

Sistemas Operativos

Formulario de auto-evaluación

Modulo 2. Sesión 1. Llamadas al sistema para el S. Archivos. Parte I

Nombre y apellidos:

David Sánchez Jiménez

a) Cuestionario de actitud frente al trabajo.

El tiempo que he dedicado a la preparación de la sesión antes de asistir al laboratorio ha sido de 40 minutos.

1. He resuelto todas las dudas que tenía antes de iniciar la sesión de prácticas: Si (si/no). En caso de haber contestado “no”, indica los motivos por los que no las has resuelto:

2. Tengo que trabajar algo más los conceptos sobre:

3. Comentarios y sugerencias:

b) Cuestionario de conocimientos adquiridos.

Mi solución al **ejercicio 1** ha sido:

```
# [david @ David-pc in ~/Escritorio] $ ./tarea1

# [david @ David-pc in ~/Escritorio] $ cat archivo
abcdefghijABCDEFGHIJ%

# [david @ David-pc in ~/Escritorio] $ od -c archivo
0000000  a  b  c  d  e  f  g  h  i  j  \0 \0 \0 \0 \0 \0
0000020  \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0
0000040  \0 \0 \0 \0 \0 \0 \0 \0  A  B  C  D  E  F  G  H
0000060  I  J
0000062
```

Mi solución a la **ejercicio 2** ha sido:

```
#include <errno.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

int main(int argc, char const *argv[]) {
    int f_read, f_write, dim, i = 1;
    char buffer[80];
    char *output_file = "salida.txt";

    if (argc < 2) {
        f_read = STDIN_FILENO;
    } else {
        char *pathname = argv[1];
        f_read = open(pathname, O_RDONLY);
    }

    f_write = open(output_file, O_CREAT | O_TRUNC | O_WRONLY, S_IRUSR | S_IWUSR);

    if (f_read < 0) {
        printf("\nError %d en el open", errno);
        perror("\nNo se ha podido abrir el archivo de lectura.");
        exit(-1);
    }

    if (f_write < 0) {
        printf("\nError %d en el write", errno);
        perror("\nNo se ha podido abrir el archivo de escritura.");
        exit(-1);
    }
}
```

```
while ((dim = read(f_read, &buffer, 80)) > 0) {
    char buffer_ID[12];
    sprintf(buffer_ID, "##Bloque %i##", i);
    write(f_write, &buffer_ID, sizeof(buffer_ID));
    write(f_write, "\n", 1);

    write(f_write, &buffer, dim);
    write(f_write, "\n", 1);
    write(f_write, "\n", 1);

    i++;
}

if (lseek(f_write, 0, SEEK_SET) < 0) {
    perror("\nError en lseek");
    exit(-1);
}

char num_bloques[27];
sprintf(num_bloques, "El numero de bloques es %i", i - 1);

write(fd2, &num_bloques, sizeof(num_bloques));
write(fd2, "\n", 1);

close(f_write);
close(f_read);
return 0;
}
```

Mi solución a la **ejercicio 3** ha sido:

Este programa muestra de que tipo son los ficheros que se le pasan como argumento.

Estos pueden ser archivos regulares, directorios, especiales (de caracteres o de bloques), FIFO, enlaces simbolicos, socket o desconocido.

Mi solución a la **ejercicio 4** ha sido:

```
#include <errno.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

#define S_ISREG2(mode) (((mode)&S_IFMT) == 0100000)

int main(int argc, char const *argv[]) {
    int i;
    struct stat atributos;
    char tipoArchivo[30];

    if (argc < 2) {
        printf("\nSintaxis de ejecucion: tarea2 [<nombre_archivo>]+\n\n");
        exit(-1);
    }

    for (i = 1; i < argc; i++) {
        printf("%s: ", argv[i]);
        if (lstat(argv[i], &atributos) < 0) {
            printf("\nError al intentar acceder a los atributos de %s", argv[i]);
            perror("\nError en lstat");
        } else {
            if (S_ISREG(atributos.st_mode))
                strcpy(tipoArchivo, "Regular");
            else if (S_ISDIR(atributos.st_mode))
                strcpy(tipoArchivo, "Directorio");
            else if (S_ISCHR(atributos.st_mode))
                strcpy(tipoArchivo, "Especial de caracteres");
            else if (S_ISBLK(atributos.st_mode))
                strcpy(tipoArchivo, "Especial de bloques");
            else if (S_ISFIFO(atributos.st_mode))
                strcpy(tipoArchivo, "Tuberia con nombre (FIFO)");
            else if (S_ISLNK(atributos.st_mode))
                strcpy(tipoArchivo, "Enlace relativo (soft)");
            else if (S_ISSOCK(atributos.st_mode))
                strcpy(tipoArchivo, "Socket");
            else
                strcpy(tipoArchivo, "Tipo de archivo desconocido");
            printf("%s\n", tipoArchivo);
        }
    }
    return 0;
}
```