

Sistemas Operativos

Formulario de auto-evaluación

Molulo 2. Sesión 2. Llamadas al sistema para el S.Archivos Parte II

Nombre y apellidos:

Cristina María Garrido López

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: 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:

Máscaras y macros.

3. Comentarios y sugerencias:

b) Cuestionario de conocimientos adquiridos.

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

El programa crea un archivo1 con permisos de lectura, escritura y ejecución para el grupo, establece la máscara a cero y seguidamente crea otro archivo2 con los mismos permisos. Intenta acceder a los atributos y si se puede cambia los permisos del archivo1 de manera que se queda con los del struct stat atributos y activa la asignación del GID del propietario al GID efectivo del proceso que ejecute el archivo. Al archivo2 le asigna permisos de lectura, escritura y ejecución al usuario, de lectura y escritura para el grupo y lectura para otros. Finalmente cierra los dos archivos.

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

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

int main(int argc, char *argv[]){
    int permisos;
    char* pathname;
    struct stat atributos;
    pathname=argv[1];
    DIR* directorio;

    if(argc==3){
        directorio=opendir(pathname);
        if(pathname==NULL){
            printf("\nError %d al abrir el directorio",errno);
```

```
        perror("\nError en open");
        exit(-1);
    }

    struct dirent* arch=readdir(directorio);

    permisos=strtol(argv[2],NULL,8);

    char cadena[100];
    char cadena2[100];

    while(arch!=NULL){
        sprintf(cadena,"%s" "%s",pathname,arch->d_name);

        if(stat(cadena,&atributos) < 0) {
            printf("\nError al intentar acceder a los atributos del pathname");
            perror("\nError en lstat");
            exit(-1);
        }

        sprintf(cadena2,"%s",arch->d_name);
        printf("%s" "%o",cadena2,atributos.st_mode);

        if((chmod(cadena, permisos))<0){
            printf("Error");
        }else{
            stat(cadena,&atributos);
            printf("%o",atributos.st_mode);
        }
    }
}
```

```
        }

    }

    closedir(directorio);
} else {
    printf("\nError. El número de argumentos es incorrecto\n");
    exit(-1);
}
}
```

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

```
#include<sys/types.h>
#include<unistd.h>
#include<stdlib.h>
#include<string.h>
#include<sys/stat.h>
#include<fcntl.h>
#include<stdio.h>
#include<errno.h>
#include<dirent.h>
#define cond(mode) ((mode) & (S_IXGRP|S_IXOTH) == (S_IXGRP|S_IXOTH))
struct st{
    int suma;
    int tama;
};

struct st buscar(char nombre[]){
    DIR* directorio;
    struct dirent* arch;
    struct stat atributos;
```

```
char cadena[1000];

struct st su_ta;

struct st aux;


su_ta.suma=0;
su_ta.tama=0;


if((directorio = opendir(nombre))<0){
    printf("No se pudo abrir el directorio");
    error(-1);
}


while((arch=readdir(directorio))!=NULL){
    if(strcmp(arch->d_name,".")!=0 && strcmp(arch->d_name,"..")!=0){
        sprintf(cadena,"%s/%s",nombre,arch->d_name);

        if((stat(cadena,&atributos))<0){
            printf("No se puede acceder a los atributos");
            exit(-1);
        }

        if(S_ISDIR(atributos.st_mode)){
            aux=buscar(cadena);

            su_ta.suma=su_ta.suma+aux.suma;
            su_ta.tama=su_ta.tama+aux.tama;
        }else{
            printf("%s %ld\n",cadena,atributos.st_ino);
            if(S_ISREG(atributos.st_mode) && cond(atributos.st_mode)){
                su_ta.suma++;
                su_ta.tama=su_ta.tama+atributos.st_size;
            }
        }
    }
}
```

```
        printf("%s %ld",cadena,arch->d_ino);
    }
}
}
}
return su_ta;
}

int main(int argc, char* argv[]){
    char name[1000];
    DIR* directorio;
    struct st res;

    if(argc==2){
        strcpy(name,argv[1]);
    }else
        if(argc==1)
            strcpy(name,".");
        else{
            printf("Numero de parámetros incorrecto.");
            exit(-1);
        }

    if((directorio=opendir(name))==NULL){
        printf("Error en opendir");
        exit(-1);
    }

    printf("Los i-nodos son: ");
    res=buscar(name);
```

```
printf("\n\n");

printf("Existen %d archivos regulares con permiso de ejecución para grupo y otros",
res.suma);

printf("\n");

printf("El tamaño total ocupado por dichos archivos es %d bytes",res.tama);

}
```