

Sistemas Operativos

Formulario de auto-evaluación

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

Nombre y apellidos:

Emilio Chica 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 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:

3. Comentarios y sugerencias:

b) Cuestionario de conocimientos adquiridos.

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

Abre dos archivos y les cambia los permisos del siguiente modo:

Crea ambos archivos de modo que si existen los sobrescribe y los abre para escritura solamente.

Al archivo2 le aplica un mascara con umask, y a partir de esta instrucción todos los archivos que se abran o se creen se establecerán con esta mascara: $0666 \& \sim 0000 = 0666$.

Ahora asigna los atributos del archivo1 a un stat llamado atributos.

Cambia los permisos del archivo1 con chmod le quita los permisos de ejecución al grupo y establece que el archivo adquiera los permisos del grupo asignado al fichero.

Cambia los permisos del archivo2 a lectura, escritura y ejecución de usuario, lectura y escritura de grupo, y lectura de otros.

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

```
//PARA MEJOR LECTURA SE INCLUYE EL .C
#include <sys/types.h>
#include <dirent.h>
#include <stdio.h>
#include <sys/stat.h>
#include <unistd.h>

int main(int argc, char * argv[]){

    if(argc!=3){

        printf("\nSintaxis      de      ejecucion:      ./ejercicio2      [<nombre_directorio>]
[<permisos_octal>]\n\n");
```

```
        exit(-1);

    }else
    {

        struct stat atributos;

        char * directorio = argv[1];

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

        DIR* direc= opendir(directorio);

        if (direc == NULL) {

            perror("opendir");

            exit(-1);

        }

        struct dirent * entrada;

        char * nameArchiCompleto[100];

        while((entrada=readdir(direc))!=NULL){

            strcpy(nameArchiCompleto, directorio);

            char * nombre = entrada->d_name;

            strcat(nameArchiCompleto,"/");

            strcat(nameArchiCompleto,nombre);

            if(stat(nameArchiCompleto,&atributos)<0)

            {

                perror("stat");

                exit(-1);

            }

            printf("%s : %o %o ",nombre, atributos.st_mode, permisos);

            printf("\n");

            if(chmod(nameArchiCompleto,permisos)<0)

            {

                perror("chmod");

                exit(-1);

            }

        }

    }
```

```
        }  
    }  
}  
  
}
```

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

```
//PARA MEJOR LECTURA SE INCLUYE EL .C  
  
#include <sys/types.h>  
#include <dirent.h>  
#include <stdio.h>  
#include <sys/stat.h>  
#include <unistd.h>  
#include <string.h>  
  
void buscar(char * dir){  
    int cont=0,cont2=0;  
    printf("DIRECTORIO: %s\n",dir);  
    DIR* direc= opendir(dir);  
    if (direc == NULL) {  
        perror("opendir");  
        exit(-1);  
    }  
    struct stat atributos;  
    struct dirent * entrada;  
    printf("Los i-nodos son:\n");
```

```
char new_dir[100];

while((entrada=readdir(direc))!=NULL){

    char aux[200];

    strcpy(aux,dir);
    strcat(aux,"/");
    strcat(aux,entrada->d_name);
    if(stat(aux,&atributos)<0){

        perror("stat");
        exit(-1);
    }

    if(strcmp(entrada->d_name, ".")==0 || strcmp(entrada->d_name, "..")==0 || !
S_ISDIR(atributos.st_mode))
    {

        if(S_ISREG(atributos.st_mode) && (atributos.st_mode & S_IXGRP) &&
(atributos.st_mode & S_IXOTH))
        {

            printf("%s %d \n",entrada->d_name,entrada->d_ino );
            cont+=atributos.st_size;
            cont2++;
        }
    }
    else
    {

        strcpy(new_dir,dir);
        strcat(new_dir,"/");
        strcat(new_dir,entrada->d_name);
        buscar(new_dir);
    }

}
```

```
        printf("Existen %i archivos regulares con permiso x para grupo y otros\n",cont2);
        printf("El tamaño total ocupado por dichos archivos es %i bytes\n",cont);
    }
int main(int argc, char * argv[]){
    char dir[100];
    if(argc==1){
        strcpy(dir,".");
    }
    else
        if(argc==2)
            strcpy(dir,argv[1]);
        else
        {
            printf("\nSintaxis de ejecucion: ./ejercicio3 ó ./ejercicio3 [<directorio>]\n\n");
            exit(-1);
        }
    buscar(dir);
}
```