**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

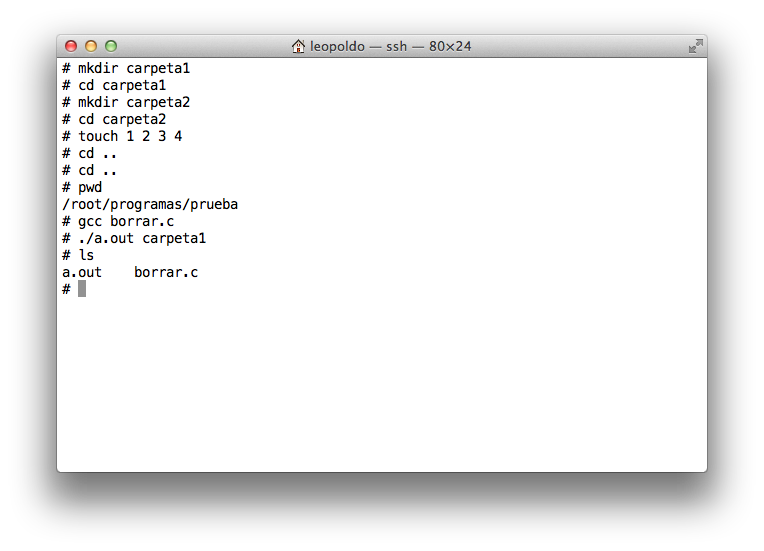
if (argc < 2)

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

else

borrar(argv[1]);

}



**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);

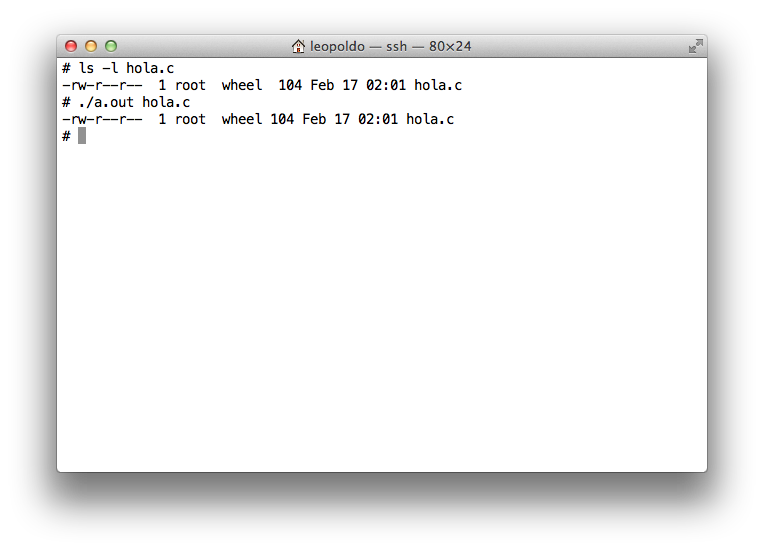
strftime(buf, sizeof(buf), "%b %d %H:%M", tm);

printf(" %s",buf);

//nombre del fichero

printf(" %s\n", argv[1]);

}



**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;

}

