

PRÁCTICA 3

**PRESENTADO POR:
DANIEL ESTEBAN YEPES PALACIO
DIANA MARCELA LOZADA GARCÍA**

**PROFESOR:
HENRY ARCILA**

**ASIGNATURA:
LABORATORIO DE SISTEMAS OPERATIVOS**

**UNIVERSIDAD DE ANTIOQUIA
FACULTAD DE INGENIERÍA
MEDELLÍN
2012**

PRÁCTICA 3

8. ¿Cuántos caracteres tiene el archivo?

El archivo tiene los siguientes caracteres:

```
0000000 68 6f 6c 61 20 6d 75 6e 64 6f 0a
000000b
```

Son 13 caracteres ASCII en hexadecimal.

9. ¿Qué hace *fopen* y que quiere decir “r”?

fopen es una función de C que sirve para abrir y crear ficheros en disco.

r es un argumento que indica el modo de abrir el archivo, el cual en este caso es de lectura.

¿Qué argumentos necesita?

La función *fopen* necesita como argumentos:

- El nombre o ruta del archivo.
- Opentype como r, w, entre otros, para indicar el modo de abrir el archivo: abrir y leer, abrir y escribir, etc.

Se define de la siguiente manera:

```
fopen(char *fileName, char *opentype)
```

¿Qué retorna la función en caso de éxito y en caso de falla?

En caso de éxito retorna el archivo abierto en el modo indicado, en caso de falla retorna *null*.

10. ¿Qué es EOF?

EOF de sus siglas *End of File*, es un indicador que retorna, en este caso, el indicador correspondiente al fin de archivo, es decir, indica que no pueden ser leídos más datos desde la fuente o archivo.

11. ¿Cuál es la diferencia entre las funciones *fgetc* y *getc*?

fgetc lee un caracter desde un archivo. Esta función tiene como parámetro un archivo y retorna el caracter leído como un char sin signo convertido a un entero, también puede retornar un EOF o error.

getc es equivalente a *fgetc* ya que igualmente lee un carácter pero también los lee desde otras fuentes como el teclado.

12. ¿Por qué el valor que retorna *fgetc* retorna en una variable *int*?

Porque *fgetc* retorna el código ASCII del carácter leído y este código está definido como entero.

13. ¿Qué hace *fputc* o *putc*?

fputc o *putc* Lo que hace esta función es escribir un caracter c recibido en un archivo entrado a la función convirtiéndolo, primero, en un char sin signo.

14. ¿Qué valor retorna la función *fputc*?

Esta función retorna un carácter escrito como un char convertido a un entero sin signo o un EOF.

¿outputfile debe estar abierto?
Sí, ya que se va a escribir en este.

15. ¿Por qué es muy importante cerrar un archivo antes de terminar un programa?

Es importante cerrar un archivo antes de terminar un programa debido a que se está haciendo uso de un identificador que corresponde a un canal de comunicación entre el ordenador y el archivo, este debe ser cerrado para ser usado por otros archivos.

¿Una vez se cierra un archivo con *fclose* es posible leer o escribir el archivo?

No es posible ya que el canal de comunicación se ha cerrado, entonces no existe comunicación entre el ordenador y el archivo. Es necesario realizar la operación de abrirlo nuevamente.

14.

```
#include <stdio.h>
#include <stdlib.h>
```

```
#include <string.h>

int main(void){
    char *buffer[5];
    char *string1 ="Hola ";
    buffer[0] = malloc(sizeof(char)*30);
    strcpy(buffer[0], string1);
    strcat(buffer[0], "Mundo.");
    puts(buffer[0]);
    free(buffer[0]);
    return 0;
}
```

15. ¿Para qué sirven estas funciones?

- **fprintf:**

```
int fprintf ( FILE * stream, const char * format, ... );
```

Escribe un string al cual apunta el apuntador format. Si existen argumentos insuficientes para el formato, no se tiene un comportamiento definido.

Esta función retorna el número de caracteres enviados o un valor negativo si se ha producido un error.

- **fscanf:**

```
int fscanf(FILE *stream, const char *formato, ...);
```

Obtiene los datos ingresados desde el stream el cual está bajo control por la cadena apuntada por el apuntador formato que especifica las secuencias de entrada permitidas.

Retorna el número de datos de entrada asignados o EOF si se ha producido un error.

- **fgets:**

```
char *fgets(char *cadena, int n, FILE *stream);
```

Lee como máximo n-1 desde el stream al arreglo apuntador por el apuntador cadena.

Retorna la cadena si esta es realizada con éxito o un puntero nulo si se encuentra un error.

- **fputs:**

```
int fputs(const char *cadena, FILE *stream);
```

Escribe la cadena apuntada por cadena al stream apuntado por el apuntador stream. Si se intenta escribir un carácter nulo, este no será escrito.

Escribe EOF si se ha producido un error de escritura, de lo contrario retorna un valor negativo.

Realice un pequeño programa que utilice las 4 funciones.

Este código se encuentra adjunto en el archivo punto15.c

16. Realice un programa que permita calcular el promedio ponderado obtenido por un estudiante en el semestre, considerando:

- 1. El programa recibe el archivo de entrada que se muestra al lado izquierdo.**
- 2. El programa debe generar el archivo de salida que se muestra al lado izquierdo.**

Este código se encuentra adjunto en el archivo punto16.c

El archivo de entrada se llama materias.txt

El archivo de salida se llama resultadomaterias.txt