

第二周实验报告

张津侨

2024 年 8 月 31 日

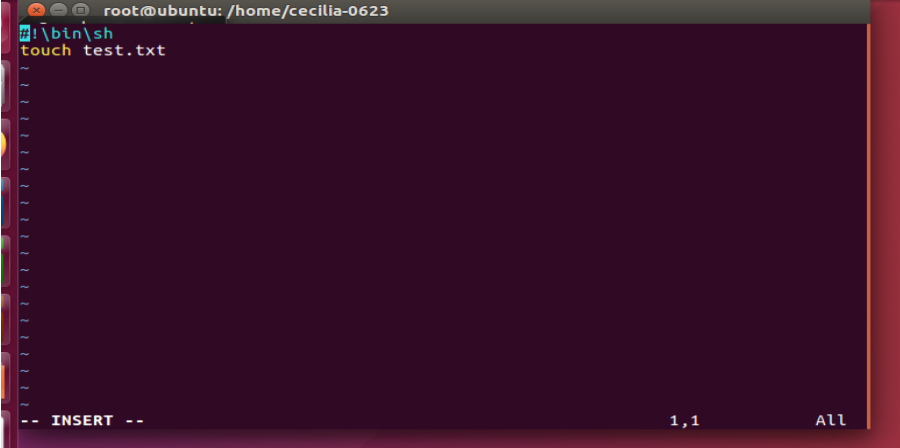
1 练习内容及结果

1.1 查看虚拟机中是否安装好了 vim, 没有则安装 vim(使用安装 vim 的指令还需安装 sudo)

```
cecilia-0623@ubuntu:~$ sudo vim min.sh
sudo: vim: command not found
cecilia-0623@ubuntu:~$ rpm -qa|grep vim
The program 'rpm' is currently not installed. You can install it by typing:
sudo apt install rpm
cecilia-0623@ubuntu:~$ yum -y install vim
The program 'yum' is currently not installed. You can install it by typing:
sudo apt install yum
cecilia-0623@ubuntu:~$ sudo apt install yum
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  debugedit librpm3 librpmbuild3 librpmio3 librpmio3 librpmio3 librpmio3 librpmio3
  python-libxml2 python-lzma python-pycurl python-rpm python-sqlite
  python-sqlitecachec python-urlgrabber rpm rpm-common rpm2cpio
Suggested packages:
```

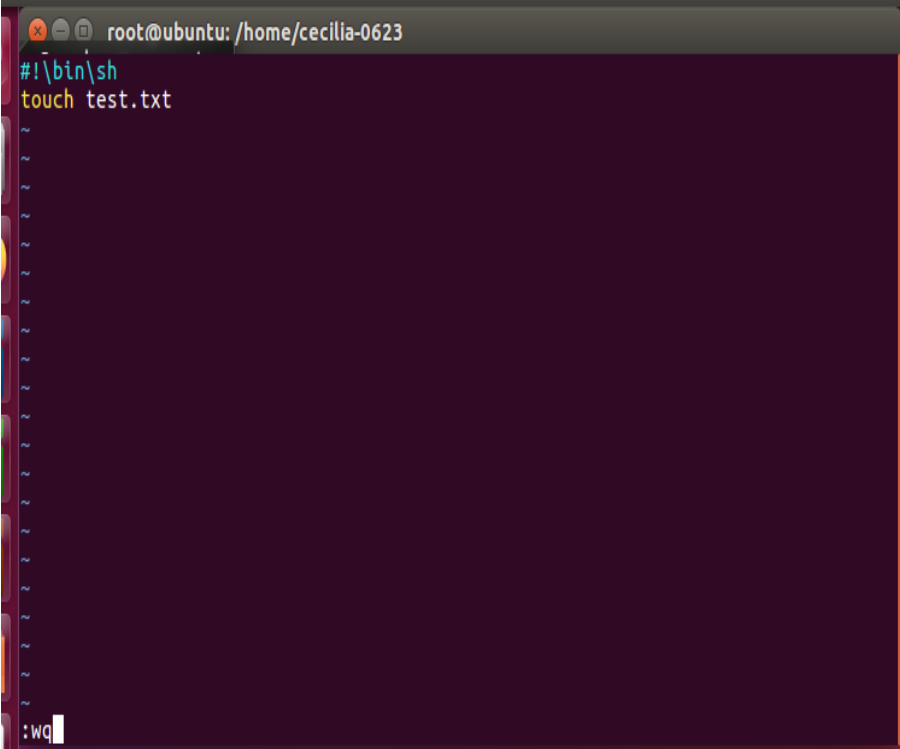
指令: yum -y install vim, sudo apt install yum

- 1.2 设置 shell 文件路径 (开头要有井号和感叹号), 创建一个用于创立文件的脚本, vim 中输入 i 进入编辑模式



```
root@ubuntu: /home/cecilia-0623
#!/bin/sh
touch test.txt
-- INSERT -- 1,1 All
```

- 1.3 vim 中输入: 进入命令模式, wq 表示保存并退出



```
root@ubuntu: /home/cecilia-0623
#!/bin/sh
touch test.txt
:wq
```

指令: wq

1.4 创建 shell 脚本并运行

```
o Mode
root@ubuntu:/home/cecilia-0623# sudo vim min.sh
root@ubuntu:/home/cecilia-0623# sudo touch min.sh
root@ubuntu:/home/cecilia-0623# ls
Desktop    Downloads      min.sh  Pictures  Templates
Documents  examples.desktop  Music   Public    Videos
root@ubuntu:/home/cecilia-0623# sudo vim min.sh
root@ubuntu:/home/cecilia-0623# vim min.sh
root@ubuntu:/home/cecilia-0623# sh min.sh
root@ubuntu:/home/cecilia-0623# ls
Desktop    Downloads      min.sh  Pictures  Templates  Videos
Documents  examples.desktop  Music   Public    test.txt
root@ubuntu:/home/cecilia-0623#
```

指令：vim min.sh, sh min.sh

1.5 查看文本目录

```
root@ubuntu:/home/cecilia-0623# ls
Desktop    Downloads      min.sh  Pictures  Templates  Videos
Documents  examples.desktop  Music   Public    test.txt
root@ubuntu:/home/cecilia-0623#
```

指令：ls

1.6 查看当前 shell 文件

```
root@ubuntu:/home/cecilia-0623# cat/etc/shells
bash: cat/etc/shells: No such file or directory
root@ubuntu:/home/cecilia-0623#
```

指令：cat/etc/shells

1.7 查看 HOME 路径, 查看 shell 脚本路径

```
root@ubuntu:/home/cecilia-0623# echo $HOME
/root
root@ubuntu:/home/cecilia-0623# echo $SHELL
/bin/bash
```

指令: `echo $HOME`, `echo $SHELL`

1.8 创建 shell 文件, 并给出访问权限, 然后运行

```
root@ubuntu:/home/cecilia-0623# vim hello.sh
root@ubuntu:/home/cecilia-0623# chmod a+x hello.sh
root@ubuntu:/home/cecilia-0623# ./hello.sh
Hello Shell
Thu Aug 29 19:02:14 PDT 2024
root
root@ubuntu:/home/cecilia-0623#
```

指令: `vim hello.sh`, `chmod a+x hello.sh`

1.9 打印 “Hello,Shell” 并输出时间和作者的脚本

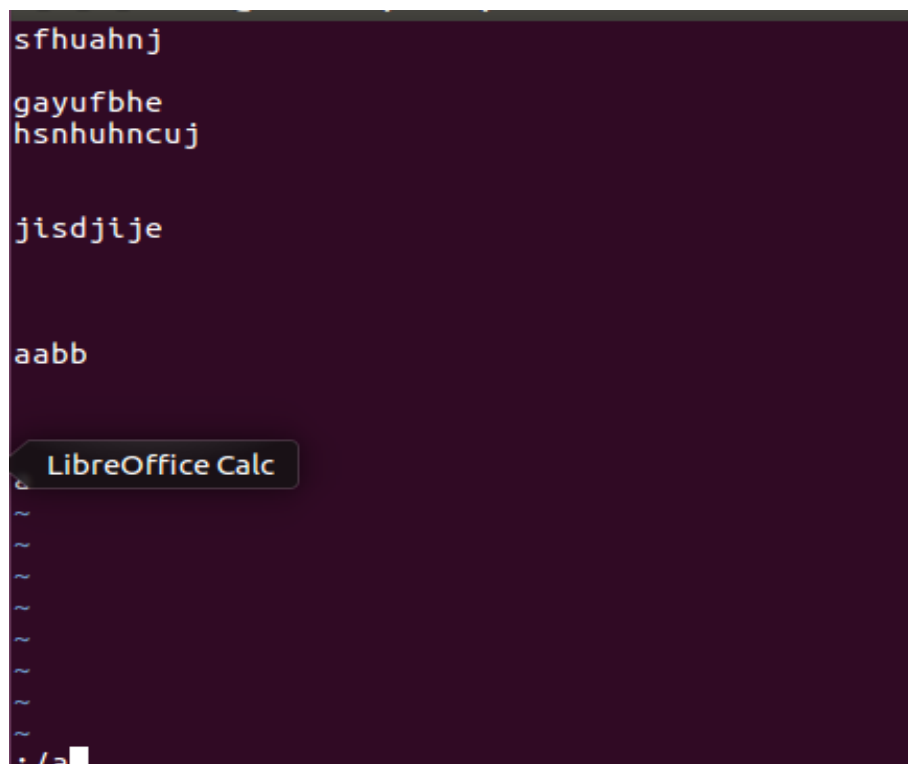
```
#!/bin/bash

echo "Hello Shell"
date
whoami

-- INSERT --
```

指令： echo"Hello,Shell",date,whoami

1.11 vim 中查找文本 (n 向下查找, N 向上查找)



指令: ./+ 查找的内容

1.12 课后习题一: vimtutor

```
root@ubuntu:/home/cecilia-0623# vimtutor

=====
=  Welcome to the VIM Tutor - Version 1.7  =
=====

Vim is a very powerful editor that has many commands, too many to
explain in a tutor such as this.  This tutor is designed to describe
enough of the commands that you will be able to easily use Vim as
an all-purpose editor.

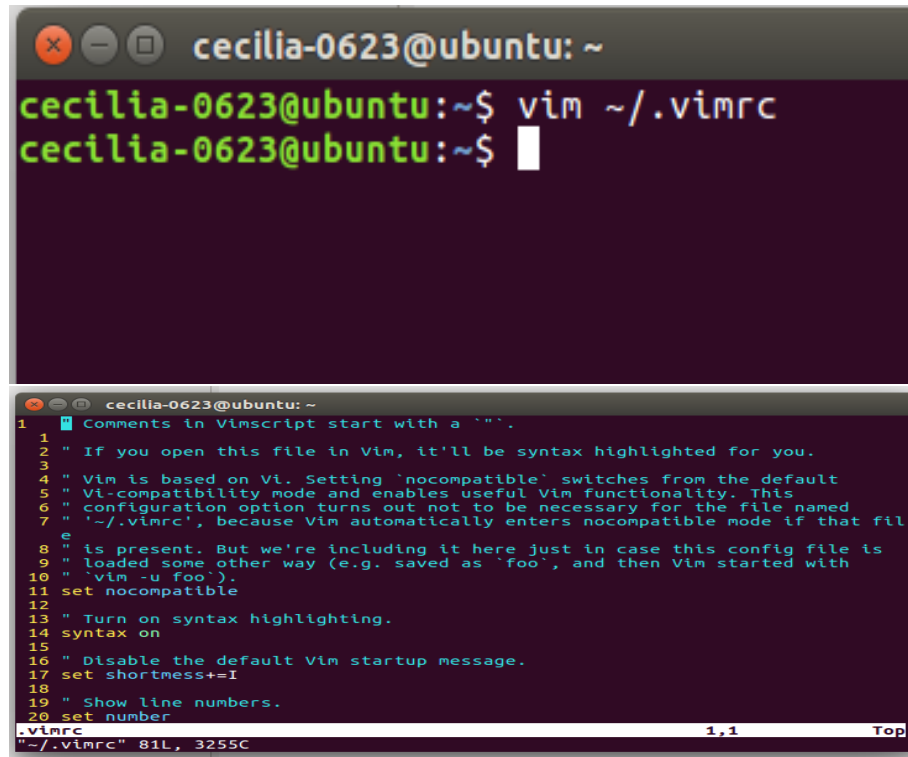
The approximate time required to complete the tutor is 25-30 minutes,
depending upon how much time is spent with experimentation.

ATTENTION:
The commands in the lessons will modify the text.  Make a copy of this
file to practise on (if you started "vimtutor" this is already a copy).

It is important to remember that this tutor is set up to teach by
use.  That means that you need to execute the commands to learn them
properly.  If you only read the text, you will forget the commands!

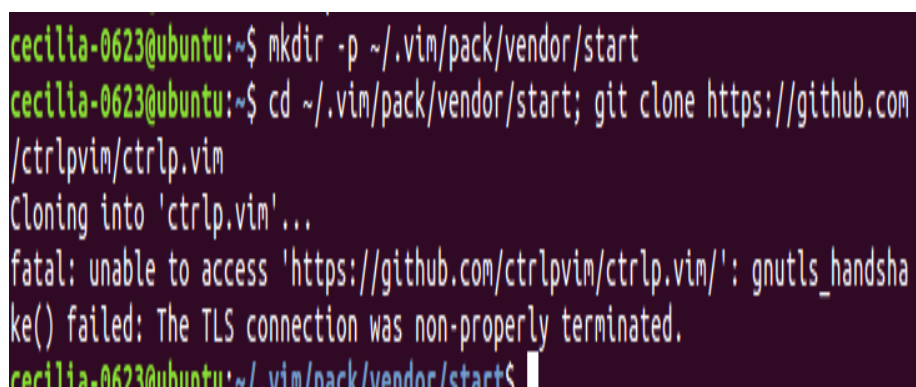
Now, make sure that your Shift-Lock key is NOT depressed and press
the j key enough times to move the cursor so that Lesson 1.1
completely fills the screen.
"/tmp/tutor0x1AHZ" 970 lines, 33248 characters
```


1.13 课后习题二：vim ~/.vimrc



```
cecilia-0623@ubuntu: ~  
cecilia-0623@ubuntu:~$ vim ~/.vimrc  
cecilia-0623@ubuntu:~$  
  
1 Comments in Vimscript start with a `"`.  
2 " If you open this file in Vim, it'll be syntax highlighted for you.  
3  
4 " Vim is based on Vi. Setting 'nocompatible' switches from the default  
5 " Vi-compatibility mode and enables useful Vim functionality. This  
6 " configuration option turns out not to be necessary for the file named  
7 " '~/.vimrc', because Vim automatically enters nocompatible mode if that fil  
8 " is present. But we're including it here just in case this config file is  
9 " loaded some other way (e.g. saved as 'foo', and then Vim started with  
10 " `vim -u foo`).  
11 set nocompatible  
12  
13 " Turn on syntax highlighting.  
14 syntax on  
15  
16 " Disable the default Vim startup message.  
17 set shortmess+=I  
18  
19 " Show line numbers.  
20 set number  
~/.vimrc 1,1 Top  
"~/.vimrc" 81L, 3255C
```

1.14 课后习题三：下载插件



```
cecilia-0623@ubuntu:~$ mkdir -p ~/.vim/pack/vendor/start  
cecilia-0623@ubuntu:~$ cd ~/.vim/pack/vendor/start; git clone https://github.com  
/ctrlpvim/ctrlp.vim  
Cloning into 'ctrlp.vim'...  
fatal: unable to access 'https://github.com/ctrlpvim/ctrlp.vim/': gnutls_handsha  
ke() failed: The TLS connection was non-properly terminated.  
cecilia-0623@ubuntu:~/.vim/pack/vendor/start$
```

1.15 拷贝/粘贴

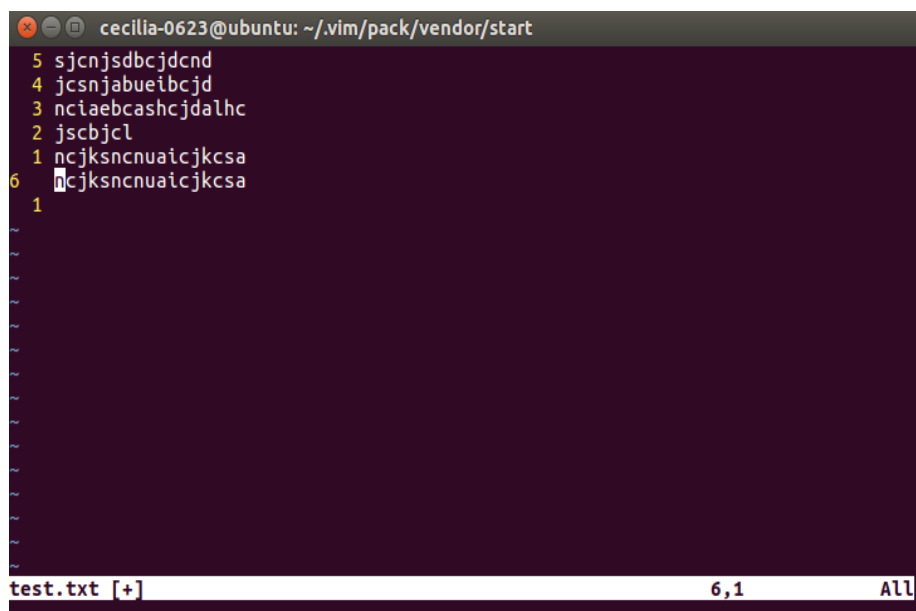
```

5 sjcnjsdbcjdcnd
4 jcsnjabueibcjd
3 nciaebcascjdahc
2 jscbjcl
1 ncjksncnuaicjkcsa
6 ncjksncnuaicjkcsa
1

```

p 表示在当前位置之后粘贴，P 表示在当前位置之前粘贴，yy 表示拷贝当前行于 ddP

1.16 简单的移动光标



The screenshot shows a terminal window with the title bar "cecilia-0623@ubuntu: ~/.vim/pack/vendor/start". The editor is displaying a file named "test.txt" with the following content:

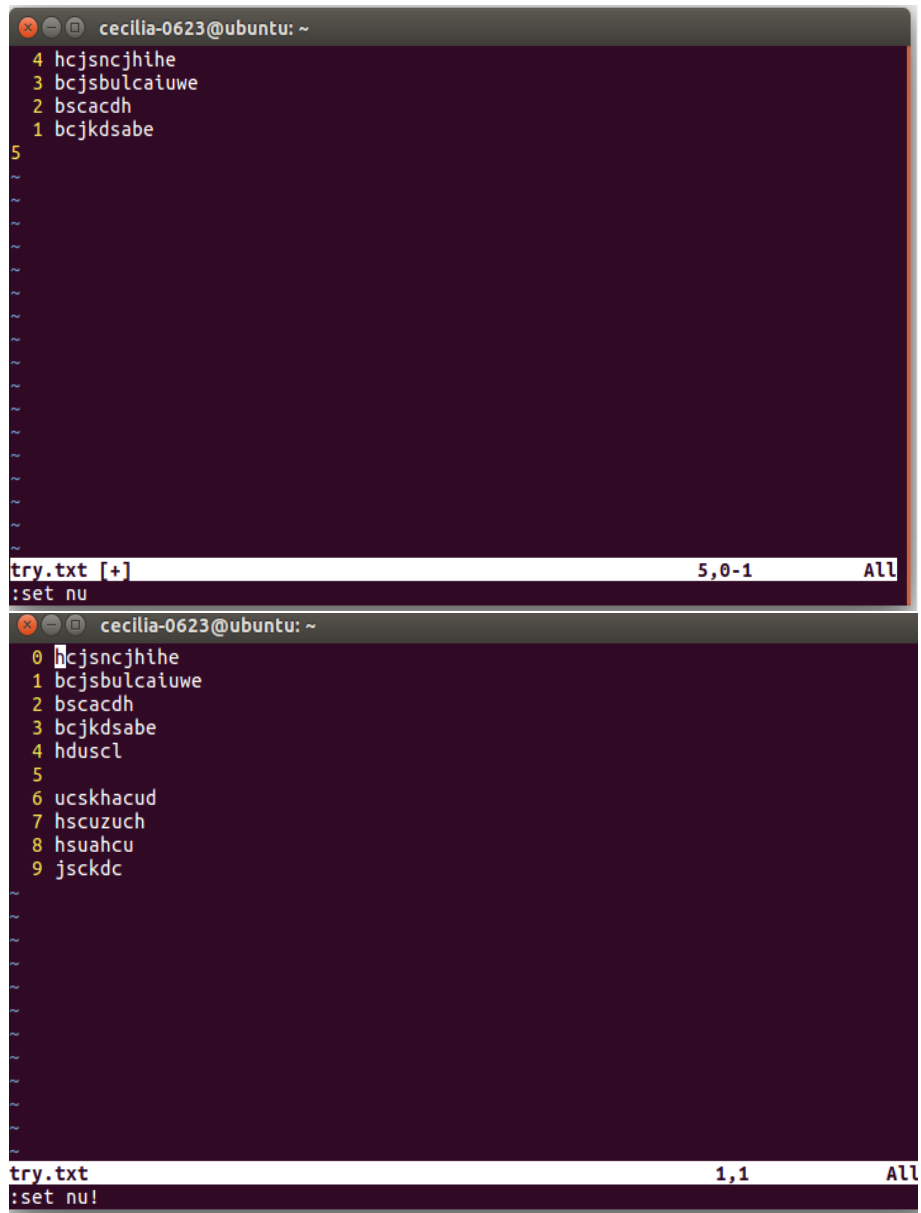
```
5 sjcnjsdbcjdcnd
4 jcsnjabueibcjd
3 nciaebcashcjdahc
2 jscbjcl
1 ncjksncnuaicjkcsa
6 ncjksncnuaicjkcsa
1
```

The cursor is positioned at the start of the 6th line. The status bar at the bottom indicates "test.txt [+]" on the left, "6,1" in the center, and "ALL" on the right.

0 → 数字零，到行头

/pattern → 搜索 pattern 的字符串（如果搜索出多个匹配，可按 n 键到下一个）

1.17 vim 显示行号/取消行号



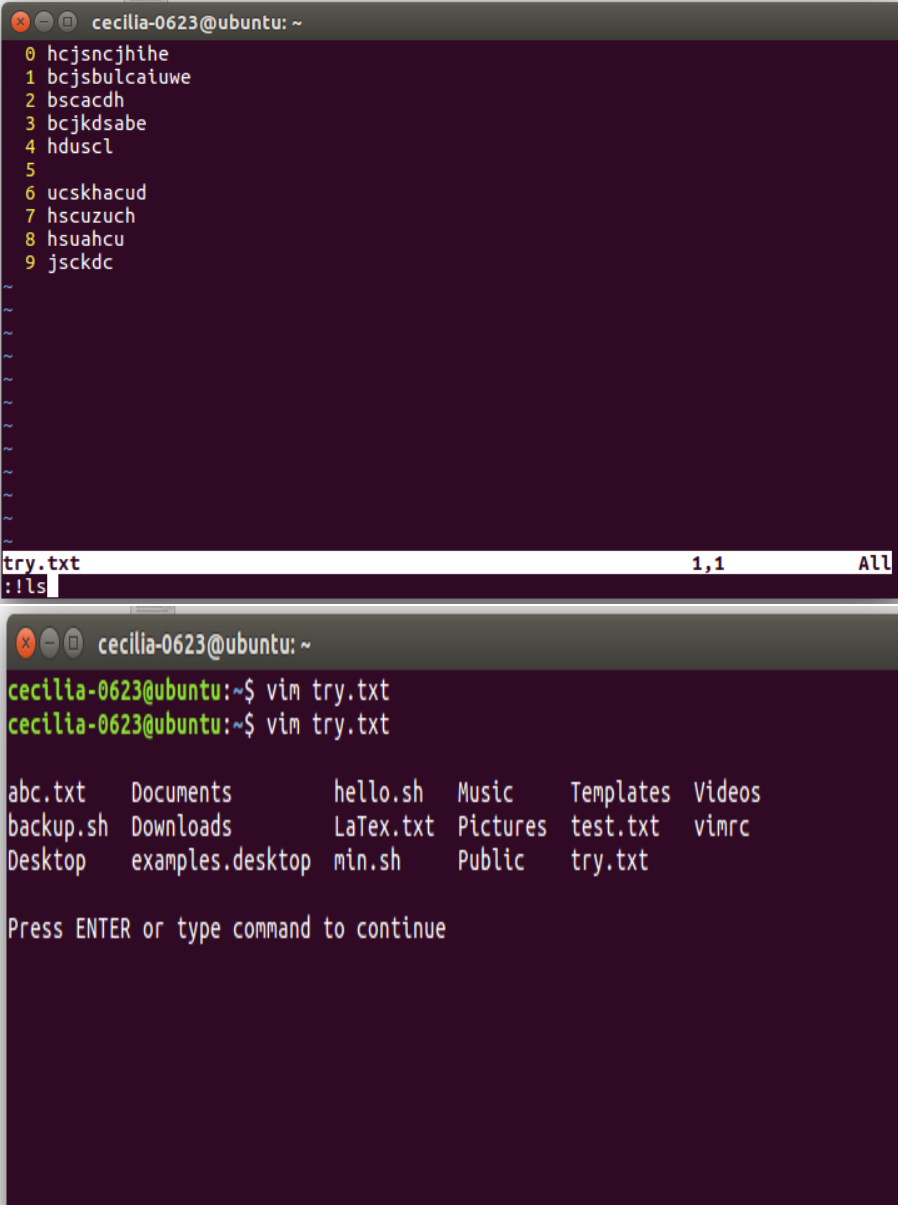
The image shows two screenshots of a vim editor window. The top screenshot shows a file named 'try.txt' with four lines of text. The line numbers 1, 2, 3, and 4 are displayed on the left side of the editor. The bottom screenshot shows the same file after the command ':set nu!' has been executed. The line numbers are now 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9, indicating that the line numbers have been reset to 0.

```
cecilia-0623@ubuntu: ~  
4 hcjsncjhihe  
3 bcjsbulcaiwe  
2 bscacdh  
1 bcjkdsabe  
5  
~  
~  
~  
~  
~  
try.txt [+]  
:set nu  
5,0-1 All
```

```
cecilia-0623@ubuntu: ~  
0 hcjsncjhihe  
1 bcjsbulcaiwe  
2 bscacdh  
3 bcjkdsabe  
4 hduscl  
5  
6 ucskhacud  
7 hscuzuch  
8 hsuahcu  
9 jsckdc  
~  
~  
~  
try.txt  
:set nu!  
1,1 All
```

set nu 表示显示行号，set nu! 表示取消行号

1.18 vim 中调用外部命令



```
cecilia-0623@ubuntu: ~  
0 hcjsncjhihe  
1 bcjsbulcatuwe  
2 bscacdh  
3 bcjksabe  
4 hduscl  
5  
6 ucskhacud  
7 hscuzuch  
8 hsuahcu  
9 jsckdc  
~  
~  
~  
~  
~  
~  
try.txt 1,1 All  
:!ls  
  
cecilia-0623@ubuntu: ~  
cecilia-0623@ubuntu:~$ vim try.txt  
cecilia-0623@ubuntu:~$ vim try.txt  
  
abc.txt  Documents  hello.sh  Music  Templates  Videos  
backup.sh Downloads  LaTeX.txt Pictures test.txt vimrc  
Desktop  examples.desktop min.sh  Public  try.txt  
  
Press ENTER or type command to continue
```

:! + 外部命令，不退出文件的同时执行外部命令，执行后按任意键可以切
换回文件

1.19 数据整理：统计 try.txt 文件中包含至少三个 a 且不以's 结尾的单词个数

```
cecilia-0623@ubuntu:~$ cat try.txt | tr "[:upper:]" "[:lower:]" | grep -E "^([a]*a){3}.*$" | grep -v "'s$" | wc -l
0
cecilia-0623@ubuntu:~$
```

```
cat try.txt | tr "[:upper:]" "[:lower:]" | grep -E "([a]*a)3.*"|grep -v "'s" | wc -l
```

1.20 数据整理：统计 try.txt 文件中共存在多少种词尾两字母组合

```
cecilia-0623@ubuntu:~$ cat try.txt | tr "[:upper:]" "[:lower:]" | grep -E "^([a]*a){3}.*$" | grep -v "'s$" | sed -E "s/.*([a-z]{2})$/\1/" | sort | uniq | wc -l
0
cecilia-0623@ubuntu:~$
```

```
cat try.txt | tr "[:upper:]" "[:lower:]" | grep -E "([a]*a)3.*"|grep -v "'s" | sed -E "s/.*([a-z]2)//" | sort | uniq | wc -l
```

2 解题感悟

夏季学期的第二周，我对于大致学习了 shell 脚本、vim 编辑器以及数据处理，对其进行了初步了学习。

Shell 脚本可以帮助我们简化重复的工作，省下很多时间和精力，是日后使用 Shell 时一大力工具。

说到 Vim，它的模式切换是一个关键点，普通模式、插入模式和命令模式的熟悉程度直接影响编辑效率。掌握一些常用的快捷键，可以大大提高我们的编辑速度。此外，安装一些插件能增强 Vim 的功能，让我们的使用体验更好。

数据整理使得我们对数据的整理与归纳更加方便，可以提高我们处理数据的效率

虽然对于这三个工具我只是进行了初步学习，但是我的视野得到了拓展，此后有对于相关工具的使用需要时也能更快地上手操作。

3 gitbit 账号链接:<https://github.com/Cecilia-hub>