#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>

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

{

pid\_t PID;

int fd[2];

if(pipe(fd)<0)

{

perror("error al crear el pipe");

exit(1);

}

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

{

perror("error al crear hijo1");

exit(1);

}

struct dirent \*entrada;

DIR \*carpeta;

struct stat atributos;

if(PID!=0)

{

close(fd[0]);

if((carpeta=opendir("./prueba"))==NULL)//pongo esto para que no me la lie cuando imprima archivos .cpp

{

perror("error al acceder al directorio");

exit(1);

}

while ((entrada=readdir(carpeta))!=NULL)//voy recorriendo todos los ficheros del directorio que le paso como ruta

{

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

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

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

if(stat(entrada->d\_name,&atributos)<0)

{

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

exit (-1);

}

if(S\_ISREG(atributos.st\_mode))//tengan permisos de lectura

{

printf("\nEl archivo [%s] su INODO <%llo> UID<%u>",entrada->d\_name,entrada->d\_ino,atributos.st\_uid);// %

if(write(fd[1],entrada->d\_name,sizeof(char)\*256)<0)//es peor si pongo entrada->d\_name

{

perror("Error al escribir en el cauce\n");

exit (-1);

}

}

}

}

else

{

char \*p;

int fo;

int tam\_block[512];

int leido;

close(fd[1]);

char cadena[256];

while(read(fd[0],cadena,sizeof(char)\*256)>0)

{

printf("\nLEIDO: %s\n", cadena);

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

{

perror("ERROR\n");

exit (-1);

}

close(fo);

}

}

return 0;

}