

# Simple Shell

---

Li Chengyao

5140219288

May 28, 2017

## 1 PROBLEM

### 1.1 DESCRIPTION

Write a program that can imitate a simple shell and execute some basic commands. You can see the detail in the end of chapter 3 of *OPERATING SYSTEM CONCEPTS WITH JAVA(Seventh Edition)*, page 127.

## 2 ALGORITHM

### 2.1 SHELL INTERFACE

The key is to create an external process to let the Linux operating system to help execute most of the command based on its command document in the Linux. So this program can only be executed in the OS like Linux other than Windows because the command in Linux is in Command documents. This kind of commands is like *ls*, *cat*, *pwd*. What we need to do is to read in user's commands and transmit them to Linux Command Documents.

Some commands in the requirements like *cd*, *history*, *!* and *exit* are not in the system, and we have to implement them by ourselves.

### 2.2 IMPLEMENTATION OF *cd*

This command includes a few operations. For instance,

`cd ..` (1)

`cd /usr/bin` (2)

`cd src` (3)

Thus we need to get the current directory using *getProperty()* first. For (1) we can get the directory by wiping out the last file. But if the current directory has already been *root*, the directory

has to be set as "/". For (2) and (3), we need to judge whether the directory is legal or not by *isDirectory()*. Then we replace the directory according to the command for (2) or add the folder after current directory for (3).

### 2.3 IMPLEMENTATION OF *history*

This command is implemented with the help of *ArrayList*. Every time the user input something we can save it into an *ArrayList* as long as it's not the same as the last one. When the user input *history* the commands stored in the *ArrayList* will be printed with an order number.

If the user input "!", the last command will be executed automatically. In the meantime "!" with a command number *i* leads to the execution of the *i*th last command. In addition, if there is more characters after "!" they will be concatenated to the last command. For example if the last command is *ls* and the user input *!! -l*, the shell actually executes *ls -l*.

### 2.4 ROBUSTNESS CONSIDERATION

If the user input is illegal a *IOException* will be thrown. The shell interface will give out an error message and continue to work until an *exit* is input to terminate the shell.

## 3 RESULTS

### 3.1 ENVIRONMENT

- Ubuntu 14.04 LTS
- Java Development Kit 1.8.0\_131
- Eclipse

### 3.2 SCREENSHOTS OF THE RESULT

We use Linux Terminal to compile and execute the program. The result is shown in Fig. 3.1.

### 3.3 THOUGHTS

It is the first time I use an coding environment under Linux. The process of encountering the problems and addressing them is very challenging and helps me a lot.

```
darwin@darwin-INVALID:~/Desktop$ java SimpleShell
jsh>pwd
/home/darwin/Desktop
jsh>cd ..
jsh>pwd
/home/darwin
jsh>!!
pwd
/home/darwin
jsh>cd Desktop
jsh>pwd
/home/darwin/Desktop
jsh>history
0 pwd
1 cd ..
2 pwd
3 cd Desktop
4 pwd
5 history
jsh>!2
pwd
/home/darwin/Desktop
jsh>exit
darwin@darwin-INVALID:~/Desktop$ |
```

Figure 3.1: Screenshot of Simple Shell Interface