

PROGRAMACIÓN AVANZADA

Archivos de Texto

ARCHIVOS DE TEXTO



Lo son todos los archivos que podemos crear usando un editor o procesador de textos.

Por ejemplo, archivos con extensiones *.txt* o *.doc*

Por simplicidad, solo estudiaremos los archivos con extensión *.txt* y *.csv*

CARACTERÍSTICAS DE LOS ARCHIVOS DE TEXTO



Un archivo de texto es una colección de caracteres guardados en un almacenamiento secundario (por ejemplo, en un disco).



Para marcar el final de un archivo de texto, la computadora coloca un caracter especial de fin de archivo: `<eof>`.



Al crear un archivo de texto con un editor, al presionar la tecla `<enter>` se crea el caracter de nueva línea (representado en C como `'\n'`).

EJEMPLO

Archivo.txt:

Esto es un archivo de texto<*newline*>

Tiene dos líneas<*eof*>

Cada línea termina con el caracter <*newline*>, y con el caracter <*eof*> se termina el archivo.

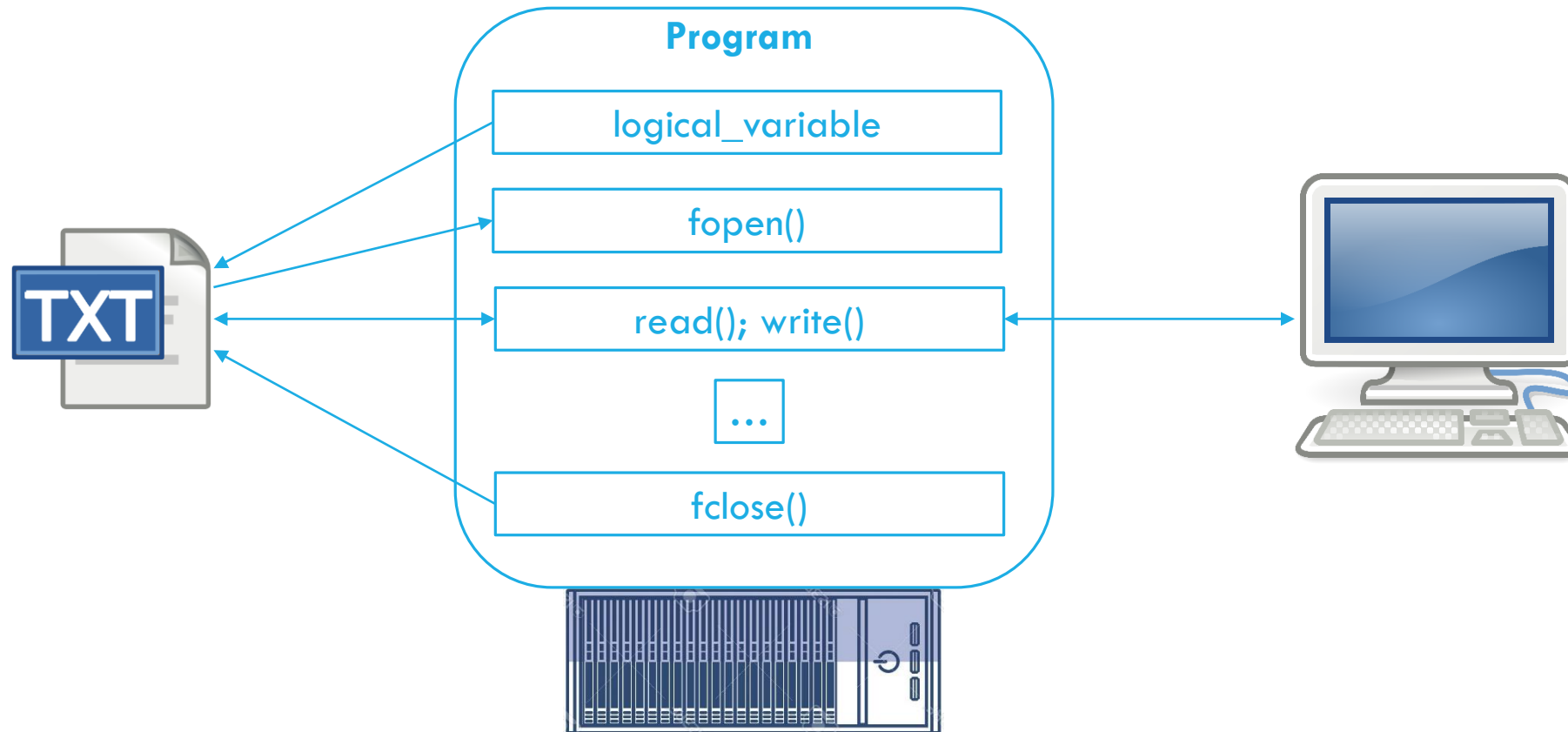
En realidad, el texto se almacena como una sola cadena incluyendo los caracteres especiales. Este principio también se utiliza cuando entramos datos por teclado y o los mostramos en pantalla.

PUNTEROS A ARCHIVOS DE TEXTOS

Aunque no hemos estudiado en qué consisten los “punteros”, vamos a utilizarlos para manipular el tipo de dato archivo.

Con relación a los punteros, por ahora es suficiente con mencionar que son una herramienta para acceder directamente a los datos almacenados en memoria a partir de su dirección. Se denotan con el caracter *.

PROGRAMACIÓN DE ARCHIVOS DE TEXTO



EJEMPLO: REALICE UNA COPIA DE UN ARCHIVO DE TEXTO

```
FILE *datos;
FILE *resultado;
datos = fopen("Archivo.txt", "r");
resultado = fopen("Salida.txt", "w");
char c = getc(datos);
while (c != EOF) {
    putc(c, resultado);
    c = getc(datos);
}
fclose(datos);
fclose(resultado);
```

Declaración

Preparación: "r" para leer,
"w" para escribir

Leer un caracter

Preguntar si se llegó al final de la línea

Escribir un caracter

Cerrar los archivos (por precaución)

ALGUNAS FUNCIONES EN `<STDIO.H>` PARA MANIPULAR ARCHIVOS TEXTOS

Funciones	Descripción
fopen	Abre un archivo
fclose	Cierra un archivo
getc	Lee un carácter
putc	Escribe un carácter
fgets	Lee una línea
fputs	Escribe una línea
fscanf	Lee un entero con 'formato'

LIBRERÍA <STRING.H>

Como en los archivos de texto se escriben normalmente cadenas de caracteres, muchas veces es mejor usar el tipo de dato *string* que el tipo de dato *char*, para no tener que procesar letra a letra las palabras u oraciones.

Para ello, es útil usar diferentes funciones de la librería <string.h>. En este ejercicio usaremos las funciones *gets* y *strcmp*.

FUNCIÓN *GETS*

La función **gets** lee una entrada de teclado y la **almacena** en una variable como **string**. Es similar a **scanf**, pero **solo** almacena cadenas.

Ejemplo:

```
char cadena[20];  
gets(cadena);
```

Si escribimos por teclado “Una palabra”, y **<enter>**:

cadena ←

U	n	a		p	a	l	a	b	r	a	\0							
---	---	---	--	---	---	---	---	---	---	---	----	--	--	--	--	--	--	--

DIFERENCIAS ENTRE *GETS* Y *SCANF*

scanf lee la entrada hasta que encuentra espacios en blanco, nueva línea o EOF; mientras que ***gets*** lee la entrada hasta que encuentra nueva línea o EOF.

Por lo tanto, la función ***scanf*** elimina los espacios al leer una cadena.

En el ejemplo anterior, si usamos:

```
scanf("%s", &cadena);
```

en la variable se almacenaría:

```
cadena == "Una"
```

FUNCIÓN *STRCMP*

La función ***strcmp*** **compara** dos cadenas que se le pasan como parámetros y **devuelve** verdadero si son idénticas o falso si no.

Ejemplo, continuando del **anterior**:

`strcmp("Una palabra", cadena) == 0` → Lo que significa que ambas cadenas son **iguales**.

ARCHIVOS *CSV*

Los archivos CSV (del inglés comma-separated values) son documentos en formato sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas y las filas por saltos de línea.

Estos archivos también se consideran Archivos de Texto, con las características antes mencionadas.

Varios programas nos permiten guardar los documentos en este formato, por ejemplo: Word, Excel, entre otros. En estos casos, los documentos se cambian a formato texto simple.

ACTIVIDAD

Realice un programa en C que permita crear un archivo con nombre “Estudiantes.csv” donde se almacenarán las calificaciones de diferentes estudiantes. El programa debe desplegar un menú con las siguientes opciones:

1. Crear Archivo con encabezado (Nombre, Carrera, Matemática, Física, Química).
2. Mostrar el contenido del archivo.
3. Agregar los datos de un nuevo estudiante al final del archivo.
4. Calcule la nota promedio en la materia Computación (*strtok()* y *atoi()*).
5. Devuelva el nombre del estudiante con la nota más alta en Física.
6. Dada una carrera, cree otro archivo con el nombre de la carrera y copie en él solo los datos de los alumnos que la estudian.
7. Salir.

EJEMPLO STRTOK()

```
#include<stdio.h>
#include <string.h>
int main() {
    char string[50] = "Hello! We are learning about strtok";
    // Extract the first token
    char * token = strtok(string, " "); // “ ” -> delimiter
    // loop through the string to extract all other tokens
    while (token != NULL) {
        printf( " %s\n", token ); //printing each token
        token = strtok(NULL, " "); } }
```

MUCHAS GRACIAS