**程序：**实现简单的交互式shell。

**程序功能：**

1、给出提示符，让用户输入一行命令，识别程序名和参数并调用适当的exec函数执行程序，待执行完成后再次给出提示符。

2、该程序可识别和处理以下符号：

1) 简单的标准输入输出重定向：< or >

2) 管道(|): Shell进程先调用pipe创建管道，然后fork出两个子进程。

一个子进程关闭读端，调用dup2将写端赋给标准输出，另一个子进程关闭写端，调用dup2把读端赋给标准输入，

两个子进程分别调用exec执行程序，而Shell进程把管道的两端都关闭，调用wait等待两个子进程终止。

类似于“兄弟进程间ls | wc –l”练习的实现。

3、你的程序应该可以处理以下命令：

○ ls △ -l △ -R ○ > ○ file1 ○

○ cat ○ < ○ file1 ○ | ○ wc △ -c ○ > ○ file1 ○

注：○ 表示零个或多个空格，△ 表示一个或多个空格

**实现思路：**

1.识别字符串中的程序和参数，通过空格进行区分

2.将识别的参数进行整合以便execvp函数进行处理

2.1 没有参数 普通参数 输入输出重定向处理 管道数处理(一个管道意味着两个子进程)

2.2 对参数末尾NULL的处理

3.因为exec函数族如果执行成功是不返回主程序的，所以可以fork出一个子进程将其执行，主进程用wait/waitpid回收即可。

3.1 输入输出重定向的处理需要dup2来修改默认输入输出，结束的时候要改回来

3.2 多个进程之间的IPC通信调用pipe，dup2修改输入和输出默认值，结束后关闭管道读写端

**实现步骤：**

1. 接收用户输入命令字符串，拆分命令及参数存储。（自行设计数据存储结构）

2. 实现普通命令加载功能

3. 实现输入、输出重定向的功能

4. 实现单重管道

5. 支持多重管道

**程序种使用的知识点：**

1.scanf gets fgets

<https://www.cnblogs.com/52php/p/5724372.html>

scanf("%s", &str) 如果输入了 **空格** 会认为输入字符串结束。

gets(str);其中str为字符串变量(字符串数组名或字符串指针)。

从标准输入中读取一行文本，并存储于作为参数传递给它的数组中。一行文本由一串字符组成，以一个换行符('\n')结尾。在返回之前，gets()函数**丢弃回车换行符('\n')**，取而代之的是以'\0'结尾。

gets函数没有限制它所读取长度，程序员应该保证buffer有足够的空间，否则buffer可能无法容纳gets所读取的内容，从而导致堆栈溢出。

fgets()函数，从流中读一行或指定个字符

char \*fgets(char \*s, int n, FILE \*stream);

**功能描述：**

从流中读入 n-1 个字符放入 s 为起始地址的空间内。如果在未读满 n-1 个字符之时，已读到一个换行符（'\n'）或一个EOF（文件结束标志），则结束本次读操作，读入的字符串中最后包含读到的换行符（'\n'）。因此，确切地说，调用 fgets() 函数时，最多只能读入 n-1 个字符。读入结束后，系统将自动在最后加 '\0'，并以 s 作为函数值返回。

**形参注释：**

\*s：结果数据的首地址；

n-1：一次读入数据块的长度，其默认值为1k，即1024；

stream：文件指针

比如，从键盘输入 fgets(s,n,stdin);

2.用于恢复默认值

ofd = dup(STDOUT\_FILENO);

ifd = dup(STDIN\_FILENO);

dup2(ofd, STDOUT\_FILENO);

dup2(ifd, STDIN\_FILENO);

个人bug：

1.命令和参数，参数与参数之间至少有一个空格，没有空格会失败。

解决方法：特殊参数处需要加一些处理。也可能会有统一的处理的方法，待拍砖。

2.多重管道换没实现。cat | cat | cat | cat  
     解决方法：没能实现。多进程管理目前只会sleep来控制fork出的子进程先后顺序。

代码贴在这里：

<https://paste.ubuntu.com/p/g8pk6pkbNS/>