#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\_origen.c -o VER

./VER archivo.txt

\*/

void manejador(int sig)

{

printf("HIJO FINALIZADO\n" );

}

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

{

if(argc!=2)

{

perror("Error numero argumentos");

exit(EXIT\_FAILURE);

}

struct sigaction sa;

sa.sa\_handler=manejador;

if(sigaction(SIGCHLD,&sa,NULL)<0) perror("error signation"),exit(1);

int fd[2];

pid\_t hijo1,hijo2;

int estado2;

int estado;

if(pipe(fd)<0)

{

perror("Error en el pipe");

exit(EXIT\_FAILURE);

}

if((hijo1=fork())<0)

{

perror("Error en el fork");

exit(EXIT\_FAILURE);

}

if(hijo1==0)

{

estado=10;

close(fd[0]);

if((dup2(fd[1],STDOUT\_FILENO))<0)

{

perror("Error en el dup2\n");

exit(EXIT\_FAILURE);

}

execlp("ls","ls","-l",argv[1],NULL);

perror("Error execlp");

exit(EXIT\_FAILURE);

}

else

{

if((hijo2=fork())<0)

{

perror("Error en HIJO\n");

exit(EXIT\_FAILURE);

}

if(hijo2==0)

{

estado2=20;

close(fd[1]);

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

{

perror("Error DUP2 padre\n");

exit(EXIT\_FAILURE);

}

execlp("cut","cut","-d",">","-f2",NULL);

perror("Error execlp padre");

exit(EXIT\_FAILURE);//devuelve un 1 y el otro un 0

}

}

close(fd[0]);

close(fd[1]);

//wait(&estado);

//wait(&estado2);

waitpid(hijo1,&estado,NULL);

waitpid(hijo2,&estado2,NULL);

printf("HIJO1--> <%d> ESTADO:<%d>\n",hijo1,WEXITSTATUS(estado));

printf("HIJO2--> <%d> ESTADO:<%d>\n",hijo2,WEXITSTATUS(estado2));

return 0;

}