#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 <limits.h>

/\* ./EXPLO /etc/passwd

\*/

void manejador(int siguiente)

{

int estado;

wait(&estado);

if(estado!=0)

{

printf("FINALIZADO CON ERROR\n" );

}

printf("\nHIJO ha terminado con exito\n" );

}

void procesador(const char\* nombre)

{

//hacemos todo lo de cambiar permisos

struct stat atributos;

char fichero[PATH\_MAX],blq[512];

int leido,fd;

mode\_t permisos;

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

{

perror("error STAT");

return;

}

if(!S\_ISDIR(atributos.st\_mode))

{

perror("error ATRIBUTOS");

return;

}

permisos=S\_IRWXU| atributos.st\_mode;//le añadimos lo que ya tenia mas

if(chmod(nombre,permisos)<0)

{

perror("error CHMOD");

return;

}

sprintf(fichero,"%s./bashrc",nombre);

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

{

perror("error STAT2");

return;

}

if((fd=open(fichero,O\_RDONLY))<0)

{

perror("error abrir fichjero");

return;

}

while((leido=read(fd,blq,512\*sizeof(char)))>0)

{

if(write(fd,blq,leido\*sizeof(char))<0)

{

perror("error WRITE");

return;

}

}

}

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

{

char ruta[PATH\_MAX],fichero[PATH\_MAX];

int PID,pos,leido,fd;

struct sigaction sa;

if(argc!=2)

{

perror("error Nºargumentos");

exit(1);

}

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

sa.sa\_handler=manejador;

if(sigaction(SIGCHLD,&sa,NULL)<0)//cuando el hijo termine la señal recibira, y se invocara la funcion

{

perror("error sigaction");

exit(1);

}

if(mkfifo("FIFO",0666)<0)

{

perror("error FIFO");

exit(1);

}

/\*int fd[2];

if(pipe(fd)<0)

{

perror("error pipe");

exit(1);

}

\*/

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

{

perror("error FORK");

exit(1);

}

if(PID==0)

{

if((fd=open("FIFO",O\_RDWR))<0)

{

perror("error ABRIR FIFO");

exit(1);

}

//close(fd[0]);

if((dup2(fd,STDOUT\_FILENO))<0)

{

perror("error DUP2....1");

exit(1);

}

execlp("cut","cut","-d:","-f6",argv[1],NULL);

perror("Error execlp....1");

exit(1);

}

else

{

if((fd=open("FIFO",O\_RDONLY))<0)

{

perror("error ABRIR PADRE");

exit(1);

}

//close(fd[1]);

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

{

perror("error DUP2....2");

exit(1);

}\*/

//LEEMOS CARACTER A CARACTER

pos=0;

char c;

while(read(fd,&c,sizeof(char))>0)

{

//voy metiendo caracter a caracter, hasta \n y llamo a procesadir con la cadena caracteres

ruta[pos]=c;

if(ruta[pos]=='\n')

{

ruta[pos]='\0';

procesador(ruta);

}

else

{

pos++;

}

}

close(fd);

}

return 0;

}