



Universidad Simón Bolívar  
Departamento de Computación y Tecnología de la Información  
Computación II (CI2126)

## **LABORATORIO 1** **(Repaso: manejo de archivos)**

**MÉTODO DE TRABAJO:** el enunciado del laboratorio será publicado con antelación a la sesión de clases, de modo que el estudiante pueda preparar sus respuestas a las preguntas planteadas en el mismo. En la clase de laboratorio, el docente dará una (1) hora para que los alumnos instrumenten sus soluciones y luego de ese lapso, procederá a discutir y exponer, para todo el grupo, las soluciones alcanzadas y las suyas propias.

**Ejercicio 1:** Dado un archivo de texto, se desea que elabore un programa, en lenguaje C, que permita llenar un arreglo con el contenido secuencial del archivo. Para ello, deberá leer, desde el inicio hasta el final, el contenido del archivo de texto. Deberá colocar además, en cada entrada del arreglo, una cadena de caracteres del archivo. Se quiere además que el nombre del archivo a procesar, se lee desde la misma invocación del programa.

```
/* archivo_texto_a_arreglos_de_valores.c - v1.0    Autor: MTS

Este programa leer un archivo de texto y coloca parte de su
contenido en un arreglo de cadenas de caracteres. Cada entrada
del arreglo es una cadena, no mayor de MAXIMO_TAMANO y cada
cadena del archivo se separa por " ", por "\n" o por EOF. Se
asume además que el archivo a leer tiene un tamaño de líneas
que no supera el valor numérico de MAXIMO_LINEAS_ARCHIVO */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXIMO_TAMANO 135
#define MAXIMO_LINEAS_ARCHIVO 350

/** Principal */
int main (int argc, char **argv)
{
    FILE *fp;
    char arreglo_cadena_caracteres[MAXIMO_LINEAS_ARCHIVO][MAXIMO_TAMANO+1];
    char vector[MAXIMO_TAMANO+1];          /* Arreglo de uso temporal */
    int i, j;

    /* Verificación de los argumentos */
    if (argc != 2) {
        fprintf(stderr, "\n%s: ERROR de sintaxis en la llamada", argv[0]);
        fprintf(stderr, "\nUSO: %s <archivo_a_leer>\n", argv[0]);
        exit (1);
    }
}
```

```

}

/* Apertura del archivo a mostrar */
if ((fp=fopen(argv[1], "r")) == NULL) {
    fprintf(stderr, "\n%s: no puedo abrir %s\n", argv[0], argv[1]);
    exit (2);
}

/* Esta parte lee el archivo y llena el arreglo de cadena de caracteres */
i=0; /* Cuenta la cantidad de entradas que tendrá el arreglo */
j=0;
while (!feof (fp)) {
    fscanf (fp, "%135s", &vector[0]);
    ++i;
    if (j<MAXIMO_LINEAS_ARCHIVO) strcpy (arreglo_cadena_caracteres[j], vector);
    else break;
    ++j;
}

/* Aqui se imprime el arreglo. Esta línea muestra las líneas vacías */
for (j=0; j<i; ++j)
    fprintf (stdout, "\nEntrada %d: %s", j, arreglo_cadena_caracteres[j]);

/* Aqui se cierra el archivo y se termina el programa normalmente */
fclose (fp);
exit (0);
}

```

**Ejercicio 2:** Dado un archivo de texto, se desea que elabore un programa, en lenguaje C, que permita convertir a binario el contenido secuencial del archivo. Para ello, deberá leer, desde el inicio hasta el final, el contenido del archivo de texto. Se quiere además que el nombre del archivo a procesar, se lee desde la misma invocación del programa.

```

/* muestra_archivo_texto_como_binario.c v1.0 */

#include <stdio.h>
#include <stdlib.h>
#define NRO_ELEMENTOS 1

/** Prototipos de Funciones **/
void mostrar_archivo (FILE *ap, char memoria[]);
void rebanar_archivo (FILE *ap, char memoria[]);
void convierte(char n);

/** Principal **/
int main (int argc, char **argv)
{
    FILE *fp;
    char buffer[NRO_ELEMENTOS];
    char tecla;

    /* Verificación de los argumentos */
    if (argc != 2) {
        fprintf(stderr, "\n%s: ERROR de sintaxis en la llamada", argv[0]);
        fprintf(stderr, "\nUSO: %s <archivo_a_visualizar>\n", argv[0]);
        exit (1);
    }
}

```

```

/* Apertura del archivo a mostrar */
if ((fp=fopen(argv[1], "r")) == NULL) {
    fprintf(stderr, "\n%s: no puedo abrir %s\n", argv[0], argv[1]);
    exit (2);
}

fprintf(stdout, "\nEl archivo en su formato de texto contiene:\n");
/* Aqui se invoca el procedimiento */
mostrar_archivo (fp, buffer);

fprintf(stdout, "\nOprima una tecla para continuar ");
scanf ("%c", &tecla);

fprintf(stdout, "\nEl archivo en su formato binario contiene:\n");
/* Aqui se rebana el archivo y se muestra en binario */
rebanar_archivo (fp, buffer);

/* Aqui se cierran los archivos y se termina el programa normalmente */
fclose (fp);
exit (0);
}

/* Esta rutina muestra el contenido de un archivo de caracteres
Atención: Esta solución puede ser modificada con fscanf de este modo:
    fscanf (ap, "%c", memoria); */
void mostrar_archivo (FILE *ap, char memoria[])
{
    int cant_leida;

    rewind (ap);
    while (!feof (ap)) {
        if ((cant_leida=fread(memoria, sizeof(void), NRO_ELEMENTOS, ap)) != 1)
            break;
        fprintf(stdout, "%c", *memoria);
    }
    rewind (ap);
}

/* Esta rutina rebana el contenido de un archivo de caracteres */
void rebanar_archivo (FILE *ap, char memoria[])
{
    int cant_leida;

    rewind (ap);
    while (!feof (ap)) {
        if ((cant_leida=fread(memoria, sizeof(void), NRO_ELEMENTOS, ap)) != 1)
            break;
        convierte (*memoria);
    }
    rewind (ap);
}

/* Esta rutina convierte ASCII a binario */
void convierte (char n)
{
    int i;

    for(i=7; i>=0; i--) printf("%c", (n & (1 << i)) ? '1' : '0');
    printf(" ");
}

```

**Ejercicio 3:** Dado un archivo binario, se desea que elabore un programa, en lenguaje C, que permita copiar su contenido en otro archivo. Se quiere además que los nombres de los archivos a procesar, se lean desde la misma invocación del programa.

```
/* copia_archivo_binario.c v1.0   Autor: MTS */

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

/** Principal */
int main (int argc, char **argv)
{
    FILE *fp, *fp2;
    void *buffer;

    /* Verificación de los argumentos */
    if (argc != 3) {
        fprintf(stderr, "\n%s: ERROR de sintaxis en la llamada", argv[0]);
        fprintf(stderr, "\nUSO: %s <archivo_a_copiar> <archivo_destino>\n", argv[0]);
        exit (1);
    }

    /* Apertura del archivo a mostrar */
    if ((fp=fopen(argv[1], "rb")) == NULL) {
        fprintf(stderr, "\n%s: no puedo abrir %s\n", argv[0], argv[1]);
        exit (2);
    }

    /* Se crea el archivo destino de la copia */
    if ((fp2=fopen(argv[2], "ab")) == NULL) {
        fprintf(stderr, "\n%s: no puedo abrir %s\n", argv[0], argv[2]);
        exit (3);
    }

    rewind(fp);    /* Rebobinando el archivo a copiar */
    /* Aquí se copia el contenido del archivo */
    while (!feof(fp))
    {
        fread (&buffer, sizeof(buffer), 1, fp);
        fwrite(&buffer, sizeof(buffer), 1, fp2);
    }

    /* Aquí se cierran los archivos y se termina el programa normalmente */
    fclose (fp);
    fclose (fp2);
    exit (0);
}
```