UNIVERSIDAD NACIONAL DE ASUNCIÓN

FACULTAD POLITÉCNICA

Construyendo el futuro

Archivos

Programación

Objetivos

- Conocer la utilidad de los archivos dentro de un programa.
- Utilizar las funciones para abrir un archivo en diferentes modos.
- Utilizar funciones para lectura y escritura de archivos, determinar el final del archivo y la función para cerrar archivos.
- Conocer la manera de acceder al archivo de manera secuencial y directa.
- Escribir programas en Lenguaje C utilizando archivos.

Contenido de la clase

- Introducción
- Apertura de un archivo
- Cierre de un archivo
- Control de final de un archivo
- Escritura y lectura de un archivo
- Acceso secuencial
- Acceso directo

Introducción

Existen casos donde la entrada de datos desde el teclado no es suficiente para cumplir con las exigencias del problema planteado. Al igual que las entradas, en muchas ocasiones también existe la necesidad de almacenar de manera permanente el resultado arrojado por un programa y no simplemente imprimir el resultado en pantalla.

Para estos casos será necesario la utilización de archivos por parte del programa.

Funciones necesarias

Las funciones para le manejo de archivos se encuentran en el archivo de cabecera **stdio.h** Por lo general estas funciones empiezan con **f** haciendo referencia a *file*.

Existe un tipo **FILE** que se usará como apuntador a la información del archivo.

Secuencia para el manejo de archivos

- Crear el puntero del tipo **FILE**.
- Abrir el archivo utilizando la función **fopen**. El resultado de la función tiene que ser almacenado en el puntero al archivo.
- Realizar las operaciones (lectura, escritura).
- Cerrar el archivo utilizando la función **fclose**.

Función fopen

Esta función sirve para abrir o crear el archivo en disco.

Prototipo

FILE * fopen (const char *nombreArchivo, const char *tipoApertura);

Parámetros

nombreArchivo: una cadena que contiene un nombre de archivo.

tipoApertura: especifica el modo de apertura del archivo.

Función fopen

Modos de apertura

Modo	Descripción
r	Abre un archivo para lectura.
w	Crea un archivo para escritura. Si el archivo ya existe, descarta el contenido actual.
a	Agrega; abre o crea un archivo para escritura al final del archivo.
r+	Abre un archivo para actualización (lectura y escritura).
w+	Crea un archivo para actualización. Si el archivo ya existe, descarta el contenido actual.
a+	Agrega; abre o crea un archivo para actualización; la escritura se hace al final del archivo.

Función fopen

Modos de apertura (cont)

Modo	Descripción
rb	Abre un archivo para lectura en modo binario.
wb	Crea un archivo para escritura en modo binario. Si el archivo ya existe, descarta el contenido actual.
ab	Agrega; abre o crea un archivo para escritura al final del archivo en modo binario.
rb+	Abre un archivo para actualización (lectura y escritura) en modo binario.
wb+	Crea un archivo para actualización en modo binario. Si el archivo ya existe, descarta el contenido actual.
ab+	Agrega; abre o crea un archivo para actualización en modo binario; la escritura se hace al final del archivo.

Función fopen

Verificar apertura

Es necesario verificar si el archivo se pudo abrir antes de usarlo.
La siguiente sentencia permite verificar si el archivo se pudo abrir:

```
FILE *fp;
fp = fopen ( “datos.txt", "r" );
if (fp == NULL){
    // El archivo no se pudo abrir.
    // Puntero nulo
}
```

Función fclose

Esta función sirve para cerrar un archivo abierto.

Prototipo

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

Un valor de retorno cero indica que el archivo ha sido correctamente cerrado, si ha habido algún error, el valor de retorno es la constante **EOF**.

Ver ejemplo: AbrirCerrarArchivo.c

Función feof

Esta función sirve para determinar si el cursor dentro del archivo encontró el final.

Prototipo

```
int feof(FILE *archivo);
```

Valores devueltos por la función:

- 0: (falso) si el archivo no está en el final
- 1: (verdadero) si el archivo está en el final.

Lectura de archivos

Un archivo se puede ver como una cadena de caracteres que está guardado en el disco.

Las siguientes funciones pueden usarse para leer los datos del archivo:

- `char fgetc(FILE *archivo)`
- `char *fgets(char *buffer, int tamano, FILE *archivo)`
- `int fscanf(FILE *fichero, const char *formato, argumento, ...);`

Función fgetc

Esta función lee un carácter a la vez del archivo que esta siendo señalado con el puntero *archivo. En caso de que la lectura sea exitosa devuelve el carácter leído y en caso de que no lo sea o de encontrar el final del archivo devuelve EOF.

Prototipo

```
char fgetc(FILE *archivo);
```

Ver ejemplo: LecturaArchivo1.c

Función fgets

•Esta función lee cadena de caracteres.
•Leerá hasta n-1 caracteres o hasta un cambio de línea ‘\n’ o hasta el final del archivo EOF. En este caso, el carácter de cambio de línea '\n' también es leído.

Prototipo

```
char * fgets(char *vector, int tamaño, FILE *archivo);
```

Parámetros

vector: variable en la cual será almacenado lo leído.
tamaño: tamaño límite de caracteres a ser leídos por la función. Se leerá hasta el límite de caracteres o hasta que se encuentre el final de la línea (\n) o el final del archivo (EOF).
archivo: archivo de donde se leerá los datos.

El beneficio de esta función es que se puede obtener una línea completa a la vez.

Ver ejemplo: LecturaArchivo2.c

Función fscanf

La función fscanf funciona igual que scanf en cuanto a parámetros, pero la entrada se toma de un fichero en lugar del teclado.

Prototipo

int fscanf(FILE *archivo, const char *formato, argumento, ...)

Parámetros

archivo: archivo desde donde serán leídos los datos.
formato: formato de los datos a leer. Ejemplo: %d leerá un entero.
argumento: variables en donde serán almacenados los valores leídos.

Ver ejemplo: LecturaArchivo3.c

Ejercicio a resolver

Escribir un programa que divida el contenido de un archivo en n partes.