#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>

/\*

-gcc -g Principal.c -o PRINCIPAL

-./PRINCIPAL /home/miguel/Escritorio/SO\_T-formacion/Examen/Examen2\_angustias/

-Te pide un numero (archivo a sacar los datos)

-Llamara a LeerArchivo.c

Archivo--><zdoc.txt> INODO:<151236> TAM:<29> PERMISOS:<100644>

\*/

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

{

pid\_t PID;

if(argc!=2)

{

perror("Error numero argumetos");

exit(1);

}

int fd[2];

if(pipe(fd)<0)

{

perror("error al crear el pipe");

exit(1);

}

//ESTO LO HACE EL PADRE

struct stat atributos;

struct dirent \*entrada;

DIR \*carpeta;

if((carpeta=opendir(argv[1]))==NULL)

{

perror("error al acceder al directorio");

exit(1);

}

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

{

perror("error al crear hijo1");

exit(1);

}

///////////////////////////////////////////////////////////////////////////////////////////

if(PID==0)

{

close(fd[0]);

int numero;//COMO HAGO QUE EL HIJO LEA UN NUMERO DE STDIN\_FILENO

printf("Introduzca un numero\n");

scanf("%d",&numero );

//printf("El numero es:%d\n",numero );

int i;

for (i = 0; i < numero; i++)

{

entrada=readdir(carpeta);

}

if(entrada!=NULL )

{

//FILTRO DE MAXIMO 10 CARACTERES

int longitud = strlen(entrada->d\_name);

if (longitud > 10) {

longitud = 10;

}

if((write(fd[1],&entrada->d\_name,longitud\*sizeof(char)))<0)

{

perror("error WRITE1");

exit(1);

}

}

else

{

if((write(fd[1],"FIN",sizeof(char)\*3))<0)

{

perror("error WRITE2");

exit(1);

}

}

}

////////////////////////////////////////////////////////////////////////////

else

{

//continua el padre creando el 2ª hijo

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

{

perror("error al crear hijo2");

exit(1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if(PID==0)

{

close(fd[1]);

if((dup2(fd[0],STDIN\_FILENO))<0)

{

perror("error en el dup2");

exit(1);

}

//char c;

/\*while(read(fd[0],&c,sizeof(char))>0) //comprobando lo que se mete en el cauce

printf("%c",c);

printf("\n");

\*/

execlp("./LEER","./LEER",NULL); //NO CONSIGUO

perror("\nError en el execlp");

exit(-1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

else

{

close(fd[0]);//cierro los DESCRIPTORES del padre

close(fd[1]);//cierro los DESCRIPTORES del padre

wait(NULL);

wait(NULL);

printf("\nFIN DEL PROGRAMA\n");

}

}

return 0;

}