

Sistemas Operativos

Formulario de auto-evaluación

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

Nombre y apellidos:

Jesús Manuel García Palma

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 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:

Flags

3. Comentarios y sugerencias:

b) Cuestionario de conocimientos adquiridos.

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

El ejercicio 1 crea dos arrays de 10 caracteres cada uno. Crea un entero fd al que se le asigna el valor de la llamada al sistema open del archivo "archivo". Da error si es menor que 0.

Se crea con O_CREAT si no existe o O_TRUNC si existe el fichero y tiene habilitada la escritura y lo sobrescribe a tamaño 0.

Con O_WRONLY se impone que solo se permite escritura.

Con S_IRUSR se comprueba que el usuario tiene permiso de lectura.

Con S_IWUSR se comprueba que el usuario tiene permiso de escritura.

A continuación comprueba que si después de utilizar la orden write en el buf1 el resultado es distinto de 10 se muestra error ya que se han escrito los 10 caracteres del buf1 en el archivo.

Después con lseek se pone el puntero del archivo en la posición 40 desde SEEK_SET(el inicio del fichero), y dará error si el resultado de la operación es menor que 0.

Esto da lugar a que el puntero se coloque después de la posición 40, al final del buf1.

Se llama a write para el segundo bufer. Imprime los 10 caracteres.

Se hace el cat:

```
abcdefghijABCDEFGHIJ
```

```
luego oc -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<sys/types.h>
#include<sys/stat.h>
#include<fcntl.h>
#include<string.h>
#include<stdlib.h>
#include<stdio.h>
#include<errno.h>
#include<unistd.h>
int main(int argc, char *argv[])
{
int cont=1,leidos;
int filein,fileout;
char cadena[30];
char cad_bloque[40];
char salto_linea[2]="\n";
char caracter[1];
int num_char = 1;
if (argc==2)
{
filein=open(argv[1], O_RDONLY);
}
else{
filein=STDIN_FILENO;
}
fileout=open("archivo_salida", O_CREAT|O_TRUNC|O_WRONLY,S_IRUSR|S_IWUSR);
if (fileout < 0){
printf("El fichero de salida no se pudo abrir correctamente\n");
exit(-1);
}
```

```
while((leidos=read(filein,caracter,1))!=0){
if (num_char == 1 || num_char%80 == 0){
if (num_char != 1)
write(fileout, salto_linea, strlen(salto_linea));
else{
sprintf(cad_bloque, "El numero de bloques es <%d>\n",cont);
write(fileout, cad_bloque, strlen(cad_bloque));
}

sprintf(cad_bloque, "%s%d\n", "Bloque", cont);
write(fileout, cad_bloque, strlen(cad_bloque));
cont++;
}
write(fileout, caracter, 1);
num_char++;
}
sprintf(cad_bloque, "El numero de bloques es <%d>\n", cont);
lseek(fileout,0,SEEK_SET);
write(fileout, cad_bloque, strlen(cad_bloque));
close(filein);
close(fileout);
return 0;
}
```

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

Una vez se ejecuta el ejercicio pasándole como argumento el nombre del archivo, nos dice qué tipo de archivo es.

El programa lo que hace es comprobar cada flag para ver que tipo de archivo es, y si se activa lo mostrará

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

```
#define S_ISREG2(mode) (mode & S_IFMT == S_IFREG)
```