#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <errno.h>

#include <string.h>

#include <dirent.h>

#include <signal.h>

#include <signal.h>

/\*

gcc -g VER-home.c -o VER2

./VER2

\*/

int main(int argc, char const \*argv[])

{

//int fd[2];

int PID;

int fd;

int estado;

char \*home=getenv("HOME");

struct dirent \*entrada;

struct stat atributos;

DIR \*carpeta;

char ruta[256];

sprintf(ruta,"%s/Escritorio/prueba",home);

//printf("RUTA-><%s>\n",ruta );

if((carpeta=opendir(ruta))==NULL)//abro el argumeto que le paso(el directorio)

{

perror("no se ha podido leer el directorio\n");

exit (-1);

}

while((entrada=readdir(carpeta))!=NULL)

{

//VICENTE NO SE DECLARA DOS VARIABLES Y MACHACA RUTA

//char direccion[512];// me creo un char para guardar el directorio/nombre.txt actual

sprintf(ruta,"%s/Escritorio/prueba/%s",home,entrada->d\_name);//esto es el directorio/lo de dentro(nombre)

//printf("\nDIRECCION:--> %s\n\n",ruta );

if(lstat(ruta,&atributos)<0)

{

perror("Error al intentar acceder a los atributos del archivo\n");

exit (-1);

}

if(S\_ISLNK(atributos.st\_mode))

{

printf("\nEl archivo [%s] Usuario <%d> y Grupo propietario [%d]",entrada->d\_name,atributos.st\_uid,atributos.st\_gid);

if((PID=fork())<0)

{

perror("Error en el fork");

exit(1);

}

else

{

execlp("./VER","./VER",ruta,NULL);//creo que lo deberia de hacer el hijo1

perror("Error en el execlp 22");

exit(1);

}

wait(&estado);

if(estado==EXIT\_SUCCESS)

{

///hijo2 --> creo hijo pk si no me termina el padre al execlp

if((PID=fork())<0)

{

perror("Error en el fork");

exit(1);

}

if(PID==0)

{

execlp("mountpoint","mountpoint",ruta,NULL);

perror("Error en el execlp");

exit(1);

}

wait(NULL);

}

else{

printf("FALLO ESTADO HIJO\n");

}

}

}

return 0;

}