

0.目录

0.目录.....	1
1. 预期功能.....	3
2. 模块细则.....	3
2.1 提示信息模块 <i>infor_print</i> ()	3
2.1.1 使用函数.....	3
2.1.2 思路.....	3
2.2 指令获取与初步判断模块 <i>int checkValid(char* command)</i>	4
2.2.1 使用函数.....	4
2.2.2 编写思路.....	4
2.3 指令切分调整模块 <i>char** commandSplit(char *command)</i>	4
2.3.1 使用函数.....	4
2.3.2 编写思路.....	4
2.4 指令执行模块 <i>sigCommPerform(char** commSplitted)</i>	5
2.4.0 指令类型判断.....	5
2.4.1 普通指令执行 <i>normalExcute(char** commSplitted)</i>	5
2.4.1.1 使用函数.....	5
2.4.1.2 编写思路.....	5
2.4.1.2.1 cd 指令.....	5
2.4.1.2.2 普通指令.....	6
2.4.2 重定向指令的执行 <i>redirectExcute(char** commSplitted, int arrPlace)</i>	6
2.4.2.1 使用函数.....	6
2.4.2.2 编写思路.....	6
2.4.3 管道指令 <i>pipeExcute(char** commandSplitted, int linePlace)</i>	6
2.4.3.1 使用函数和模块.....	7
2.4.3.2 编写思路.....	7
3.数据测试.....	8
3.1 信息提示模块.....	8
3.1.1root 用户.....	8
3.1.2 普通用户.....	8
3.2 指令执行模块.....	8
3.2.1exit 指令.....	8
3.2.2cd 指令.....	9
3.2.3 普通指令.....	9
3.2.3.1 /bin/**执行文件.....	9
3.2.3.2/usr/bin/**执行文件.....	10
3.2.4 重定向指令.....	11
3.2.5 管道指令.....	12
3.2.5.1 单个管道.....	12
3.2.5.2 管道自身嵌套.....	12
3.2.5.3 管道与重定向嵌套.....	13

3.2.6 非正规输入指令测试.....	14
4. 一些遐想.....	14
4.1 执行函数 <code>execv</code> 和 <code>popen</code> 的选择.....	14
4.2 管道嵌套的思考.....	15
4.3 指令执行过程中关于 <code>malloc</code> 的小缺陷.....	16
4.4 管道和重定向选用 <code>ls -al</code> 而非 <code>ps aux</code> 的思考.....	16
4.5 重定向指令对于 <code>fflush()</code> 的使用.....	16

1. 预期功能

1)设计一个 C 语言程序，完成最基本的 shell 角色：给出 命令行提示符、能够逐次接受命令； 对于命令分成三种

内部命令（例如 help 命令、exit 命令等）

外部命令（常见的 ls、cp 等，以及其他磁盘上的可执行程序 HelloWrold 等）

无效命令（不是上述二种命令

2)具有支持管道的功能，即在 shell 中输入诸如 “dir | more”能够执行 dir 命令并将其输出通过管道将 其输入传送给 more。

3)具有支持重定向的功能，即在 shell 中输入诸如“dir > direct.txt”能够执行 dir 命令并将结果输出到 direct.txt

4)将上述步骤直接合并完

2. 模块细则

2.1 提示信息模块 *infor_print* ()

该部分主要掌管当前 shell 的运行信息的获取与呈现。

主要涉及到：用户登录名，用户 host 名，当前 shell 工作目录，当前用户权限部分
该模块不需要传入参数，为自获取自显示

2.1.1 使用函数

```
1. getlogin();//Log in name
2. gethostname(hostName, 256);//Log in machine name
3. getcwd(workSpace,256);//current working place
4. getuid();//current uid to get the level mark
5. printf("\033[1m\033[32m%s@\033[37m:\033[31m%s\033[37m%c\033[0m \033[5m",userName, hostName,workSpace, userLevel);
```

2.1.2 思路

通过各个信息获取函数进行信息的获取。之后同过 printf 函数及其中的/\033[**m 进行打印格式的颜色格式规定

2.2 指令获取与初步判断模块 *int checkValid(char* command)*

该部分负责命令行指令的输入和指令的非空性判断

具体该指令的执行情况交由指令执行的模块处理

初步判断模块需要传入的参数为：获取到的，未切分的原指令 *command*。当判定为非空，则返回 1，反之返回 0

2.2.1 使用函数

```
1.  getline(&command, &length, stdin);  
2.  strcmp(command, "exit\n")
```

2.2.2 编写思路

使用 `getline` 函数获取整行的输入之后，进行初步的比较。当为 `exit` 指令则直接处理，跳出命令获取循环。若不是则送入 *checkValid* 模块，当判断不是空指令则继续执行。

2.3 指令切分调整模块 *char** commandSplit(char* command)*

该部分负责指令的格式调整和整体切分。

主要涉及到：非规范输入的调整，非必要输入的剔除，指令的切割

需要传入的参数为：未切分的原指令 *command*。其会返回调整后切割好的字符数组

2.3.1 使用函数

```
1.  strtok(char*, " ");
```

2.3.2 编写思路

每个命令将以字符“\n”结尾。在此之前，如果有|，&，；，>，并且附近没有空格，添加空格，然后继续调整。如果有<，则将其视为正常空白。

之后将第一次调整完的指令进行二次调整，将所有/t符号掠过，所有相连的空格符只保留一个。

最后，将调整好的函数调用 `strtok` 函数进行以空格符为标志的切分并进行存储。将切割好的指令全部返回即可。

2.4 指令执行模块 *sigCommPerform(char** commSplitted)*

该部分主要涉及到切割好指令的执行

主要包括：普通指令执行，重定向指令的执行，管道指令的执行。

在这个模块当中对切割好的终端指令进行类型判断。然后将指令传递给各自的执行模块自行处理

需要传入：分割好的指令 *commSplitted*

2.4.0 指令类型判断

遍历一遍切割好的函数的各头字符。当有 > 和 | 符号时，且为第一次遇到的时候，将位置进行记录。随后，根据符号的类型，两者都没有的时候则调用 *normalExcute*；当存在 > 而不存在 | 的时候，调用 *redirectExcute*；只要有 | 符号，则调用 *pipeExcute*。

2.4.1 普通指令执行 *normalExcute(char** commSplitted)*

该模块负责可以一次执行完成的内部指令

包括但不限于 ls, echo, cd 等

同时该部分兼有执行 < 符号的重定向指令功能

需要传入：切割好的单个指令 *commSplitted*

2.4.1.1 使用函数

```
1.  strcmp(commSplitted[0], "cd")
2.  chdir(commSplitted[1])
3.  getcwd(currentDir, 256)
4.  strcat(currentDir, commSplitted[1])
5.  chdir(currentDir)
6.  fork()
7.  wait(NULL)
8.  execv(ecvFile, commSplitted );
```

2.4.1.2 编写思路

2.4.1.2.1 cd 指令

当对指令的类型判断为 cd 的时候，调用 chdir 指令进行工作位置的切换。

考虑到有时候输入的并不是规范的带 / 符号的文件名。当直接按照参数切换失败的时

候，当该路径符号首位不为 / 时，为该路径参数添加 / 符号。之后将其与当前工作文件夹的目录进行组合生成绝对路径，进行再次切换。

2.4.1.2.2 普通指令

当为普通的内部指令的时候，将 `fork()` 一个子进程。使用第一位字符向量来形成执行文件路径，将组合好的路径和所需要执行的命令切割符号传入 `execv()` 函数即可。

如果 “/bin/***” 失败，则尝试 “/usr/bin/***” 执行路径

2.4.2 重定向指令的执行 *redirectExcute(char** commSplitted, int arrPlace)*

该模块负责含有 > 符号的输出重定向指令

具有 < 的输入重定向指令交由 *normalExcute*

需要传入：分割好的重定向指令 *commSplitted*，分割好的重定向指令中重定向符号的箭头位置 *arrPlace*。

2.4.2.1 使用函数

```
1. fopen(commSplitted[arrPlace+1], "w+");
2. popen(command, "r");//critical execute function
3. fgetc(result);
4. fputc(bridge, output);
5. fflush(output);
```

2.4.2.2 编写思路

在 *arrplace* 之前，是需要执行的命令，之后是存储结果的文件。

将拆分需要执行的命令恢复为一个完整的命令。由于此时需要将执行的结果转存则不调用 `execv` 函数，而使用 `popen` 执行 *commad*，然后将结果转存到目标文件中。

由于输入重定向向实际等效于普通指令，则在指令过滤分割的过程中已经调整为普通命令。由于重定向的执行结果实际为文件，不可能进行其自身的输出重定向嵌套，则只使用一次即可。

2.4.3 管道指令 *pipeExcute(char** commandSplitted, int linePlace)*

该部分负责带有 | 符号的指令的执行。

需要传入：切割好的管道指令 `commandSplitted`，管道命令的管道符号位置

2.4.3.1 使用函数和模块

```
1.  redirectExcute(sigSplittedCom, arrPlace);
2.  fopen(commSplitted[arrPlace+1], "w+");
3.  popen(command, "r");//critical execute function
4.  fgetc(result);
5.  fputc(bridge, output);
6.  fflush(output);
7.  getcwd(filePath, 256);
8.  strcat(filePath, "/former");
9.  sigCommPerform(newSplittedComm);
10. pipeExcute(newSplittedComm, newLinePlace);
```

2.4.3.2 编写思路

对于重定向命令，`arrPlace` 后面的位置是需要的存储结果的文件，它已经由重定向执行过程存储完成，只需打开它并将其存储到临时文件“former”即可。其可以在后续的命令中用作文件参数。

对于非重定向的命令，将拆分的命令恢复为一个完整的命令，使用 `popen()` 执行命令，并将结果存储在临时文件“current”中，然后将结果转存到文件“former”。former 的路径可以在后续的命令中用作执行参数输入。

由于管道会存在多次嵌套，则每次只取下一个管道符之前的命令作为需要执行的命令，剩下的内容作为下一次需要执行的命令进行嵌套。

在生成嵌套命令的过程中：

对于重定向命令的执行文件添加，在转存下一次切分指令的过程中，当第一次遇到重定向指令的时候将 `former` 文件的路径添加作为参数，然后再继续存储

作为非重定向指令，在转存下一次切分指令的过程中，当第一次遇到管道符的时候将 `former` 文件的路径添加作为参数，然后再继续存储。若走完了命令都没有添加文件路径，则将其添加到指令末尾。

若在过程中既没有发现 `>` 符号，也没有发现 `|` 符号，则将其当作普通指令，调用 `sigCommPerform` 进行执行。

3.数据测试

3.1 信息提示模块

3.1.1root 用户

系统终端

```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
honorifica@honorifica-virtual-machine:~$ su root
密码:
root@honorifica-virtual-machine:/home/honorifica#
```

自编程终端

```
root@honorifica-virtual-machine:/home/honorifica#
root@honorifica-virtual-machine:/home/honorifica# '/home/honorifica/OS_LessonDesign/jyqshell'
honorifica@honorifica-virtual-machine:/home/honorifica#
```

3.1.2 普通用户

系统终端

```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
honorifica@honorifica-virtual-machine:~$
```

自编程 shell

```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
honorifica@honorifica-virtual-machine:~$ '/home/honorifica/OS_LessonDesign/jyqshell'
honorifica@honorifica-virtual-machine:/home/honorifica$
```

3.2 指令执行模块

3.2.1exit 指令

```
hell
honorifica@honorifica-virtual-machine:/home/honorifica$exit
honorifica@honorifica-virtual-machine:~$
```

shell 正常退出

3.2.2cd 指令

```
hell'  
honorifica@honorifica-virtual-machine:/home/honorifica$ cd /home  
honorifica@honorifica-virtual-machine:/home$cd honorifica  
honorifica@honorifica-virtual-machine:/home/honorifica$
```

工作目录正常切换，且提示信息变化正常。

对于正规的带斜杠的目录和非正规的无斜杠目录接可以进行切换。

3.2.3 普通指令

3.2.3.1 /bin/**执行文件

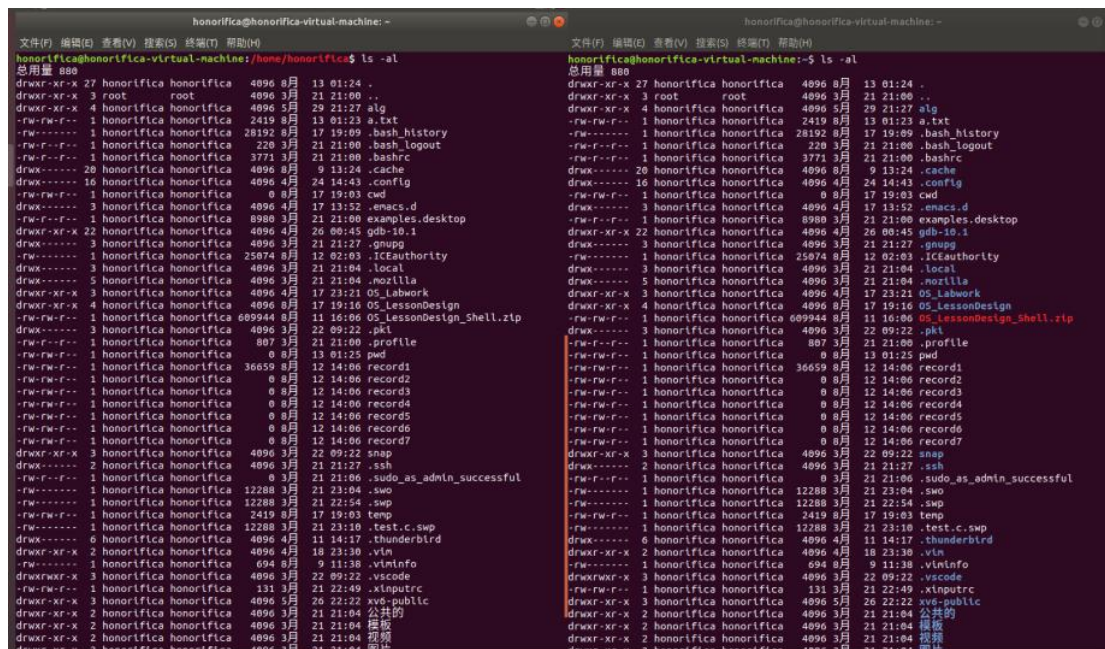
对于 which 指令，可用于查看当前指令的执行文件



```
honorifica@honorifica-virtual-machine:/home/honorifica$ which ls  
/bin/ls  
honorifica@honorifica-virtual-machine:/home/honorifica$ which echo  
/bin/echo  
honorifica@honorifica-virtual-machine:/home/honorifica$ which bash  
/bin/bash  
honorifica@honorifica-virtual-machine:/home/honorifica$ which grep  
/bin/grep  
honorifica@honorifica-virtual-machine:/home/honorifica$ which touch  
/usr/bin/touch  
honorifica@honorifica-virtual-machine:/home/honorifica$
```

该命令运行正常

对于 ls 指令：



```
honorifica@honorifica-virtual-machine:/home/honorifica$ ls -al  
总用量 880  
drwxr-xr-x 27 honorifica honorifica 4096 8月 13 01:24 .  
drwxr-xr-x 3 root root 21:00 ..  
drwxr-xr-x 4 honorifica honorifica 4096 5月 29 21:27 alg  
-rw-rw-r-- 1 honorifica honorifica 2419 8月 13 01:23 a.txt  
-rw-r--r-- 1 honorifica honorifica 28192 8月 17 19:09 .bash_history  
-rw-r--r-- 1 honorifica honorifica 220 3月 21 21:00 .bash_logout  
-rw-r--r-- 1 honorifica honorifica 3771 3月 21 21:00 .bashrc  
drwx----- 20 honorifica honorifica 4096 8月 9 13:24 .cache  
drwx----- 16 honorifica honorifica 4096 4月 24 14:43 .config  
-rw-rw-r-- 1 honorifica honorifica 0 8月 17 19:03 cwd  
drwx----- 3 honorifica honorifica 4096 4月 17 13:52 .enacs.d  
-rw-r--r-- 1 honorifica honorifica 8980 3月 21 21:00 examples.desktop  
drwxr-xr-x 22 honorifica honorifica 4096 4月 26 00:45 gdb-10.1  
drwx----- 3 honorifica honorifica 4096 3月 21 21:27 .gnupg  
-rw-r--r-- 1 honorifica honorifica 25874 8月 12 02:03 .ICEauthority  
drwx----- 3 honorifica honorifica 4096 3月 21 21:04 .local  
drwxr-xr-x 3 honorifica honorifica 4096 4月 17 23:21 OS_Labwork  
drwxr-xr-x 4 honorifica honorifica 4096 8月 17 19:16 OS_LessonDesign  
-rw-rw-r-- 1 honorifica honorifica 609944 8月 11 16:06 OS_LessonDesign_Shell.zip  
drwx----- 3 honorifica honorifica 4096 3月 22 09:22 .pkd  
-rw-rw-r-- 1 honorifica honorifica 807 3月 21 21:00 .profile  
-rw-rw-r-- 1 honorifica honorifica 0 8月 13 01:25 pwd  
-rw-rw-r-- 1 honorifica honorifica 36659 8月 12 14:06 record1  
-rw-rw-r-- 1 honorifica honorifica 0 8月 12 14:06 record2  
-rw-rw-r-- 1 honorifica honorifica 0 8月 12 14:06 record3  
-rw-rw-r-- 1 honorifica honorifica 0 8月 12 14:06 record4  
-rw-rw-r-- 1 honorifica honorifica 0 8月 12 14:06 record5  
-rw-rw-r-- 1 honorifica honorifica 0 8月 12 14:06 record6  
-rw-rw-r-- 1 honorifica honorifica 0 8月 12 14:06 record7  
drwxr-xr-x 3 honorifica honorifica 4096 3月 22 09:22 snap  
drwx----- 2 honorifica honorifica 4096 3月 21 21:27 .ssh  
-rw-rw-r-- 1 honorifica honorifica 0 3月 21 21:06 .sudo_as_admin_successful  
-rw-r--r-- 1 honorifica honorifica 12288 3月 21 23:04 .swp  
-rw-r--r-- 1 honorifica honorifica 12288 3月 21 22:54 .swp  
-rw-rw-r-- 1 honorifica honorifica 2419 8月 17 19:03 temp  
-rw-r--r-- 1 honorifica honorifica 12288 3月 21 23:10 .test.c.swp  
drwx----- 6 honorifica honorifica 4096 4月 11 14:17 .thunderbird  
drwxr-xr-x 3 honorifica honorifica 4096 4月 18 23:30 .vim  
-rw-r--r-- 1 honorifica honorifica 694 8月 9 11:38 .viminfo  
drwxrwxr-x 3 honorifica honorifica 4096 3月 22 09:22 .vscode  
-rw-rw-r-- 1 honorifica honorifica 131 3月 21 22:49 .xinputrc  
drwxr-xr-x 3 honorifica honorifica 4096 3月 26 22:22 xvs-public  
drwxr-xr-x 3 honorifica honorifica 4096 3月 21 21:04 公共的  
drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 模板  
drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 视频  
drwxr-xr-x 3 honorifica honorifica 4096 3月 21 21:04 网络
```

可见其运行结果与自带终端一致

对于 echo 指令

```
honorfica@honorfica-virtual-machine:/home/honorfica$ echo this_is_a_test
this_is_a_test
honorfica@honorfica-virtual-machine:/home/honorfica$ echo bibi, testing shell
bibi, testing shell
honorfica@honorfica-virtual-machine:/home/honorfica$ echo 123,321,pupu testing
123,321,pupu testing
honorfica@honorfica-virtual-machine:/home/honorfica$
```

可见该指令运行正常。

对于 bash 指令

```
honorfica@honorfica-virtual-machine:/home/honorfica$ bash bash1.sh
this is bash1.sh
a.txt      infor_print.c  pwd          record5      sigConnPerform.c
bash1.sh   jyqshell       record1      record6      temp
bash1.sh-  jyqshell.c     record2      record7      temp1
checkValid.c normalExcute.c record3      records      temp2
commandSplit.c pipeExcute.c  record4      redirectExcute.c temp3
honorfica@honorfica-virtual-machine:/home/honorfica$

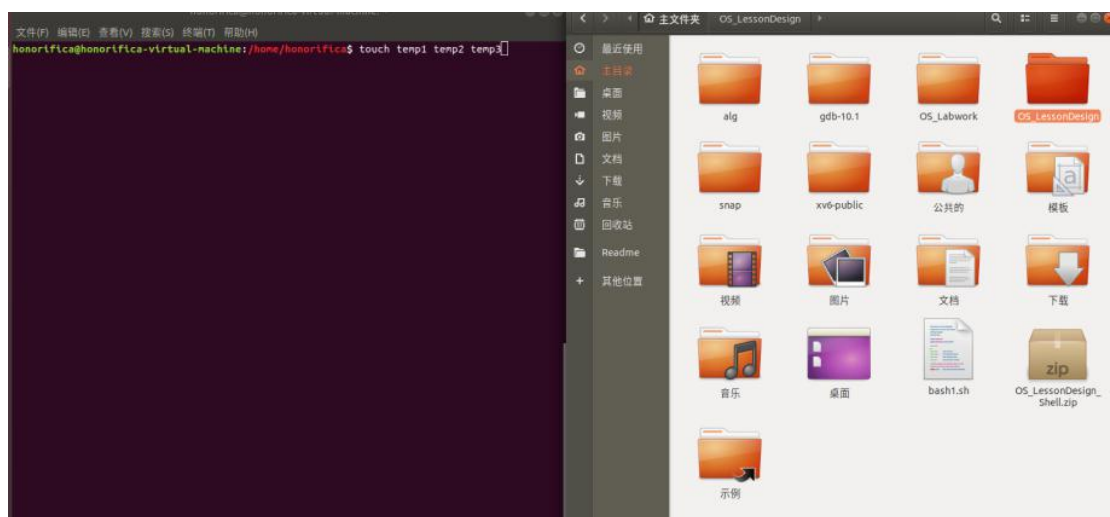
honorfica@honorfica-virtual-machine:/home/honorfica$ bash bash1.sh
this is bash1.sh
a.txt      infor_print.c  pwd          record5      sigConnPerform.c
bash1.sh   jyqshell       record1      record6      temp
bash1.sh-  jyqshell.c     record2      record7      temp1
checkValid.c normalExcute.c record3      records      temp2
commandSplit.c pipeExcute.c  record4      redirectExcute.c temp3
honorfica@honorfica-virtual-machine:/home/honorfica$ cd OS_LessonDesign
honorfica@honorfica-virtual-machine:/home/honorfica/OS_LessonDesign$ ls
a.txt      infor_print.c  pwd          record5      sigConnPerform.c
bash1.sh   jyqshell       record1      record6      temp
bash1.sh-  jyqshell.c     record2      record7      temp1
checkValid.c normalExcute.c record3      records      temp2
commandSplit.c pipeExcute.c  record4      redirectExcute.c temp3
honorfica@honorfica-virtual-machine:/home/honorfica/OS_LessonDesign$
```

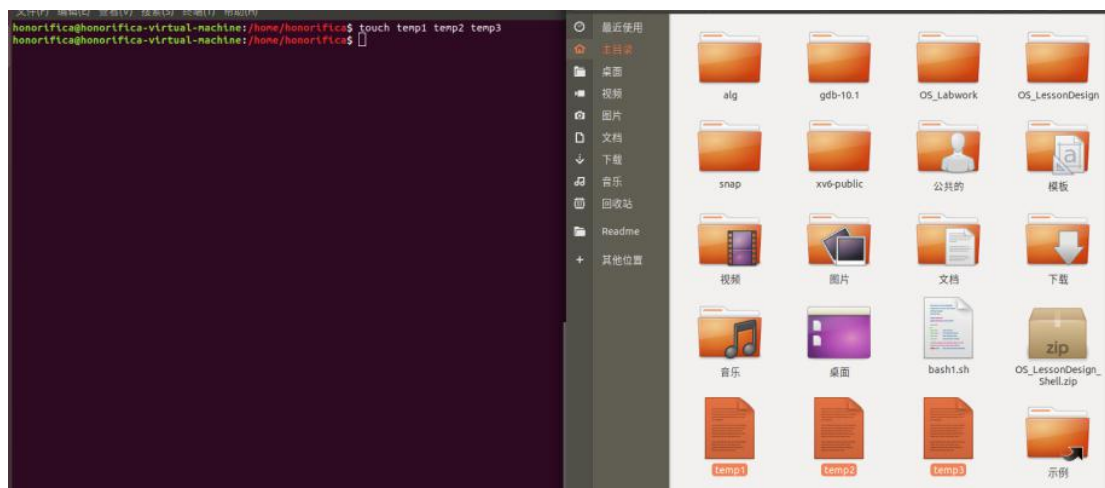
经验证，文件执行正确

经过测试，该路径的文件执行正确

3.2.3.2/usr/bin/**执行文件

对于指令 touch 其会新建文件名为各参数的文件于工作目录下





经过测试，该执行文件目录执行正确

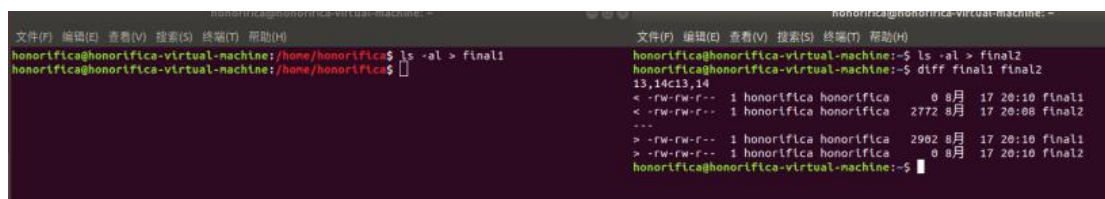
3.2.4 重定向指令

重定向测试指令为：

自编程终端： `ls -al > final1`

系统终端： `ls -al > final2`

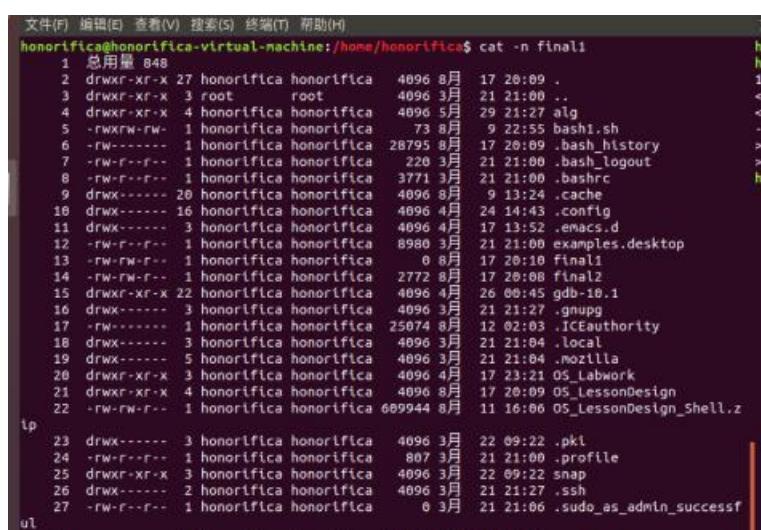
其中 `final1` 和 `final2` 文件已经事先新建完成



发现两个文件的记录只有自身不一样

则指令执行成功

同时调用 `cat -n final1` 对结果进行查看与编号：



指令可以正常执行

并且同自带终端的运行结果一致


```
honorificahonorifica-virtual-machine:/home/honorifica$ cat -n final1
1 总用量 848
2 drwxr-xr-x 27 honorifica honorifica 4096 8月 17 20:09 .
3 drwxr-xr-x 3 root root 4096 3月 21 21:00 ..
4 drwxr-xr-x 4 honorifica honorifica 4096 5月 29 21:27 alg
5 -rwxrwxrwx 1 honorifica honorifica 73 8月 9 22:55 bash1.sh
6 -rw-r----- 1 honorifica honorifica 28795 8月 17 20:09 .bash_history
7 -rw-r--r-- 1 honorifica honorifica 220 3月 21 21:00 .bash_logout
8 -rw-r--r-- 1 honorifica honorifica 3771 3月 21 21:00 .bashrc
9 drwx----- 20 honorifica honorifica 4096 8月 9 13:24 .cache
10 drwx----- 16 honorifica honorifica 4096 4月 24 14:43 .config
11 drwx----- 3 honorifica honorifica 4096 4月 17 13:52 .emacs.d
12 -rw-r--r-- 1 honorifica honorifica 8980 3月 21 21:00 examples.desktop
13 -rw-rw-r-- 1 honorifica honorifica 0 8月 17 20:10 final1
14 -rw-rw-r-- 1 honorifica honorifica 2772 8月 17 20:08 final2
15 drwxr-xr-x 22 honorifica honorifica 4096 4月 26 00:45 gdb-10.1
16 drwx----- 3 honorifica honorifica 4096 3月 21 21:27 .gnupg
17 -rw-r----- 1 honorifica honorifica 25974 6月 12 02:03 .iccauthority
18 drwx----- 3 honorifica honorifica 4096 3月 21 21:04 .local
19 drwx----- 5 honorifica honorifica 4096 3月 21 21:04 .mozilla
20 drwxr-xr-x 3 honorifica honorifica 4096 4月 17 23:21 05_Labwork
21 drwxr-xr-x 4 honorifica honorifica 4096 8月 17 20:09 05_LessonDesign
22 -rw-rw-r-- 1 honorifica honorifica 689944 8月 11 16:06 05_LessonDesign_Shell.z
ip
23 drwx----- 3 honorifica honorifica 4096 3月 22 09:22 .pkt
24 -rw-r--r-- 1 honorifica honorifica 807 3月 21 21:00 .profile
25 drwxr-xr-x 3 honorifica honorifica 4096 3月 22 09:22 snap
26 drwx----- 2 honorifica honorifica 4096 3月 21 21:27 .ssh
27 -rw-r--r-- 1 honorifica honorifica 0 3月 21 21:06 .sudo_as_admin_successf
ut
28 -rw-r----- 1 honorifica honorifica 12288 3月 21 23:04 .swp
29 -rw-r----- 1 honorifica honorifica 12288 3月 21 22:54 .swp
30 -rw-r----- 1 honorifica honorifica 12288 3月 21 23:10 .test.c.swp
31 drwx----- 6 honorifica honorifica 4096 4月 11 14:17 .thunderbird
32 drwxr-xr-x 2 honorifica honorifica 4096 4月 18 23:30 .vim
33 -rw-r----- 1 honorifica honorifica 694 8月 9 11:38 .viminfo
34 drwxrwxr-x 3 honorifica honorifica 4096 3月 22 09:22 .vscode
35 -rw-rw-r-- 1 honorifica honorifica 131 3月 21 22:49 .xinputrc
36 drwxr-xr-x 3 honorifica honorifica 4096 5月 26 22:22 xv6-public
37 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 公共的
38 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 模板
39 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 视频
40 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 图片
41 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 文档
42 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 下载
43 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 音乐
44 drwx----- 3 honorifica honorifica 4096 8月 6 14:09 桌面
honorificahonorifica-virtual-machine:/home/honorifica$

honorificahonorifica-virtual-machine:~$ cat -n final1
1 总用量 848
2 drwxr-xr-x 27 honorifica honorifica 4096 8月 17 20:09 .
3 drwxr-xr-x 3 root root 4096 3月 21 21:00 ..
4 drwxr-xr-x 4 honorifica honorifica 4096 5月 29 21:27 alg
5 -rwxrwxrwx 1 honorifica honorifica 73 8月 9 22:55 bash1.sh
6 -rw-r----- 1 honorifica honorifica 28795 8月 17 20:09 .bash_history
7 -rw-r--r-- 1 honorifica honorifica 220 3月 21 21:00 .bash_logout
8 -rw-r--r-- 1 honorifica honorifica 3771 3月 21 21:00 .bashrc
9 drwx----- 20 honorifica honorifica 4096 8月 9 13:24 .cache
10 drwx----- 16 honorifica honorifica 4096 4月 24 14:43 .config
11 drwx----- 3 honorifica honorifica 4096 4月 17 13:52 .emacs.d
12 -rw-r--r-- 1 honorifica honorifica 8980 3月 21 21:00 examples.desktop
13 -rw-rw-r-- 1 honorifica honorifica 0 8月 17 20:10 final1
14 -rw-rw-r-- 1 honorifica honorifica 2772 8月 17 20:08 final2
15 drwxr-xr-x 22 honorifica honorifica 4096 4月 26 00:45 gdb-10.1
16 drwx----- 3 honorifica honorifica 4096 3月 21 21:27 .gnupg
17 -rw-r----- 1 honorifica honorifica 25974 6月 12 02:03 .iccauthority
18 drwx----- 3 honorifica honorifica 4096 3月 21 21:04 .local
19 drwx----- 5 honorifica honorifica 4096 3月 21 21:04 .mozilla
20 drwxr-xr-x 3 honorifica honorifica 4096 4月 17 23:21 05_Labwork
21 drwxr-xr-x 4 honorifica honorifica 4096 8月 17 20:09 05_LessonDesign
22 -rw-rw-r-- 1 honorifica honorifica 689944 8月 11 16:06 05_LessonDesign_Shell.z
ip
23 drwx----- 3 honorifica honorifica 4096 3月 22 09:22 .pkt
24 -rw-r--r-- 1 honorifica honorifica 807 3月 21 21:00 .profile
25 drwxr-xr-x 3 honorifica honorifica 4096 3月 22 09:22 snap
26 drwx----- 2 honorifica honorifica 4096 3月 21 21:27 .ssh
27 -rw-r--r-- 1 honorifica honorifica 0 3月 21 21:06 .sudo_as_admin_successf
ut
28 -rw-r----- 1 honorifica honorifica 12288 3月 21 23:04 .swp
29 -rw-r----- 1 honorifica honorifica 12288 3月 21 22:54 .swp
30 -rw-r----- 1 honorifica honorifica 12288 3月 21 23:10 .test.c.swp
31 drwx----- 6 honorifica honorifica 4096 4月 11 14:17 .thunderbird
32 drwxr-xr-x 2 honorifica honorifica 4096 4月 18 23:30 .vim
33 -rw-r----- 1 honorifica honorifica 694 8月 9 11:38 .viminfo
34 drwxrwxr-x 3 honorifica honorifica 4096 3月 22 09:22 .vscode
35 -rw-rw-r-- 1 honorifica honorifica 131 3月 21 22:49 .xinputrc
36 drwxr-xr-x 3 honorifica honorifica 4096 5月 26 22:22 xv6-public
37 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 公共的
38 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 模板
39 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 视频
40 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 图片
41 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 文档
42 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 下载
43 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 音乐
44 drwx----- 3 honorifica honorifica 4096 8月 6 14:09 桌面
honorificahonorifica-virtual-machine:~$
```

3.2.5 管道指令

3.2.5.1 单个管道

测试指令为 `cat -n final1 | grep root`

即为从 `final1` 文件中带编号的输出，再筛选结果中带有 `root` 的内容输出。

终端结果：

```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
honorificahonorifica-virtual-machine:~$ cat -n final1 | grep root
3 drwxr-xr-x 3 root root 4096 3月 21 21:00 ..
honorificahonorifica-virtual-machine:~$
```

自编终端结果：

```
44 drwx----- 3 honorifica honorifica 4096 8月 6 14:09 桌面
honorificahonorifica-virtual-machine:/home/honorifica$ cat -n final1 | grep root
3 drwxr-xr-x 3 root root 4096 3月 21 21:00 ..
honorificahonorifica-virtual-machine:/home/honorifica$
```

单个管道运行良好

3.2.5.2 管道自身嵌套

嵌套指令为：`cat final1 | grep honorifica | grep 21 | cat -n`

指令意义为：从 `final1` 文件中输出内容，筛选出带有 `honorifa` 的表达，再筛选出其中带有 `21` 的表达之后，编号在控制台输出

```
honorifica@honorifica-virtual-machine: /home/honorifica$ cat final1 | grep honorifica | grep 21 | cat -n
1 drwxr-xr-x 4 honorifica honorifica 4096 5月 29 21:27 alg
2 -rw-r--r-- 1 honorifica honorifica 220 3月 21 21:00 .bash_logout
3 -rw-r--r-- 1 honorifica honorifica 3771 3月 21 21:00 .bashrc
4 -rw-r--r-- 1 honorifica honorifica 8980 3月 21 21:00 examples.deskt
op
5 drwx----- 3 honorifica honorifica 4096 3月 21 21:27 .gnupg
6 drwx----- 3 honorifica honorifica 4096 3月 21 21:04 .local
7 drwx----- 5 honorifica honorifica 4096 3月 21 21:04 .mozilla
8 drwxr-xr-x 3 honorifica honorifica 4096 4月 17 23:21 OS_Labwork
9 -rw-r--r-- 1 honorifica honorifica 807 3月 21 21:00 .profile
10 drwx----- 2 honorifica honorifica 4096 3月 21 21:27 .ssh
11 -rw-r--r-- 1 honorifica honorifica 0 3月 21 21:06 .sudo_as_admin
_successful
12 -rw----- 1 honorifica honorifica 12288 3月 21 23:04 .swp
13 -rw----- 1 honorifica honorifica 12288 3月 21 22:54 .swp
14 -rw----- 1 honorifica honorifica 12288 3月 21 23:10 .test.c.swp
15 -rw-rw-r-- 1 honorifica honorifica 131 3月 21 22:49 .xinputrc
16 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 公共的
17 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 模板
18 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 视频
19 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 图片
20 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 文档
21 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 下载
22 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 音乐
honorifica@honorifica-virtual-machine: /home/honorifica$

honorifica@honorifica-virtual-machine:~$ cat final1 | grep honorifica | grep 21 | cat -n
1 drwxr-xr-x 4 honorifica honorifica 4096 5月 29 21:27 alg
2 -rw-r--r-- 1 honorifica honorifica 220 3月 21 21:00 .bash_logout
3 -rw-r--r-- 1 honorifica honorifica 3771 3月 21 21:00 .bashrc
4 -rw-r--r-- 1 honorifica honorifica 8980 3月 21 21:00 examples.deskt
op
5 drwx----- 3 honorifica honorifica 4096 3月 21 21:27 .gnupg
6 drwx----- 3 honorifica honorifica 4096 3月 21 21:04 .local
7 drwx----- 5 honorifica honorifica 4096 3月 21 21:04 .mozilla
8 drwxr-xr-x 3 honorifica honorifica 4096 4月 17 23:21 OS_Labwork
9 -rw-r--r-- 1 honorifica honorifica 807 3月 21 21:00 .profile
10 drwx----- 2 honorifica honorifica 4096 3月 21 21:27 .ssh
11 -rw-r--r-- 1 honorifica honorifica 0 3月 21 21:06 .sudo_as_admin
_successful
12 -rw----- 1 honorifica honorifica 12288 3月 21 23:04 .swp
13 -rw----- 1 honorifica honorifica 12288 3月 21 22:54 .swp
14 -rw----- 1 honorifica honorifica 12288 3月 21 23:10 .test.c.swp
15 -rw-rw-r-- 1 honorifica honorifica 131 3月 21 22:49 .xinputrc
16 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 公共的
17 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 模板
18 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 视频
19 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 图片
20 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 文档
21 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 下载
22 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 音乐
honorifica@honorifica-virtual-machine:~$
```

经过测试，嵌套的结果同系统终端的结果一致

3.2.5.3 管道与重定向嵌套

采用指令为

个人编写终端使用：`cat final1 | grep honorifica | grep 4096 | cat -n > final3`

指令意义为：从 final1 文件中输出内容，筛选出带有 honorifica 的表达，再筛选出其中带有 4096 的表达之后，将编号之后的数据存入到 final3 文件当中

系统自带终端采用：`cat final1 | grep honorifica | grep 4096 | cat -n > final4`

指令意义一致，只是将结果存入 final4 当中

执行结果：

```
honorifica@honorifica-virtual-machine:~$ cat final1 | grep honorifica | grep 4096 | cat -n > final3
honorifica@honorifica-virtual-machine:~$ diff final3 final4
honorifica@honorifica-virtual-machine:~$
```

可见在客户端调用了 diff 函数之后，两个文件没有差异。执行结果为一一致

```
1 drwxr-xr-x 27 honorifica honorifica 4096 8月 17 20:09 .
2 drwxr-xr-x 4 honorifica honorifica 4096 5月 29 21:27 alg
3 drwx----- 20 honorifica honorifica 4096 8月 9 13:24 .cache
4 drwx----- 16 honorifica honorifica 4096 4月 24 14:43 .config
5 drwx----- 3 honorifica honorifica 4096 4月 17 13:52 .emacs.d
6 drwxr-xr-x 22 honorifica honorifica 4096 4月 26 00:45 gdb-10.1
7 drwx----- 3 honorifica honorifica 4096 3月 21 21:27 .gnupg
8 drwx----- 3 honorifica honorifica 4096 3月 21 21:04 .local
9 drwx----- 5 honorifica honorifica 4096 4月 17 23:21 OS_Labwork
10 drwxr-xr-x 4 honorifica honorifica 4096 8月 17 20:09 OS_LessonDesign
11 drwx----- 3 honorifica honorifica 4096 3月 22 09:22 .pkg
12 drwxr-xr-x 3 honorifica honorifica 4096 3月 22 09:22 snap
13 drwx----- 2 honorifica honorifica 4096 3月 21 21:27 .ssh
14 drwx----- 6 honorifica honorifica 4096 4月 11 14:17 .thunderbird
15 drwxr-xr-x 2 honorifica honorifica 4096 4月 18 23:30 .vln
16 drwxr-xr-x 3 honorifica honorifica 4096 3月 22 09:22 .vscode
17 drwxr-xr-x 3 honorifica honorifica 4096 5月 26 22:22 xv6-public
18 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 公共的
19 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 模板
20 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 视频
21 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 图片
22 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 文档
23 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 下载
24 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 音乐
25 drwx----- 3 honorifica honorifica 4096 8月 6 14:09 桌面
26 drwx----- 3 honorifica honorifica 4096 8月 6 14:09 桌面

1 drwxr-xr-x 27 honorifica honorifica 4096 8月 17 20:09 .
2 drwxr-xr-x 4 honorifica honorifica 4096 5月 29 21:27 alg
3 drwx----- 20 honorifica honorifica 4096 8月 9 13:24 .cache
4 drwx----- 16 honorifica honorifica 4096 4月 24 14:43 .config
5 drwx----- 3 honorifica honorifica 4096 4月 17 13:52 .emacs.d
6 drwxr-xr-x 22 honorifica honorifica 4096 4月 26 00:45 gdb-10.1
7 drwx----- 3 honorifica honorifica 4096 3月 21 21:27 .gnupg
8 drwx----- 3 honorifica honorifica 4096 3月 21 21:04 .local
9 drwx----- 5 honorifica honorifica 4096 4月 17 23:21 OS_Labwork
10 drwxr-xr-x 4 honorifica honorifica 4096 8月 17 20:09 OS_LessonDesign
11 drwx----- 3 honorifica honorifica 4096 3月 22 09:22 .pkg
12 drwxr-xr-x 3 honorifica honorifica 4096 3月 22 09:22 snap
13 drwx----- 2 honorifica honorifica 4096 3月 21 21:27 .ssh
14 drwx----- 6 honorifica honorifica 4096 4月 11 14:17 .thunderbird
15 drwxr-xr-x 2 honorifica honorifica 4096 4月 18 23:30 .vln
16 drwxr-xr-x 3 honorifica honorifica 4096 3月 22 09:22 .vscode
17 drwxr-xr-x 3 honorifica honorifica 4096 5月 26 22:22 xv6-public
18 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 公共的
19 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 模板
20 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 视频
21 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 图片
22 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 文档
23 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 下载
24 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 音乐
25 drwx----- 3 honorifica honorifica 4096 8月 6 14:09 桌面
26 drwx----- 3 honorifica honorifica 4096 8月 6 14:09 桌面
```

这同直接打开文件查看的粗略结果一致

3.2.6 非正规输入指令测试

个人编写终端使用:

```
cat final1|grep honorifica'\t''\t'' ' ' ' '\t'|grep 4096|'\t''\t''\t''\t'' ' ' ' ' ' 'cat -n >
final5' '\t'' '\t'''
```

指令意义为: 从 **final1** 文件中输出内容, 筛选出带有 **honorifa** 的表达, 再筛选出其中带有 **4096** 的表达之后, 将编号之后的数据存入到 **final5** 文件当中

其涉及到：空格干扰，tab 干扰，无分隔干扰

其执行结果应当同 `final3` 一致

测试结果如下：

```

honorific@honorifica-ncrui-machine:~$ diff final3 final5
honorific@honorifica-virtual-machine:~/home/honorific$ cat final1 | grep honorific
cat -n > f
honorific@honorifica-virtual-machine:~/home/honorific$

```

可见执行结果正确且吻合，对于干扰能够滤除

打开(O) ▾	Final3	保存(S) ▾	打开(O) ▾	Final5	保存(S) ▾
1 dnxw-xr-x 27 honoriflca honoriflca 4096 8月 17 20:09 .			1 dnxw-xr-x 27 honoriflca honoriflca 4096 8月 17 20:09 .		
2 dnxw-xr-x 4 honoriflca honoriflca 4096 5月 29 21:27 alg			2 dnxw-xr-x 4 honoriflca honoriflca 4096 5月 29 21:27 alg		
3 dnxw----- 20 honoriflca honoriflca 4096 8月 9 13:24 .cache			3 dnxw----- 20 honoriflca honoriflca 4096 8月 9 13:24 .cache		
4 dnxw----- 16 honoriflca honoriflca 4096 4月 24 14:43 .config			4 dnxw----- 16 honoriflca honoriflca 4096 4月 24 14:43 .config		
5 dnxw----- 3 honoriflca honoriflca 4096 4月 17 13:52 .emacs.d			5 dnxw----- 3 honoriflca honoriflca 4096 4月 17 13:52 .emacs.d		
6 dnxw-xr-x 22 honoriflca honoriflca 4096 4月 26 00:45 gdb-10.1			6 dnxw-xr-x 22 honoriflca honoriflca 4096 4月 26 00:45 gdb-10.1		
7 dnxw----- 3 honoriflca honoriflca 4096 3月 21 21:27 .gnupg			7 dnxw----- 3 honoriflca honoriflca 4096 3月 21 21:27 .gnupg		
8 dnxw----- 3 honoriflca honoriflca 4096 3月 21 21:04 .local			8 dnxw----- 3 honoriflca honoriflca 4096 3月 21 21:04 .local		
9 dnxw----- 5 honoriflca honoriflca 4096 3月 21 21:04 .mozilla			9 dnxw----- 5 honoriflca honoriflca 4096 3月 21 21:04 .mozilla		
10 dnxw-xr-x 3 honoriflca honoriflca 4096 4月 17 23:21 OS_Labwork			10 dnxw-xr-x 3 honoriflca honoriflca 4096 4月 17 23:21 OS_Labwork		
11 dnxw-xr-x 4 honoriflca honoriflca 4096 8月 17 20:09 OS_LessonDesign			11 dnxw-xr-x 4 honoriflca honoriflca 4096 8月 17 20:09 OS_LessonDesign		
12 dnxw----- 3 honoriflca honoriflca 4096 3月 22 09:22 .pkt			12 dnxw----- 3 honoriflca honoriflca 4096 3月 22 09:22 .pkt		
13 dnxw-xr-x 3 honoriflca honoriflca 4096 3月 22 09:22 snap			13 dnxw-xr-x 3 honoriflca honoriflca 4096 3月 22 09:22 snap		
14 dnxw----- 2 honoriflca honoriflca 4096 3月 21 21:27 .ssh			14 dnxw----- 2 honoriflca honoriflca 4096 3月 21 21:27 .ssh		
15 dnxw----- 6 honoriflca honoriflca 4096 4月 11 14:17 .thunderbird			15 dnxw----- 6 honoriflca honoriflca 4096 4月 11 14:17 .thunderbird		
16 dnxw-xr-x 2 honoriflca honoriflca 4096 4月 18 23:30 .vln			16 dnxw-xr-x 2 honoriflca honoriflca 4096 4月 18 23:30 .vln		
17 dnxwnox-x 3 honoriflca honoriflca 4096 3月 22 09:22 .vscode			17 dnxwnox-x 3 honoriflca honoriflca 4096 3月 22 09:22 .vscode		
18 dnxw-xr-x 3 honoriflca honoriflca 4096 5月 26 22:22 xv6-public			18 dnxw-xr-x 3 honoriflca honoriflca 4096 5月 26 22:22 xv6-public		
19 dnxw-xr-x 2 honoriflca honoriflca 4096 3月 21 21:04 公共的			19 dnxw-xr-x 2 honoriflca honoriflca 4096 3月 21 21:04 公共的		
20 dnxw-xr-x 2 honoriflca honoriflca 4096 3月 21 21:04 模板			20 dnxw-xr-x 2 honoriflca honoriflca 4096 3月 21 21:04 模板		
21 dnxw-xr-x 2 honoriflca honoriflca 4096 3月 21 21:04 视频			21 dnxw-xr-x 2 honoriflca honoriflca 4096 3月 21 21:04 视频		
22 dnxw-xr-x 2 honoriflca honoriflca 4096 3月 21 21:04 图片			22 dnxw-xr-x 2 honoriflca honoriflca 4096 3月 21 21:04 图片		
23 dnxw-xr-x 2 honoriflca honoriflca 4096 3月 21 21:04 文档			23 dnxw-xr-x 2 honoriflca honoriflca 4096 3月 21 21:04 文档		
24 dnxw-xr-x 2 honoriflca honoriflca 4096 3月 21 21:04 下载			24 dnxw-xr-x 2 honoriflca honoriflca 4096 3月 21 21:04 下载		
25 dnxw-xr-x 2 honoriflca honoriflca 4096 3月 21 21:04 音乐			25 dnxw-xr-x 2 honoriflca honoriflca 4096 3月 21 21:04 音乐		
26 dnxw----- 3 honoriflca honoriflca 4096 8月 6 14:09 桌面			26 dnxw----- 3 honoriflca honoriflca 4096 8月 6 14:09 桌面		

4. 一些遐想

4.1 执行函数 `execv` 和 `popen` 的选择

一方面在执行文件选择上:

`execv` 是调用了通过自己传入的可执行文件作为执行参照，将所有的参数分割完成之后传入函数当中进行执行。其执行文件是需要自己去定义适配的

但是 `popen` 不同，他需要将指令还原为完整的一个指令，其内部自动调用 `pipe`, `fork`

和 `sh` 进行执行。具体的执行过程和切分其实不归我们管。

这就使得其在执行 `shell` 文本的时候存在差异。

要是对于 `execv`，要么直接进行命令首单词的后三位比对，若匹配 `sh` 则直接添加 `/bin/bash` 进行执行，要么就得自己将指令输入为 `bash ***.h` 执行。但是 `popen` 就很方便了。可以直接传入，其自行调用

另一方面，`execv` 对于执行需要直接输出到控制台的指令十分方便，`popen` 相当于把管道的整个过程给自己封装了进去，使用的时候只拿结果就行。相对于自己去执行 `pipe()`，然后再 `fork()` 的过程较为讨巧，但是其结果正确路却会很高。

4.2 管道嵌套的思考

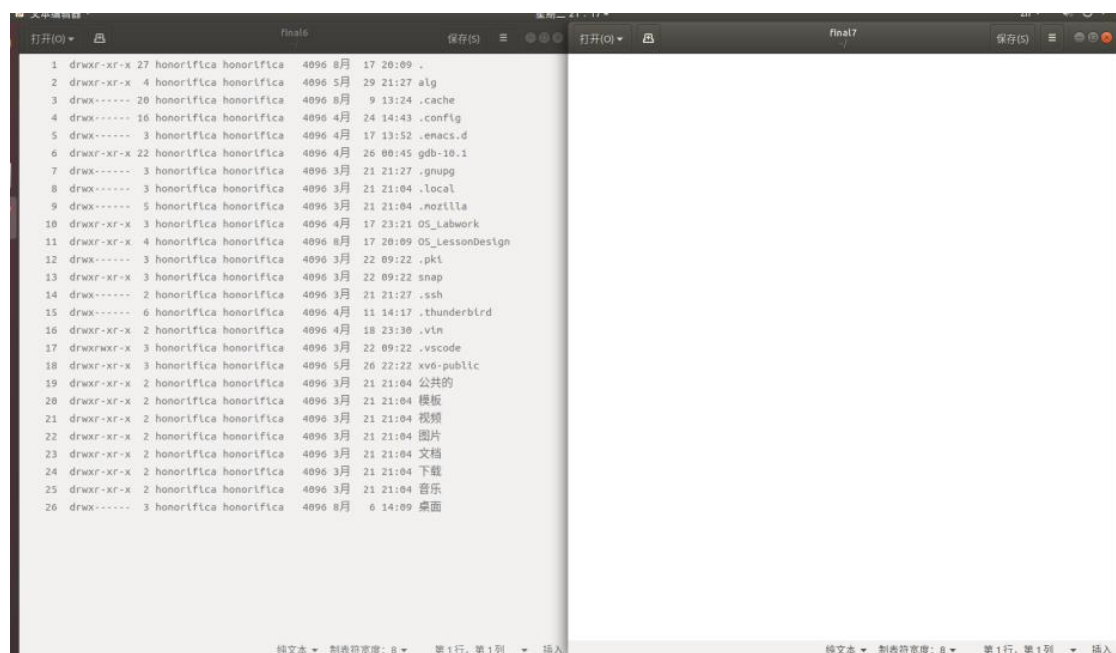
在实现管道嵌套的时候采用的思路是用 `popen` 将结果存出来之后，将其结果进行转存，并在下一条执行指令的恰地方添加其转存路径作为参数进行调用。这和实际的管道执行虽然结果相同但是过程却有所偏差。

实际的管道是将标准输入输出流的结果转给后续的指令当作执行参数。但是由于理解的偏差而采用中转文件的方法会使得不管有没有标准输入输出，结果全部都会被当作执行参数进行传递。例如下面的例子：

还是使用之前测试的指令 `cat final1 | grep honorifica | grep 4096 | cat -n > final3`

将其改为：`cat final1 | more | more | grep honorifica | cat | cat | grep 4096 | cat -n > final6 | cat -n | grep 10 > final7`

这个指令正常而言对于自带终端会在结果存入了 `final6` 之后，`final7` 之中是没有内容的



```
1 drwxr-xr-x 27 honorifica honorifica 4096 8月 17 20:09 .
2 drwxr-xr-x 4 honorifica honorifica 4096 5月 29 21:27 alg
3 drwx----- 20 honorifica honorifica 4096 8月 9 13:24 .cache
4 drwx----- 16 honorifica honorifica 4096 4月 24 14:43 .config
5 drwx----- 3 honorifica honorifica 4096 4月 17 13:52 .emacs.d
6 drwxr-xr-x 22 honorifica honorifica 4096 4月 26 00:45 gdb-10.1
7 drwx----- 3 honorifica honorifica 4096 3月 21 21:27 .gnupg
8 drwx----- 3 honorifica honorifica 4096 3月 21 21:04 .local
9 drwx----- 5 honorifica honorifica 4096 3月 21 21:04 .mozilla
10 drwxr-xr-x 3 honorifica honorifica 4096 4月 17 23:21 OS_labwork
11 drwxr-xr-x 4 honorifica honorifica 4096 8月 17 20:09 OS_LessonDesign
12 drwx----- 3 honorifica honorifica 4096 3月 22 09:22 .pkt
13 drwxr-xr-x 3 honorifica honorifica 4096 3月 22 09:22 snap
14 drwx----- 2 honorifica honorifica 4096 3月 21 21:27 .ssh
15 drwx----- 6 honorifica honorifica 4096 4月 11 14:17 .thunderbird
16 drwxr-xr-x 2 honorifica honorifica 4096 4月 18 23:30 .vim
17 drwxr-xr-x 3 honorifica honorifica 4096 3月 22 09:22 .vscode
18 drwxr-xr-x 3 honorifica honorifica 4096 5月 26 22:22 xv6-public
19 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 公共的
20 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 模板
21 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 视频
22 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 图片
23 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 文档
24 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 下载
25 drwxr-xr-x 2 honorifica honorifica 4096 3月 21 21:04 音乐
26 drwx----- 3 honorifica honorifica 4096 8月 6 14:09 桌面
```

但是对于这个自己编写的 `shell`，最后一个指令还是将 `final6` 的内容当作了输入进行了执行。

这算是程序编写和现实的出入

4.3 指令执行过程中关于 malloc 的小缺陷

在指令执行的过程中，由于对于新空间的申请采用的都是 malloc，一开始没有注意到内容的回收。基于 malloc 的空间不会自己释放而是需要自己进行 free。

这一点的忽略使得在指令执行编码正确的基础上，会出现 malloc 失败的情况，当注意到之后，执行的指令大多数情况下是没有问题。但是有的时候在使用久了之后会出现执行的指令和输入的指令参数不符的情况。这应该是自己在进行 free 和 memset 的时候出现了部分的冲突和管理的失误。这是值得改进的地方。

4.4 管道和重定向选用 ls -al 而非 ps aux 的思考

在重定向对 ps auxd 的内容进行分析的时候，会发现终端的执行情况和自己的 shell 不一致。经过对比发现其实自己写的终端也是个进程，再系统自带终端的 ps aux 和运行自己的 shell 时使用 ps aux 就会出现数量的不一致，虽然 ps aux 更多，更利于测试，但不利于

但是对于 ls -al，其内容敲定了基本上不会有所变化，除非有涉及到下属文件的操作，便于测试，便于比较纠错，便于反应 shell 的功能。而且其相对于 ls，具有更多的细节，可以供 grep 的内容较多，也较为利于测试

4.5 重定向指令对于 fflush()的使用

在编写管道指令的时候，发现有的时候结果执行归执行，但是没有被存到结果文件当中就被取出。进行测试之后发觉是没有彻底写入造成的。

这是由于调用 fpuc 的时候会先存进缓冲区，当快满的时候再将其一次性写入。而 fflush()可以保证无论缓冲了多少都会被写入。

这个函数在打断点测试的时候也很有用处。有时候将结果输出进行查看，但是经过了语句之后却没有反映。这时候使用 `fflush(stdout)`即可将存在 `stdout` 文件缓冲区的内容一次性输出而不用等待