#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>

/\*

gcc -g migrep.c -o MIGREP

./MIGREP d ~/Escritorio/prueba/

Archivo<IntroVirtualizacion.pdf> contiene <3779>

Archivo<SOFLINK> contiene <0>

Archivo<archivo1> contiene <0>

Archivo<ejercicio2\_perfecto.cpp> contiene <32>

Archivo<calculopi.cpp> contiene <52>

Archivo<Ejercicio2.c> contiene <23>

Archivo<archivo3> contiene <3>

\*/

int contador=0;

void manejador(int sig)

{

printf("\nFINALIZAMOS\n Total Contador <%d>",contador );

exit(EXIT\_SUCCESS);

}

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

{

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

/\*int fderror;

if((fderror=open("Errores.txt",O\_WRONLY| O\_CREAT,0744))<0)

{

perror("error numero argumentos");

exit(EXIT\_FAILURE);

}

dup2(fderror,STDERR\_FILENO);

\*/

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

if(argc!=3)

{

perror("error numero argumentos");

exit(EXIT\_FAILURE);

}

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

struct sigaction sa;

sa.sa\_handler=manejador;

if(sigaction(SIGINT,&sa,NULL)<0)

{

perror("error signation");

exit(EXIT\_FAILURE);

}

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

int numero;

struct dirent \*entrada;

DIR \*carpeta;

struct stat atributos;

int fd[2],PID;

if(pipe(fd)<0)

{

perror("error Abrir directorio");

exit(EXIT\_FAILURE);

}

if((carpeta=opendir(argv[2]))==NULL)

{

perror("error Abrir directorio");

exit(EXIT\_FAILURE);

}

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

{

char ruta[256];

sprintf(ruta,"%s/%s",argv[2],entrada->d\_name);

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

{

perror("error al ACCEDER A LOS ATRIBUTOS");

exit(EXIT\_FAILURE);

}

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

{

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

{

perror("error FORK");

exit(EXIT\_FAILURE);

}

if(PID==0)

{

// printf("Archivo---->>>><%s> \n",entrada->d\_name);

close(fd[0]);//cierro el descriptor de lectura del hijo1, pk se va a encargar de escribir en el cauce y no espera ningun dato de teclado

if((dup2(fd[1],STDOUT\_FILENO))<0)//aqui lo que hago es que cuando haga el excep no lo muestre en pantalla y lo escriba en el cauce

{

perror("\nError en el dup2 hijo1");

exit(-1);

}

//printf("\nARGV[1]<%s>\n",argv[1]);

//printf("ARGV[1]<%s>\n",argv[2]);

execlp("grep","grep","-c",argv[1],ruta,NULL);//grep -c ";" home/miguel/Escritorio/archivo.txt te envia <nº>

perror("error en el EXCELP");

exit(EXIT\_FAILURE);

}

else

{

char c;

//close(fd[1]);

//printf("Archivo<%s> \n",entrada->d\_name);

printf("Archivo<%s> contiene <", entrada->d\_name);

while(read(fd[0],&c,sizeof(char))>0 && c != '\n')//\n es pk el grep mete un salto de linea

{

printf("%c", c);

}

printf(">\n");

contador\*=numero;

wait(PID);

}

}

}

exit(EXIT\_SUCCESS);

}