

ÍNDICE

8	Manejo de Archivos	138
8.1	Declaración y apertura de un archivo	138
8.2	Cierre de un archivo	139
8.3	Lectura de información de un archivo	140
8.4	Escritura de información en un archivo	141
8.5	Ejercicio del manejo de archivo mediante el uso de funciones.	142

Universidad Autónoma Metropolitana – Iztapalapa	Trimestre: 08-I
Curso de Introducción a la Programación	Capítulo 8

8 Manejo de Archivos

En la utilización un archivo requiere del establecimiento de un canal de comunicación entre el programa y el archivo, este canal de comunicación se reconoce como *flujo (stream)*. Al momento de obtener la información del archivo a traves del flujo los datos están pasando por dicho canal de comunicación para poder ser leídos por el programa. Lo mismo sucede cuando se desea escribir información del programa a un archivo, todos los datos pasan por dicho canal de comunicación.

Para el manejo de archivos se requiere incluir la librería **stdlib.h** la cual contiene funciones y constantes para el manejo de archivos, algunas de estas constantes, estructura y funciones serán utilizadas a lo largo de este capítulo.

Dentro del programa los archivos tendrán un nombre (identificador) con el cual se hará referencia, dicho nombre deberá ser declarado a un tipo de dato en especifico, que es en si un *puntero a archivo* (véase referencia) que contiene información referente al archivo con al cual se desea trabajar. El tipo de dato con el cual se maneja a los archivos es **FILE**.

8.1 Declaración y apertura de un archivo

Antes de empezar a trabajar con un archivo es necesario abrirlo, pero para abrirlo se requiere verificar que realmente se encuentre archivo en la ruta proporcionada y además que se haga referencia a un nombre con su extensión.

Definición 17: Estructura de declaración y apertura de un tipo de dato **Archivo (FILE)**:

Pseudocódigo
<pre>//La declaración de un tipo de dato Archivo no esta definida en Pseudocódigo. //Apertura de un tipo de dato Archivo. // Abrir ("<ruta y nombre del archivo>") Abrir("C:\arch.txt")</pre>
Lenguaje C
<pre>//Declaración de un tipo de dato Archivo FILE *archivo; //Apertura de una Archivo // fopen (nombre_archivo, modo_apertura) fopen ("C:\\arch.txt", "r");</pre>

Ejemplo: “Apertura con Validación de un Archivo”

```
/*
 Archivo: Archivos.c .....
*/

/*Incluir librerias*/
#include <stdio.h>
#include <stdlib.h>

/**Definición de constantes*/
#define TAMCADENA 30 //Tamaño máximo de ruta y nombre del archivo

int main ()
{
    FILE *arch;
    //Utilice doble diagonal para representar una \ en C
```

```

char ruta_nombre[] = "C:\\PWS.txt";
//Apertura del Archivo
arch = fopen(ruta_nombre, "r");

//Verificación de apertura
if(arch == NULL )
{
    printf("\n\nError al abrir el archivo %s ", ruta_nombre);
    getch();
    exit( -1); //Finalizar el programa
}
else //No es necesario 'otro caso'
{
    printf("\n\nOK archivo %s abierto!!!!", ruta_nombre);
}
getch();
return (0);
}

```

Al utilizar la función *fopen()* se requiere del uso de dos parámetros:

- Una cadena que indica la ruta, nombre y extensión del archivo a abrir.
- El modo de apertura del archivo.

El modo de apertura indica si el archivo será de lectura, escritura o se agregará información al final del archivo. Además es posible indicar si dicho archivo será de texto o binario [X].

Definición 18: Modos de apertura de un archivo:

Modo	Significado
"r"	Abre el archivo para <i>lectura</i> .
"w"	Crea y abre un nuevo archivo, si ya existía se <i>pierde</i> la información.
"a"	Abre el archivo <i>existente</i> para escribir al <i>final</i> .
"r+"	Abre archivo ya existente para <i>leer o escribir</i> la información.
"w+"	Crea un archivo para <i>leer o escribir</i> , si ya existía se <i>pierde</i> la información.
"a+"	Abre el archivo <i>existente</i> para <i>leer o escribir</i> al <i>final</i> .

Los modos de archivo para abrir un archivo que se muestran en la definición 18, son utilizados para abrir texto por defecto; sin embargo es posible utilizar el carácter 't' o 'b' para indicar que se trata de un archivo de texto o un archivo binario, respectivamente.

- Modos para apertura de un archivo de *texto*:
"rt", "wt", "at", "r+t", "w+t", "a+t"
- Modos para apertura de un archivo *binario*:
"rb", "wb", "ab", "r+b", "w+b", "a+b"

8.2 Cierre de un archivo

Una vez que se haya trabajado con el archivo y ya no se desee trabajar mas con ellos, es importante que se cierre el archivo indicando así que ya no requiere que se este pasando información por el flujo de datos.

Definición 19: Cierre de un Archivo:

Pseudocódigo
//Cierre de un tipo de dato Archivo. // Cerrar (<identificador_del_archivo>) Cerrar (arch)
Lenguaje C
//Cierre de una Archivo // fclose (identificador_del_archivo) fclose (archivo);

Universidad Autónoma Metropolitana – Iztapalapa	Trimestre: 08-I
Curso de Introducción a la Programación	Capítulo 8

8.3 Lectura de información de un archivo

Cuando se esta efectuando la lectura de la información de un archivo el compilador requiere de un indicador que le informe que se ha llegado al final del archivo y que ya no hay más datos a leer, este indicador es conocido como al *Condición de Fin De Archivo (End Of File - EOF)*.

Definición 20: Lectura de caracteres desde un Archivo utilizando *fgetc()*:

Pseudocódigo
<pre> caract ← LeeC(archivo) //lectura continua caract = LeeC(archivo) Mientras (caract != FIN) Comienza caract = LeeC(archivo) Termina </pre>
Lenguaje C
<pre> //Declaración de un tipo de dato Archivo FILE *archivo; char caract; //Apertura..... //Lectura continua //Se lee el primer caracter carac = fgetc(archivo); while (carac != EOF) { ///Otras instrucciones carac = fgetc(archivo); } </pre>

Ejemplo: “Lectura de caracteres con *fgetc()*”

En este ejemplo se indica como se recibe como parámetro un tipo de dato FILE, que previamente fue abierto. Primero se obtendrá el primer carácter utilizando la función *fgetc()* y posteriormente se hará uso de un ciclo iterativo *while* para ir verificando que no se haya llegado al fin del archivo (EOF).

```

/*
*/
void LecturaArchivoCarac(FILE *archivo)
{
    char carac;

    //Se lee el primer caracter
    carac = fgetc( archivo);
    printf("%c",carac);
    //Lectura continua
    while ( carac != EOF )
    {
        ///Otras instrucciones
        carac = fgetc( archivo);
        printf("%c",carac);
    }
}

```

Universidad Autónoma Metropolitana – Iztapalapa	Trimestre: 08-I
Curso de Introducción a la Programación	Capítulo 8

8.4 Escritura de información en un archivo

Cuando se esta efectuando la escritura de información a un archivo previamente abierto, dicha información se agregará al final del archivo siempre y cuando el modo de apertura se le indica que será de escritura.

Definición 21: Escritura de caracteres en un Archivo utilizando *fputs* ():

Pseudocódigo
EscribirC (caract , archivo)
Lenguaje C
<pre>//Declaración de un tipo de dato Archivo FILE *archivo; char caract; //Apertura..... //Escritura de un carácter en el archivo fputc(caract, archivo);</pre>

Ejemplo: “Lectura de caracteres con *fgetc*()”

En este ejemplo se indica como se recibe como parámetro un tipo de dato FILE, que previamente fue abierto con modo de escritura. Primero se le pedira al usuario que teclee una cadena, y al mismo tiempo que el usuario vaya tecleando carácter por carácter, el programa ira escribiendolo en el archivo.

```
/*
*/
void EscrituraConsolaArchivoCarac(FILE *archivo)
{
    char caract = '\0';

    printf("\n\nTeclee una cadena: \n");

    //Escritura de caracteres desde teclado
    while ( caract != '\n')
    {
        //Lectura de caracteres desde consola
        caract = getchar();

        //Guardar carácter al archivo
        fputc(caract, archivo);
    }
}
```

Universidad Autónoma Metropolitana – Iztapalapa	Trimestre: 08-I
Curso de Introducción a la Programación	Capítulo 8

8.5 Ejercicio del manejo de archivo mediante el uso de funciones.

El siguiente programa ilustra como se hace el manejo de archivos, considerando:

- Apertura.
- Cierre
- Lectura de información.
- Escritura de información.
- Obtención de la posición del apuntador en el archivo.
- Regresar el apuntado al inicio del archivo.

Para poder ejecutar el programa se requiere de algunas especificaciones las cuales se indican a continuación.

Actividad

Dentro de la función principal del programa existen líneas que están inicialmente documentadas, dichas líneas tienen un numero que hace referencia a puntos específicos del programa.

La actividad es seguir las instrucciones de cada uno de los pasos que se indican a continuación.

Paso	Especificaciones
0	Cree un archivo con el nombre de "PREUBA2.txt" en donde se genere el ejecutable, ábralo y escriba información
1	Compile y ejecute el programa, dentro de la función main existen líneas documentadas es necesario que se queden por el momento así.
2	Descomente las dos líneas del punto 3, compile y ejecute. Vea el resultado en el archivo.
3	Comente las dos líneas del punto 3, y descomente las líneas del punto 4, guarde, compile y ejecuta, vea lo que sucede con el archivo. Si se desea agregar información al final del archivo (txt) cambie el modo de apertura. Experimente!!!
4	Comente las dos líneas del punto 4 del programa, después descomente las dos líneas del punto 5, guarde compile, ejecute y vea los resultado en consola.
5	Comente las dos líneas del punto 5 del programa, después descomente las dos líneas del punto 6, guarde compile, ejecute y vea los resultado en consola. Le tiene que mandar a imprimir la posición 0.
6	Descomente las dos líneas del punto 5, no comente las del punto 6, guarde, compile, ejecute y vea los resultado: Ahora el programa regresara un valor distinto a cero, abra el archivo y cuente los caracteres que hay en el archivo para verificar.
7	Comente SOLO las dos líneas del punto 6 del programa, después descomente las dos ultimas líneas del punto 7 (printf.. y Lectura...), guarde, compile, ejecute y vea los resultado en consola. El programa despliega la información del archivo, después vuelve a tratar de leer el contenido pero el apuntador a archivo se encuentra al final de este y por esta razón no se logra ver nada de información. Ahora descomente la primera línea del punto 7 (RegresarAr...) guarde, compile, ejecute y vea los resultado en consola. El programa ahora desplegara dos veces la información completa del archivo.
8	Documente las dos líneas del punto 5 y las tres líneas del punto 7. Descomente las 5 líneas del punto 8, guarde, compile, ejecute y vea los resultado en consola. Abra el archivo y verifique que se hayan escrito nueva información.
9	Documente las 5 líneas del punto 8, descomente las 5 líneas del punto 9. Guarde, compile, ejecute y vea los resultados en consola. El programa ahora desplegara un mensaje de texto con millas dobles y al inicio con un contador incrementado. Abra el archivo (txt) y verifique que la información de pantalla este en el archivo.

Universidad Autónoma Metropolitana – Iztapalapa	Trimestre: 08-I
Curso de Introducción a la Programación	Capítulo 8

Código Proporcionado

NOTA: Complete la documentación que sea necesaria.

```

/*
  Archivo: ArchivosFunciones.c
  Descripción: Este programa permite el manejo de archivos
  Autor: Basurto Páez Gustavo
  Fecha: 28 de mayo de 2008
*/

/*Incluir librerías*/
#include <stdio.h>
#include <stdlib.h>

/**Definicion de constantes*/
#define TAMCADENA 30 //Tamano de ruta y nombre del archivo

//prototipos
FILE *AperuraArchivo (char ruta_nombre[], char modo[]);
void CierreArchivo(FILE *arch);
void LecturaArchivoCarac(FILE *archivo);
void EscrituraConsolaArchivoCarac(FILE *archivo);
void LecturaArchivoFromato(FILE *archivo);
void LecturaArchivoCadena(FILE *archivo);
void RegresarArchivo (FILE *arch);
int ObtenerPosArchivo(FILE *arch);
void EscrituraArchivoCadena(FILE *archivo);
void EscrituraArchivoCadenaFormato(FILE *archivo);

int main ()
{
    int pos =0;

    //Archivos
    FILE *archLectura;
    FILE *archEscritura;

    //Utilice doble diagonal para represntar una \ en C
    //char ruta_nombre[] = "C:\\PRUEBA.txt";
    //char ruta_nombre[] = "C:\\PREUBA2.txt";

    //Utilizar si el archivo se encuentra en donde se ejecuta el programa
    char ruta_nombre[] = "PREUBA2.txt";

    //1.- Archivo de lectura
    printf("\n\n Apertura del Archivo\n\n");
    archLectura = AperuraArchivo (ruta_nombre , "r+");

    //3.-Lectura
    //printf("\n\n Lectura del contenido del Archivo con fgetc().\n\n");
    //printf("\n\n Presione enter para ver siguiente caracter.\n\n");
    //LecturaArchivoCarac(archLectura);

    //4.-Escritura
    //printf("\n\n Escritura de un archivo con putc()\n\n");
    //EscrituraConsolaArchivoCarac(archLectura);

    //5.-Lectura
    //printf("\n\n Lectura del contenido de un archivo con fgets ()\n\n");
    //LecturaArchivoCadena(archLectura);

    //6.- Obtener posicion del archivo

```

Universidad Autónoma Metropolitana – Iztapalapa	Trimestre: 08-I
Curso de Introducción a la Programación	Capítulo 8

```

//pos = ObtenerPosArchivo(archLectura);
//printf("\n\n Posicion del archivo es: %d \n\n", pos);

//7.- rewind
//RegresarArchivo (archLectura);
//printf("\n\n Lectura del contenido de un archivo con fgets ()\n\n");
//LecturaArchivoCadena(archLectura);

//8.-Escribir con fputs
//printf("\n\n Escritura de un archivo con fputs()\n\n");
//EscrituraArchivoCadena(archLectura);
//RegresarArchivo (archLectura);
//printf("\n\n Lectura del contenido de un archivo con fgets ()\n\n");
//LecturaArchivoCadena(archLectura);

//9.-Escribir con formato printf
//printf("\n\n Escritura de un archivo con fprintf()\n\n");
//EscrituraArchivoCadenaFormato(archLectura);
//RegresarArchivo (archLectura);
//printf("\n\n Lectura del contenido de un archivo con fgets ()\n\n");
//LecturaArchivoCadena(archLectura);

//2.- Cierre
printf("\n\n Cerrar Archivo \n\n");
CerrarArchivo(archLectura);

getch();
return (0);
}

/*
* Descripcion:
* Entrada:
* Salida:
*/
FILE *AperuraArchivo (char ruta_nombre[], char modo[])
{
    FILE *arch;
    arch = fopen(ruta_nombre, modo);
    if(arch == NULL )
    {
        printf("\n\nErrores al abrir el archivo %s ", ruta_nombre);
        getch();
        exit(-1);
    }
    return( arch );
}

/*
*/
void CierreArchivo(FILE *arch)
{
    fclose (arch);
}

/*
*/
void LecturaArchivoCarac(FILE *archivo)
{
    char carac;

    carac = fgetc( archivo);
    printf("%c", carac);
    //Lectura continua
    while ( carac != EOF )
    {

```

Universidad Autónoma Metropolitana – Iztapalapa	Trimestre: 08-I
Curso de Introducción a la Programación	Capítulo 8

```

    ///Otras instrucciones
        carac = fgetc( archivo);
        printf("%c",carac);
        //getch para ir pausando el programa
        getch();
    }
}

/*
*/
void EscrituraConsolaArchivoCarac(FILE *archivo)
{
    char carac = '\0';

    printf("\n\nTeclee una cadena (finalice con enter): \n");

    //Lectura continua
    while ( carac != '\n')
    {
        //Lectura de caracteres
        carac = getchar();
        //A archivo
        fputc(carac, archivo);
    }
}

/*
*/
void LecturaArchivoFormato(FILE *archivo)
{
    char cadena [TAMCADENA];

    //fscanf(archivo, "%s", &cadena);
    //printf("%s", cadena);

    //Lectura continua
    while ( !feof(archivo) )
    {
        //Otras instrucciones
        fscanf(archivo, "%s", &cadena);
        printf("%s", cadena);
        getch();
    }
}

/*
*/
void LecturaArchivoCadena(FILE *archivo)
{
    char cadena[TAMCADENA];
    char *prueba; //puntero

    prueba = fgets(cadena, TAMCADENA, archivo);

    //Lectura continua
    while ( prueba != NULL )
    {
        //printf("%s-%s*",prueba, cadena );
        printf("%s",cadena );
        prueba = fgets(cadena, TAMCADENA, archivo);
    }
}

/*
*/

```

Universidad Autónoma Metropolitana – Iztapalapa	Trimestre: 08-I
Curso de Introducción a la Programación	Capítulo 8

```

* Descripcion:
* Entrada:
* Salida:
*/
void RegresarArchivo (FILE *arch)
{
    fpos_t comienzo=0;
    fsetpos(arch, &comienzo);
}

/*
*/
int ObtenerPosArchivo(FILE *arch)
{
    int pos = 0 ;
    fpos_t posicion=0;

    fgetpos( arch, &posicion );
    pos = posicion;
    return (pos);
    //(int)
    //fgetpos( archLectura, &posicion );
}

void EscrituraArchivoCadena(FILE *archivo)
{
    char cadena[] = "Hola Mundo con .\t. Archivos - fputs()\n";
    int cont =0;
    for(cont =0; cont <10; cont++)
    {
        fputs( cadena, archivo);
    }
}

void EscrituraArchivoCadenaFormato(FILE *archivo)
{
    char cadena[] = "\tHola Mundo con\t. Archivos - fprintf()\n";
    int cont =0;
    for(cont =0; cont <10; cont++)
    {
        //printf("%d - : %s \n", cont, cadena);
        fprintf(archivo, "%d - : %s", cont, cadena);
    }
}

```