

Sistemas Operativos

Formulario de auto-evaluación

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

Nombre y apellidos:

Néstor Rodríguez Vico

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 ... 60... minutos.

1. He resuelto todas las dudas que tenía antes de iniciar la sesión de prácticas: (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:

Llamadas al sistema

3. Comentarios y sugerencias:

b) Cuestionario de conocimientos adquiridos.

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

Abre “archivo”, comprueba que se ha abierto correctamente y a continuación copia el primer buffer, luego cambia la posición de escritura actual en el archivo y escribe el segundo buffer. AL final programa el contenido de nuestro archivo es:
abcdefghijkl#####ABCDEFGHIJ
Si usamos la orden “od -c archivo” se nos muestran todos los caracteres del archivo, incluso los que cat no puede mostrarnos.

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

Mi código fuente (con la modificación incluida) es:

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

char aux[80];
char bloque[80];
char total[80];
int contador=0;
char pantalla[5];

int main(int argc, char *argv[]){
    int entrada, salida;
    ssize_t r, w;

    if(argc == 1){
        scanf("%s", pantalla);
    }

    if(argc == 2){
        if( (entrada=open(argv[1],O_RDONLY,S_IRUSR|S_IWUSR))<0) {
            printf("\nError %d en open entrada",errno);
            perror("\nError en open entrada");
            printf("\n");
            exit(EXIT_FAILURE);
        }
    }
}
```

```
if( (salida=open("salida.txt",O_CREAT|O_TRUNC|O_WRONLY,S_IRUSR|S_IWUSR))<0) {
    printf("\nError %d en open salida",errno);
    perror("\nError en open salida");
    printf("\n");
    exit(EXIT_FAILURE);
}

if(lseek(salida,100,SEEK_SET) < 0) {
    perror("\nError en lseek");
    exit(EXIT_FAILURE);
}

if(argc == 2){
    r=read(entrada,aux,80);
    while(r!=0){
        contador=contador+1;
        sprintf(bloque, "Bloque %d", contador);
        write(salida, "\n", 1);
        write(salida,bloque, 80);
        write(salida,"\n", 1);
        w=write(salida,aux,(unsigned int) r);
        if(w!=(unsigned int) r) {
            perror("Error en write. \n");
            exit(EXIT_FAILURE);
        }
        r=read(entrada,aux,80);
    }

    if(lseek(salida,0,SEEK_SET) < 0) {
        perror("\nError en lseek");
        exit(EXIT_FAILURE);
    }
    write(salida,"Numero de bloques ", 18);
    sprintf(total, "%d", contador);
    write(salida,total,80);
    write(salida,"\n",1);
}

if(argc==1){
    r=read(0,aux,80);
```

```

while(r!=0){
    contador=contador+1;
    sprintf(bloque, "Bloque %d", contador);
    write(salida, "\n", 1);
    write(salida,bloque, 80);
    write(salida,"\n", 1);
    w=write(salida,aux,(unsigned int) r);
    if(w!=(unsigned int) r) {
        perror("Error en write. \n");
        exit(EXIT_FAILURE);
    }
    r=read(0,aux,80);
}

if(lseek(salida,0,SEEK_SET) < 0) {
    perror("\nError en lseek");
    exit(EXIT_FAILURE);
}

write(salida,"Numero de bloques ", 18);
sprintf(total, "%d", contador);
write(salida,total,80);
write(salida,"\n",1);
}

close(entrada);
close(salida);
return EXIT_SUCCESS;
}

```

Para ver si funciona correctamente, estoy usando un archivo que contiene:

aaaaaaaaaaaaaaaaaaaaa

bbbbbbbbbbbb

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

dddddd

ee

ffff

gg

hhhhhhhhhhhhhhhhhhhh

[illegible]

.....


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

int main(int argc, char *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_ISREG2(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;  
}
```

Cuando lo ejecutamos usando la orden “./a.out ejercicio2.c src” nos devuelve:

ejercicio2.c: Regular

src: Directorio