

Shell 与 Vim 实验报告

王杰 (23160001074)

2025 年 9 月 16 日

1 实验目的

- 掌握常用 Shell 基本命令、重定向与管道。
- 掌握 Vim 常见编辑操作、查找替换与窗口管理。
- 提升命令行环境下的文件管理与文本编辑效率。

2 实验环境

- 操作系统: Windows 11 (WSL2: Ubuntu 24.04)
- Shell: bash
- 编辑器: Vim 9.x
- L^AT_EX 编译器: XeLaTeX
- 在线编辑: Overleaf

3 练习内容

3.1 Shell (10 个)

1. 查看当前路径与用户名

```
pwd  
whoami
```

```
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools$ find . -type f -name "*.txt"  
.a.txt  
.hello.txt
```

2. 列出目录内容（普通/详细/含隐藏）

```
ls  
ls -lah
```

```
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools$ ls  
a.txt  dark.jpg  test_Python1.py  
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools$ ls -lah  
总计 36K  
drwxrwxr-x 2 tingol tingol 4.0K 9月 12 10:26 .  
drwxrwxr-x 3 tingol tingol 4.0K 9月  5 09:45 ..  
-rw-rw-r-- 1 tingol tingol   6 9月  5 10:23 a.txt  
-rw-rw-r-- 1 tingol tingol  17K 9月 12 10:25 dark.jpg  
-rw-rw-r-- 1 tingol tingol  201 9月 12 10:26 test_Python1.py
```

3. 创建、进入与删除目录

```
mkdir lab01 && cd lab01  
cd ..  
rmdir lab01
```

```
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools$ mkdir lab01 && cd lab01  
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools/lab01$ cd ..  
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools$ rmdir lab01/
```

4. 新建文件与查看内容

```
echo "Hello shell" > hello.txt  
cat hello.txt
```

```
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools$ echo "hello shell" > hello.txt  
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools$ cat hello.txt  
hello shell
```

5. 追加内容与行数统计

```
echo "line2" >> hello.txt  
wc -l hello.txt
```

```
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools$ echo "line2" >> hello.txt  
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools$ wc -l hello.txt  
2 hello.txt
```

6. 管道与过滤

```
ls /bin | head -n 5  
ls -l | grep '^d' # 仅显示目录
```

```
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools$ ls /bin | head -n 5
[
aa-enabled
aa-exec
aa-features-abi
aconnect
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools$ ls -l | grep '^d'
drwxrwxr-x 2 tingol tingol 4096 9月 16 17:26 lab
```

7. 查找文件

```
find . -type f -name "*.txt"
```

```
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools$ find . -type f -name "*.txt"
./a.txt
./hello.txt
```

8. 搜索文本

```
grep -n "line" hello.txt
```

```
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools$ grep -n "line" hello.txt
2:line2
```

9. 权限与执行脚本

```
printf '#!/usr/bin/env bash\necho Hi\n' > run.sh
chmod +x run.sh
./run.sh
```

```
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools$ printf '#!/usr/bin/env bash\necho Hi\n' > run.sh
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools$ chmod +x run.sh
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools$ ./run.sh
Hi
```

10. 打包与查看压缩包内容

```
tar -czf hosts.tgz /etc/hosts
tar -tzf hosts.tgz | head
```

```
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools$ tar -czf hosts.tgz /etc/hosts
tar: 从成员名中删除开头的"/"
tingol@tingol-VMware-Virtual-Platform:~/桌面/lab/2025q2/SystemTools$ tar -tzf hosts.tgz | head
etc/hosts
```

3.2 Vim (10 个)

1. 启动、保存与退出

```
vim notes.txt  
# 在普通模式下输入:  
# :w 保存  
# :q 退出  
# :wq 保存并退出  
# :q! 强制不保存退出
```

2. 进入插入模式与返回普通模式

```
# 普通模式下:  
i # 在光标前插入  
a # 在光标后插入  
o / O # 另起下一/上一行插入  
<Esc> # 返回普通模式
```

3. 光标移动（高效导航）

```
h j k l # 左下上右  
w / b # 下一个/上一个单词开头  
0 / $ # 行首 / 行尾  
gg / G # 文件开头 / 结尾
```

4. 查找与高亮

```
/pattern # 向下查找  
n / N # 下一个 / 上一个匹配  
:set hlsearch incsearch # 高亮与增量查找
```

5. 替换（当前行与全文）

```
:s/old/new/g # 当前行全部替换  
:%s/old/new/gc # 全文替换并逐个确认
```

6. 撤销、重做与重复

```
u # 撤销  
<C-r> # 重做 (Ctrl+r)  
. # 重复上一次操作
```

7. 复制/粘贴/删除（操作行与词）

```
yy      # 复制当前行  
p      # 在光标后粘贴  
dd      # 删除（剪切）当前行  
dw / yw # 删除/复制一个词  
x      # 删除一个字符
```

8. 可视模式与缩进

```
v / V / <C-v> # 字符/行/块可视  
> / <      # 右/左缩进已选文本  
=          # 自动对齐已选文本
```

9. 分屏与窗口管理

```
:split file2.txt    # 水平分屏  
:vsplit file3.txt  # 垂直分屏  
<C-w> h/j/k/l    # 窗口间移动  
:close            # 关闭当前窗口
```

10. 多缓冲区（多文件）切换

```
:args a.txt b.txt c.txt # 打开多个文件  
:ls                  # 查看缓冲区  
:bn / :bp           # 下一个/上一个缓冲  
:bd                 # 关闭当前缓冲
```



4 常见问题与解决

- **命令找不到 (command not found)**: 确认是否安装并在 PATH 中; WSL 下可用 `sudo apt install <pkg>` 安装。
- **权限不足 (Permission denied)**: 脚本需加执行位 `chmod +x file.sh`, 或用 `sudo` 执行需要管理员权限的命令。
- **重定向覆盖文件**: 使用 `>` 会覆盖, 谨慎; 若需追加用 `>>`。
- **Vim 不能保存 (E212: Can't open file for writing)**: 无写权限目录, 使用 `:w !sudo tee %` 或在 Shell 中 `sudo vim file`。
- **Vim 查找不高亮或不增量**: 执行 `:set hlsearch incsearch`, 写入 `~/.vimrc` 持久化。
- **分屏后方向键移动到中文输入法候选框**: 保持普通模式下用 `<C-w> h/j/k/l` 移动窗口。

5 心得体会

Shell 的管道、重定向与文本处理工具能高效完成日常文件操作; Vim 的模式化编辑在熟悉常用按键后能显著提升速度。将二者结合, 在终端内即可完成“查找—编辑—验证”的快速闭环; 后续计划把常用设置写入 `~/.bashrc` 与 `~/.vimrc`, 进一步优化工作流。

6 github 地址

<https://github.com/yourname/shell-vim-lab.git>