

# Shell 和 Vim 实例报告

姓名：孔维凯      学号：23020007051

2024 年 9 月 12 日

目录

1	Shell	5
1.1	实例 1	5
1.1.1	练习内容	5
1.1.2	结果	5
1.1.3	解题感悟	5
1.2	实例 2	5
1.2.1	练习内容	5
1.2.2	结果	5
1.2.3	解题感悟	5
1.3	实例 3	6
1.3.1	练习内容	6
1.3.2	结果	6
1.3.3	解题感悟	6
1.4	实例 4	6
1.4.1	练习内容	6
1.4.2	结果	6
1.4.3	解题感悟	6
1.5	实例 5	6
1.5.1	练习内容	6
1.5.2	结果	7
1.5.3	解题感悟	7
1.6	实例 6	7
1.6.1	练习内容	7
1.6.2	结果	7
1.6.3	解题感悟	7
1.7	实例 7	7
1.7.1	练习内容	7
1.7.2	结果	7
1.7.3	解题感悟	8
1.8	实例 8	8
1.8.1	练习内容	8
1.8.2	结果	8
1.8.3	解题感悟	8
1.9	实例 9	8

1.9.1	练习内容	8
1.9.2	结果	9
1.9.3	解题感悟	9
1.10	实例 10	9
1.10.1	练习内容	9
1.10.2	结果	9
1.10.3	解题感悟	9
<b>2</b>	<b>Vim</b>	<b>9</b>
2.1	实例 1	9
2.1.1	练习内容	9
2.1.2	结果	9
2.1.3	解题感悟	9
2.2	实例 2	9
2.2.1	练习内容	9
2.2.2	结果	10
2.2.3	解题感悟	10
2.3	实例 3	10
2.3.1	练习内容	10
2.3.2	结果	10
2.3.3	解题感悟	10
2.4	实例 4	11
2.4.1	练习内容	11
2.4.2	结果	11
2.4.3	解题感悟	11
2.5	实例 5	11
2.5.1	练习内容	11
2.5.2	结果	11
2.5.3	解题感悟	11
2.6	实例 6	11
2.6.1	练习内容	11
2.6.2	结果	11
2.6.3	解题感悟	11
2.7	实例 7	12
2.7.1	练习内容	12
2.7.2	结果	12

2.7.3	解题感悟	12
2.8	实例 8	12
2.8.1	练习内容	12
2.8.2	结果	12
2.8.3	解题感悟	12
2.9	实例 9	12
2.9.1	练习内容	12
2.9.2	结果	12
2.9.3	解题感悟	12
2.10	实例 10	13
2.10.1	练习内容	13
2.10.2	结果	13
2.10.3	解题感悟	13
3	github 链接	13

# 1 Shell

## 1.1 实例 1

### 1.1.1 练习内容

用./+ 文件名编译 Shell 脚本

### 1.1.2 结果

如果直接使用./+ 文件名，系统会显示权限不足，这时要使用 `chmod +x 文件夹名/文件名` 指令来增加权限，再使用./+ 文件名的指令就可以编译 Shell 脚本了。

### 1.1.3 解题感悟

使用这个指令来编译 Shell 脚本非常方便快捷。

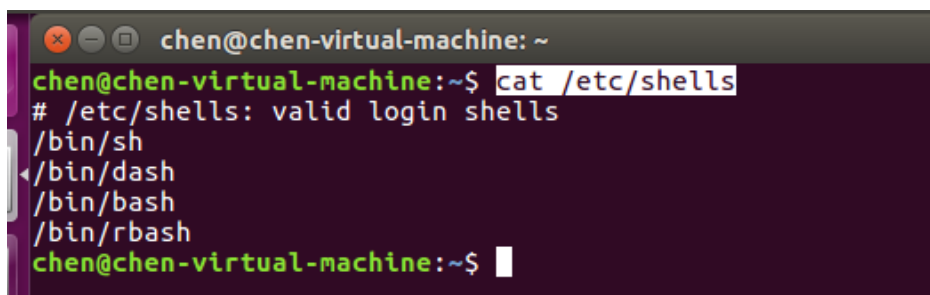
## 1.2 实例 2

### 1.2.1 练习内容

查看 Shell 编译器环境有哪些

### 1.2.2 结果

使用 `cat /etc/shells` 指令即可查看所有 shell 环境



```
chen@chen-virtual-machine: ~  
chen@chen-virtual-machine:~$ cat /etc/shells  
# /etc/shells: valid login shells  
/bin/sh  
/bin/dash  
/bin/bash  
/bin/rbash  
chen@chen-virtual-machine:~$
```

图 1: 例图 1

### 1.2.3 解题感悟

使用此指令有助于用户查看可以使用的 Shell 编辑环境。

1.3 实例 3

1.3.1 练习内容

查看当前有哪些文件

1.3.2 结果

使用 `ls` 指令查询当前文件

1.3.3 解题感悟

`ls` 便于查询当前文件，并将文件名用不同颜色显示表明其性质。

1.4 实例 4

1.4.1 练习内容

想要看一下从前的使用记录

1.4.2 结果

使用 `ls -l /bin/ | grep sh` 指令可以看到带有 `sh` 的所有使用记录

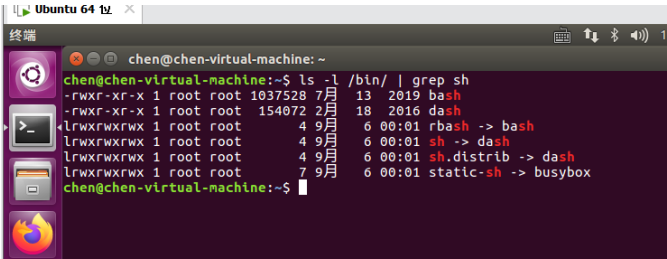


图 2: 例图 2

1.4.3 解题感悟

使用此指令方便快捷查看历史记录。

1.5 实例 5

1.5.1 练习内容

想要查看当前 `shell` 编辑环境

### 1.5.2 结果

使用 `echo $SHELL` 指令查看当前 shell 编辑环境。

### 1.5.3 解题感悟

使用该指令可以查看当前 shell 编辑环境便于 shell 脚本的编写。

## 1.6 实例 6

### 1.6.1 练习内容

查看过往记录中的代码

### 1.6.2 结果

使用 `less /bin/+ 文件名` 查看文件代码。

### 1.6.3 解题感悟

使用这个指令可以打开这个文件查看。

## 1.7 实例 7

### 1.7.1 练习内容

创建文件并写一个 shell 脚本

### 1.7.2 结果

首先使用 `mkdir+ 文件夹名` 创建一个文件夹，在使用 `cd+ 文件夹名` 转移到该文件夹，使用 `touch+ 文件名` 创建一个 shell 脚本文件，然后使用 `vim+ 文件名` 代开该脚本进行编写 编写完成后，使用：`wq` 指令保存并

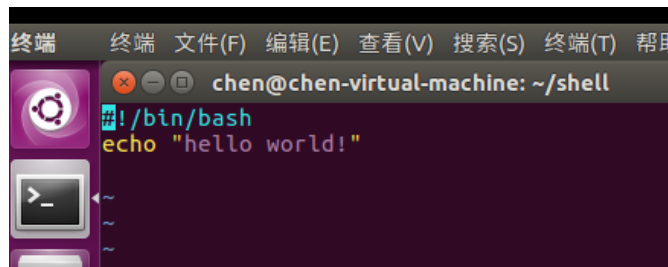
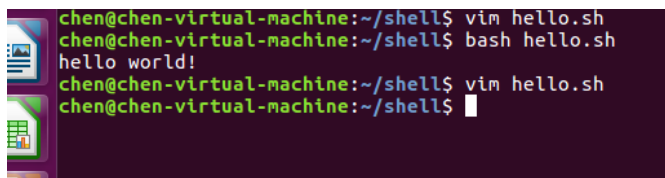


图 3: 例图 4

退出，最后使用 `bash+ 文件名` 运行 shell 脚本。



```
chen@chen-virtual-machine:~/shell$ vim hello.sh
chen@chen-virtual-machine:~/shell$ bash hello.sh
hello world!
chen@chen-virtual-machine:~/shell$ vim hello.sh
chen@chen-virtual-machine:~/shell$
```

图 4: 例图 5

### 1.7.3 解题感悟

使用这些指令可以创建文件夹并书写 shell 脚本。

## 1.8 实例 8

### 1.8.1 练习内容

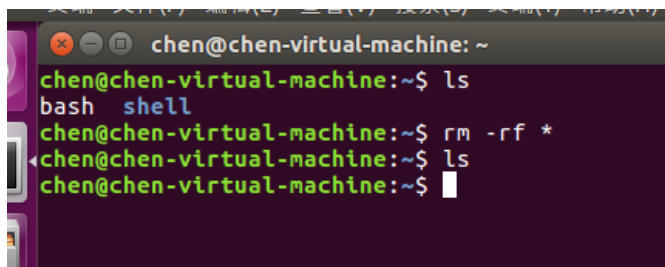
删除所有文件

### 1.8.2 结果

使用 `rm -rf *` 指令删除所有文件

### 1.8.3 解题感悟

此指令便捷删除所有文件，使页面焕然一新。



```
chen@chen-virtual-machine: ~
chen@chen-virtual-machine:~$ ls
bash shell
chen@chen-virtual-machine:~$ rm -rf *
chen@chen-virtual-machine:~$ ls
chen@chen-virtual-machine:~$
```

图 5: 例图 3

## 1.9 实例 9

### 1.9.1 练习内容

将文件移动到一个文件夹中



### 1.9.2 结果

使用 `mv+ 文件名 + 目的地文件夹名`即可将文件转移到文件夹中。

### 1.9.3 解题感悟

使用这个指令可以将文件转移到文件夹中使文件进行分区便于以后查找。

## 1.10 实例 10

### 1.10.1 练习内容

如何直接查看 shell 脚本的代码

### 1.10.2 结果

使用 `cat+ 文件地址`可以直接查看 shell 脚本的代码。

### 1.10.3 解题感悟

这个指令不仅可以查看 shell 脚本代码甚至可以查看所有类型的文件的代码。

## 2 Vim

### 2.1 实例 1

#### 2.1.1 练习内容

vim 是用来编辑代码的编辑器，如何下载 gcc 编辑器

#### 2.1.2 结果

首先使用 `sudo apt update` 指令，然后输入密码，再使用 `sudo apt install gcc` 指令来下载 gcc 编译器。

#### 2.1.3 解题感悟

使用这些指令就可以下载 gcc 编译器。

### 2.2 实例 2

#### 2.2.1 练习内容

如何编译代码

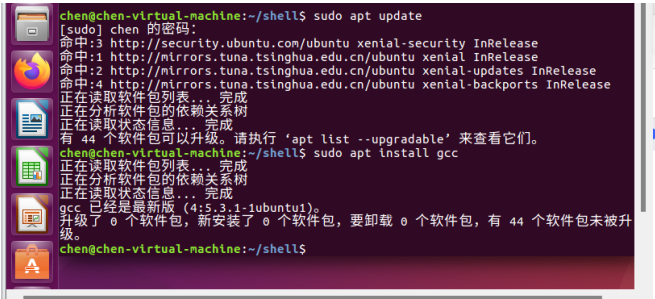


图 6: 例图 6

2.2.2 结果

使用 vim+ 文件名创建并打开一个文件书写代码，然后使用 gcc+ 文件名编译文件，最后使用 ./a.out 指令来运行文件。

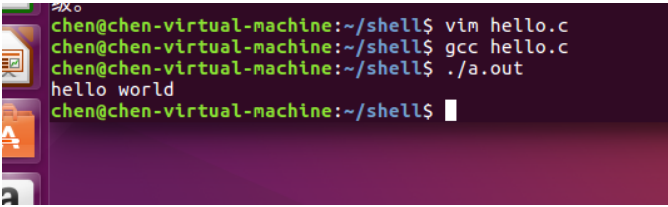


图 7: 例图 7

2.2.3 解题感悟

vim 可以用来编译所有文件。

2.3 实例 3

2.3.1 练习内容

vim 编辑器切换模式

2.3.2 结果

命令模式下输入：进入末行模式，编辑模式下可以输入文本，然后按 ESC 键退出编辑模式返回命令模式，末行模式输入：w 保存，: q 退出，: q! 强制退出不保存，: wq 保存并退出

2.3.3 解题感悟

这些指令能灵活切换模式并使用该模式下的操作。

## 2.4 实例 4

### 2.4.1 练习内容

使用光标移动操作

### 2.4.2 结果

使用 h、j、k、l 键控制光标左移、下移、上移、右移，使用 gg 移动到文件开头，G 移动到文件结尾，0 移动到行首，\$ 移动到行尾。

### 2.4.3 解题感悟

这些指令能使我们灵活使用光标移动。

## 2.5 实例 5

### 2.5.1 练习内容

使用查找功能

### 2.5.2 结果

使用 /pattern 指令从当前位置向前查找，使用 ?pattern 指令从当前位置向后查找。

### 2.5.3 解题感悟

这两个指令方便查找，尤其在大规模工程时节约了许多时间。

## 2.6 实例 6

### 2.6.1 练习内容

如何使用撤销操作

### 2.6.2 结果

输入 u 撤销上一次操作。

### 2.6.3 解题感悟

使用这个操作能便于使用者撤回操作。

## 2.7 实例 7

### 2.7.1 练习内容

如何便捷插入文本

### 2.7.2 结果

输入 `i` 可以在当前光标位置插入文本，输入 `a` 可以在光标后面插入文本，输入 `o` 可以在当前行的下方插入新的一行，输入 `O` 可以在当前行的上方插入新的一行。

### 2.7.3 解题感悟

这些操作方便了用户，用户不必一个个移动，极大提高了工作效率。

## 2.8 实例 8

### 2.8.1 练习内容

如何显示当前行号

### 2.8.2 结果

首先按 `ESC` 键返回到命令模式，然后输入 `:set nu` 指令就可以显示行号了。

### 2.8.3 解题感悟

这一操作是为删除一行代码服务的。

## 2.9 实例 9

### 2.9.1 练习内容

如何删除字符

### 2.9.2 结果

将光标移到要删除的字符上按下 `x` 键就可以删除该字符了。

### 2.9.3 解题感悟

这个操作方便用户快速删除字符。

## 2.10 实例 10

### 2.10.1 练习内容

如何匹配单字符

### 2.10.2 结果

使用 `.` 字符，比如 `a.b` 可以匹配到任何前面字符是 `a`，后面字符是 `b` 的语句。

### 2.10.3 解题感悟

这个操作便于用户快速查找语句。

## 3 github 链接