



Gómez Miranda Leopoldo

Tarea: Desarrollo de comandos.

Programa que borrar directorios de forma recursiva.

```
#include <string.h>
#include <sys/stat.h>
#include <dirent.h>
#include <stdlib.h>
#include <stdio.h>

void borrar(char *dir)
{
    DIR *dp;
    struct dirent *entrada;
    struct stat statbuf;

    rmdir(dir);

    if((dp = opendir(dir)) == NULL)
        return;

    chdir(dir);
    while((entrada = readdir(dp)) != NULL)
    {
        lstat(entrada->d_name,&statbuf);
        if(S_ISDIR(statbuf.st_mode)) //pregunta
        {
            //Encuentra un directorio
            if(strcmp(".",entrada->d_name) == 0 || strcmp("..",entrada->d_name)
== 0)
                continue;
            if(rmdir(entrada->d_name) != 0)
                borrar(entrada->d_name); //Llama de nuevo para borrar dir
        }
        else remove(entrada->d_name); //borrar ficheros
    }
    chdir("..");
    closedir(dp);
    borrar(dir); //Llama de nuevo para borrar dir
}
```



```
int main(int argc, char *argv[])
{
    // Analisis de los argumentos de la linea de ordenes
    if (argc < 2)
        fprintf(stderr, "Forma de uso: %s nombre_directorio. \n", argv [0]);
    else
        borrar(argv[1]);
}
```

```
# mkdir carpeta1
# cd carpeta1
# mkdir carpeta2
# cd carpeta2
# touch 1 2 3 4
# cd ..
# cd ..
# pwd
/root/programas/prueba
# gcc borrar.c
# ./a.out carpeta1
# ls
a.out      borrar.c
#
```

Programa myls.c

```
// Mi comando ls -l
#include <unistd.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <pwd.h> // Para getpwuid
#include <grp.h> // Para getgrid

char permisos [] = {'x', 'w', 'r'};

int main(int argc, char *argv[])
{
    // Analisis de los argumentos de la linea de ordenes
    if (argc < 2){
        fprintf(stderr, "Forma de uso: %s nombre_fichero. \n", argv [0]);
    }
}
```



```
struct stat filestat;
struct passwd *pw;
struct group *gr;

//comprobar stat
if ( stat(argv[1], &filestat) == -1){
    perror (argv[1]);
}
//tipo de fichero
switch (filestat.st_mode & S_IFMT) {
case S_IFREG:
    printf ("-");
    break;
case S_IFDIR:
    printf ("d");
    break;
case S_IFCHR:
    printf ("c");
    break;
case S_IFBLK:
    printf ("b");
    break;
case S_IFIFO:
    printf ("f");
    break;
}

//Presentación de los permisos de lectura, escritura y ejecución
int i;
for (i = 0; i < 9; i++)
    if (filestat.st_mode & (0400 >> i))
        printf ("%c",permisos [(8-i)%3]);
    else
        printf("-");
//Ligas
printf (" %d", filestat.st_nlink);
//UID y GID
if ((pw = getpwuid (filestat.st_uid))== NULL)
    printf(" ???");
else
    printf (" %s", pw->pw_name);
    if ((gr = getgrgid (filestat.st_gid))== NULL)
        printf(" ???");
    else
```



```
printf(" %s", gr->gr_name);

//Tamaño del fichero
printf(" %d", filestat.st_size);

//Fecha de ultima modificacion
struct tm *tm;
char buf[200];
tm =localtime(&filestat.st_mtime);
strftime(buf, sizeof(buf), "%b %d %H:%M", tm);
printf(" %s",buf);

//nombre del fichero
printf(" %s\n", argv[1]);
}
```

```
# ls -l hola.c
-rw-r--r-- 1 root  wheel  104 Feb 17 02:01 hola.c
# ./a.out hola.c
-rw-r--r-- 1 root  wheel 104 Feb 17 02:01 hola.c
# █
```

Programa mystat

```
// Mi comando stat -x
#include <unistd.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <pwd.h> // Para getpwuid
#include <grp.h> // Para getgrid

char permisos [] = {'x', 'w', 'r'};

int main(int argc, char *argv[])
{
    // Analisis de los argumentos de la linea de ordenes
    if (argc < 2){
```



```
        fprintf (stderr, "Forma de uso: %s nombre_fichero. \n", argv [0]);
    }

    struct stat filestat;
    struct passwd *pw;
    struct group *gr;
    //comprobar stat
    if ( stat(argv[1], &filestat) == -1){
        perror (argv[1]);
    }
    //Fichero
    printf(" File: \"%s\" \n", argv[1]);

    //Tamaño del fichero
    printf(" Size: %d", filestat.st_size);

    //Tipo de Fichero
    printf ("   FileType: ");
    //Análisis del tipo de dispositivo.
    switch (filestat.st_mode & S_IFMT) {
    case S_IFREG:
        printf ("Regular File\n");
        break;
    case S_IFDIR:
        printf ("Directory\n");
        break;
    case S_IFCHR:
        printf ("Character \n");
        break;
    case S_IFBLK:
        printf ("Block\n");
        break;
    case S_IFIFO:
        printf ("Fifo\n");
        break;
    }

    //Presentación de los permisos de lectura, escritura y ejecución en la forma
    //(0777/rwxrwxrwx)
    printf (" Mode: (0%o", filestat.st_mode & 0777);
    int i;
    for (i = 0; i < 9; i++)
        if (filestat.st_mode & (0400 >> i))
            printf ("%c",permisos [(8-i)%3]);
        else
```



```
        printf("-");
printf(")");
//UID y GID
printf ("      Uid: (  %d/  ", filestat.st_uid);
if ((pw = getpwuid (filestat.st_uid))== NULL)
    printf("??? ) ");
else
    printf ( "%s)", pw->pw_name);

printf (" Gid: (  %d/  ", filestat.st_gid);
if ((gr = getgrgid (filestat.st_gid))== NULL)
    printf("??? ) ");
else
    printf ( "%s)\n", gr->gr_name);

//Dispositivo
printf("Device: %d,%d",
(filestat.st_dev & 0xff00) >> 8, filestat.st_dev & 0x00ff);

//Numero de nodo-i
printf ("  Inode: %d", filestat.st_ino);

//Numero de enlaces.
printf ("  Links: %d\n", filestat.st_nlink);

//Presentacion de los numeros de dispositivo
switch (filestat.st_mode & S_IFMT){
case S_IFCHR:
case S_IFBLK:
    printf (" Numeros de dispositivo: %d, %d\n",
(filestat.st_rdev & 0xff00) >> 8, filestat.st_rdev & 0x00ff);
}

//Fechas del fichero
printf("Access: %s",
    asctime (localtime (&filestat.st_atime)));
printf("Modify: %s",
    asctime (localtime (&filestat.st_mtime)));
printf("Change: %s",
    asctime (localtime (&filestat.st_ctime)));

    return 0;
}
```



```
# gcc mystat.c
# ./a.out hola.c
  File: "hola.c"
  Size: 104      FileType: Regular File
  Mode: (0644/rw-r--r--)  Uid: (  0/   root)  Gid: (  0/   wheel)
Device: 0,0  Inode: 131064  Links: 1
Access: Sun Feb 17 02:02:23 2013
Modify: Sun Feb 17 02:01:46 2013
Change: Sun Feb 17 02:01:46 2013
#
# stat -x hola.c
  File: "hola.c"
  Size: 104      FileType: Regular File
  Mode: (0644/-rw-r--r--)  Uid: (  0/   root)  Gid: (  0/   wheel)
Device: 0,0  Inode: 131064  Links: 1
Access: Sun Feb 17 02:02:23 2013
Modify: Sun Feb 17 02:01:46 2013
Change: Sun Feb 17 02:01:46 2013
# █
```