#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>

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

{

pid\_t PID;

int fd[2];

char mempadre[60];

sprintf(mempadre,"/proc/%d/maps",getppid()); //quiere el pid del shell

if (pipe(fd)<0)

{

perror("Error al crear los descriptores");

exit(10);

}

if((PID=fork())<0) //PUEDO PONER SOLAMENTE PID Y ME AHORRO hijo1 y hijo2

{

perror("Error al crear hijo ");

exit(10);

}

else

{

if(PID!=0) //padre

{

//char c[2]={0,'\0'};

//char c[2];

//c[1]='\0';

char c;

int deboescribir;

close(fd[1]);//no escribo en el cauce

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

{

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

exit(-1);

}\*/

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

{

if( c=='/' ) //el c aqui

{

deboescribir=1;

}

if(deboescribir)

{

printf("%c",c);

//write(STDOUT\_FILENO,c,sizeof(char));

}

if(c =='\n')

deboescribir=0;

}

//wait(NULL);

}

else //hijo

{

close(fd[0]);

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

{

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

exit(-1);

}

execlp("cat","cat",mempadre,NULL);

perror("\nError en el execl hijo1");//cuando se ejecuta bien el execlp no llegaría aquí nunca

exit(-1);

}

}

return 0;

}