#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>

void manejador( int iden)

{

int estado,pid;

waitpid(pid,&estado,0);

printf(" El hijo1 con pid-> %d ha terminado con estado %d \n",pid, estado);

}

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

{

if(argc!=2)

{

perror("error argumentos");

exit(EXIT\_FAILURE);

}

if(setvbuf(stdout,NULL,\_IONBF,0)) {//NO SE MUY BIEN LO QUE ES

perror("\nError en setvbuf");

}

struct sigaction sa;

sa.sa\_handler=manejador;

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

int fd,PID;

struct dirent \*entrada;

struct stat atributos;

DIR \*carpeta;

char ruta[PATH\_MAX];

char rutaTipo[PATH\_MAX];

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

{

perror("error argumentos");

exit(EXIT\_FAILURE);

}

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

{

sprintf(ruta,"%s/%s",argv[1],entrada->d\_name);

sprintf(rutaTipo,"%s\_tipo/%s\_tipo",argv[1],entrada->d\_name);//creo que petara pk no se si se crea el NUEVO\_DIRECTORIO

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

{

perror("error ATRIBUTOS");

exit(EXIT\_FAILURE);

}

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

{

perror("error hijo");

exit(EXIT\_FAILURE);

}

if(PID==0)

{

if((fd=open(rutaTipo,O\_CREAT|O\_RDWR,0777))<0)

{

perror("error NUEVO\_DIRECTORIO\n");

exit(EXIT\_FAILURE);

}

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

{

perror("error DUP2");

exit(EXIT\_FAILURE);

}

printf("USUARIO<%d>\n",atributos.st\_uid );

//printf( "SUID:%s\n",atributos.st\_mode & S\_ISUID?"true":"false"); //?????

execlp("file","file",NULL);

perror("error EXCLP");

exit(EXIT\_FAILURE);

}

}

return 0;

}