pipe\_zhj：实现管道

负责人：章寒捷

时间：2019.6.19

备注：必须包含unistd.h和stdlib.h头文件

用户输入的指令存在char \* cmd[]数组中，输入的参数分别存在param1、param2数组中

/\*\*\*\*\* Info \*\*\*\*\*/

/\* Author: zhj \*/

/\* Function: 实现管道\*/

/\*\*\*\*\* Info \*\*\*\*\*/

#include<stdio.h>

#include <unistd.h>

#include<stdlib.h>

void pipe\_zhj(){

/\* Note：如下是测试用例 \*/

/\* cmd中存放指令序列,cmd[0]为pipe前的指令，cmd为pipe后的指令 \*/

/\* param1存放cmd[0]指令的参数 \*/

/\* param2存放cmd[1]指令的参数 \*/

/\* 测试用例 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

char \* cmd[10];

char \* param1[]={"",0};

char \* param2[]={"-l",".c",0};

cmd[0]="ls";

cmd[1]="grep";

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*1.创建管道\*/

int pid;

int fd[2];//fd[0]读端，fd[1]写端

pipe(fd);

int flag;

/\*2.创建子进程\*/

if((pid=fork())==0){

//子进程，默认为pipe前的指令ls

dup2(fd[1],1);//把标准输出流重定向到管道写端

close(fd[0]);//关闭管道的文件描述符

close(fd[1]);

if((flag=execvp(cmd[0],param1))<0){

printf("son:no such command %d",flag);

}

exit(EXIT\_SUCCESS);

}

else{

//父进程，默认为pipe后的指令

waitpid(pid,NULL,0);

dup2(fd[0],0);//把标准输入流重定向到管道读端

close(fd[0]);

close(fd[1]);

if((flag=execvp(cmd[1],param2))<0){

printf("father:no such command %d",flag);

}

exit(EXIT\_SUCCESS);

}

}