

Los apuntadores a un archivo se manejan en lenguaje C como variables apuntador de tipo FILE que se define en la cabecera stdio.h. La sintaxis para obtener una variable apuntadora de archivo es la siguiente:

FILE *F;

Abrir archivo

La función fopen() abre una secuencia para que pueda ser utilizada y la asocia a un archivo. Su estructura es la siguiente:

*FILE fopen(char *nombre_archivo, char *modo);

Donde nombre_archivo es un puntero a una cadena de caracteres que representan un nombre válido del archivo y puede incluir una especificación del directorio. La cadena a la que apunta modo determina cómo se abre el archivo.

Existen diferentes modos de apertura de archivos, los cuales se mencionan a continuación, además de que se pueden utilizar más de uno solo:

- ✚ 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.
- ✚ rb: Abre un archivo en modo lectura y binario.
- ✚ wb: Crea un archivo en modo escritura y binario.

Cerrar archivo

La función fclose() cierra una secuencia que fue abierta mediante una llamada a fopen().

Escribe la información que se encuentre en el buffer al disco y realiza un cierre formal del archivo a nivel del sistema operativo.

Un error en el cierre de una secuencia puede generar todo tipo de problemas, incluyendo la pérdida de datos, destrucción de archivos y posibles errores intermitentes en el programa. La firma de esta función es:

```
int fclose(FILE *apArch);
```

Donde apArch es el apuntador al archivo devuelto por la llamada a fopen(). Si se devuelve un valor cero significa que la operación de cierre ha tenido éxito. Generalmente, esta función solo falla cuando un disco se ha retirado antes de tiempo o cuando no queda espacio libre en el mismo.

Funciones fgets y fputs

Las funciones fgets() y fputs() pueden leer y escribir, respectivamente, cadenas sobre los archivos. Las firmas de estas funciones son, respectivamente:

```
char *fgets(char *buffer, int tamaño, FILE *apArch);  
char *fputs(char *buffer, FILE *apArch);
```

La función fputs() permite escribir una cadena en un archivo específico. La función fgets() permite leer una cadena desde el archivo especificado. Esta función lee un renglón a la vez.

Funciones fscanf y fprintf

Las funciones fprintf() y fscanf() se comportan exactamente como printf() (imprimir) y scanf() (leer), excepto que operan sobre archivo. Sus estructuras son:

```
int fprintf(FILE *apArch, char *formato, ...);  
int fscanf(FILE *apArch, char *formato, ...);
```

Donde apArch es un apuntador al archivo devuelto por una llamada a la función fopen(), es decir, fprintf() y fscanf() dirigen sus operaciones de E/S al archivo al que apunta apArch. formato es una cadena que puede incluir texto o especificadores de impresión de variables. En los puntos suspensivos se agregan las variables (si es que existen) cuyos valores se quieren escribir en el archivo.

Funciones fread y fwrite

fread y fwrite son funciones que permiten trabajar con elementos de longitud conocida. fread permite leer uno o varios elementos de la misma longitud a partir de una dirección de memoria determinada (apuntador).

El valor de retorno es el número de elementos (bytes) leídos. Su sintaxis es la siguiente:

```
int fread(void *ap, size_t tam, size_t nelem, FILE *archivo)
```

fwrite permite escribir hacia un archivo uno o varios elementos de la misma longitud almacenados a partir de una dirección de memoria determinada.

El valor de retorno es el número de elementos escritos. Su sintaxis es la siguiente:

```
int fwrite(void *ap, size_t tam, size_t nelem, FILE *archivo)
```

-----AGENDA-----

Agenda.cpp

```
1  #include<stdio.h>
2  #include<string.h>
3  #include<stdlib.h>
4  #include<conio.h>
5
6  void informacion();
7  void archivo();
8
9  int a,x,i;
10 struct datos
11 {
12     char nom[12];
13     char app[16];
14     char apm[16];
15     char direc[101];
16     char tel[15];
17 };
18 struct datos contac[50];
19 FILE *fichero;
20
21 int main()
22 {
23     printf("\n\t\t\tA G E N D A \n\n\n");
24     printf("\tAnota el numero de contactos que registraras: ");
25     scanf("%d",&x);
26     if((fichero=fopen("agenda.txt","w+"))==NULL)
27     {
28         printf("\tNo se puede abrir el archivo.\n");
29     }
30     else
31     {
32         for(a=1;a<=x;a++)
33         {
34             informacion();
35         }
36         archivo();
37         fclose(fichero);
38     }
39     printf("\n\tTus archivos se guardaron\n");
40     system("pause");
41     getch();
42 }
43 void informacion()
44 {
45     printf("\t\tFavor de escribir los datos separados por comas, sin espacios.\n\n");
46     printf("\t\tCONTACTO %d\n",a);
47     printf("\t\tApellido paterno: ");
48     scanf("%s",&contac[a].app);
49     strupr(contac[a].app);
50
51     printf("\t\tApellido materno: ");
52     scanf("%s",&contac[a].apm);
53     strupr(contac[a].apm);
54
55     printf("\t\tNombre: ");
56     scanf("%s",&contac[a].nom);
57     strupr(contac[a].nom);
```

```

}
void informacion()
{
    printf("\t\tFavor de escribir los datos separados por comas, sin espacios.\n\n");
    printf("\t\tCONTACTO %d\n",a);
    printf("\t\tApellido paterno: ");
    scanf("%s",&contac[a].app);
    strupr(contac[a].app);

    printf("\t\tApellido materno: ");
    scanf("%s",&contac[a].apm);
    strupr(contac[a].apm);

    printf("\t\tNombre: ");
    scanf("%s",&contac[a].nom);
    strupr(contac[a].nom);

    printf("\t\tDireccion: ");
    scanf("%s",&contac[a].direc);
    strupr(contac[a].direc);

    printf("\t\tTelefono: ");
    scanf("%s",&contac[a].tel);
    printf("\n");
}

void archivo()
{
    fprintf("\n\t\tA G E N D A \n\n\n");
    for(a=1;a<=x;a++)
    {
        fprintf(fichero,"Nombre %d: %s %s %s\n",a,contac[a].nom,contac[a].app,contac[a].apm);
        fprintf(fichero,"Direccion %d: %s\n",a,contac[a].direc);
        fprintf(fichero,"Telefono %d: %s\n",a,contac[a].tel);
        fprintf(fichero,"\n");
    }
}

```

E:\Programacion\Agenda.exe

```

A G E N D A

Anota el numero de contactos que registraras: 3
Favor de escribir los datos separados por comas, sin espacios.

CONTACTO 1
Apellido paterno: Ortiz
Apellido materno: Sandoval
Nombre: Angelica
Direccion: CANOSAS;MZ14;LT33;CS1;HEROES;COACALCO
Telefono: 5562314665

Favor de escribir los datos separados por comas, sin espacios.

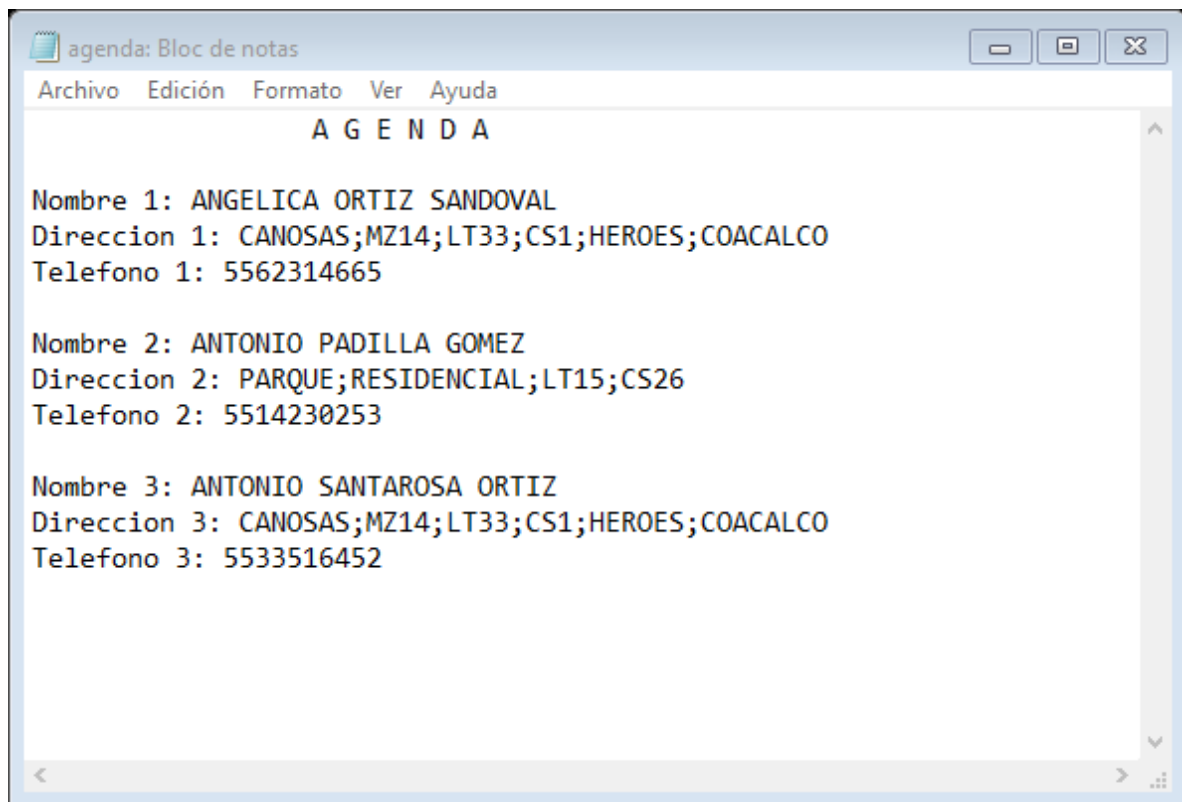
CONTACTO 2
Apellido paterno: Padilla
Apellido materno: Gomez
Nombre: Antonio
Direccion: PARQUE;RESIDENCIAL;LT15;CS26
Telefono: 5514230253

Favor de escribir los datos separados por comas, sin espacios.

CONTACTO 3
Apellido paterno: SantaRosa
Apellido materno: Ortiz
Nombre: Antonio
Direccion: CANOSAS;MZ14;LT33;CS1;HEROES;COACALCO
Telefono: 5533516452

Tus archivos se guardaron
Presione una tecla para continuar . . .

```



Conclusión:

Gracias a esta práctica, aprendimos a elaborar programas con el lenguaje C que requerían el uso de archivos de texto plano para la resolución de un problema, y entendimos los archivos como un elemento de almacenamiento secundario.