

---

# 第二周实验报告

---

22090032004 高晓琳

2024 年 9 月 5 日

---

# 目录

<b>1</b>	<b>Shell 部分</b>	<b>1</b>
1.1	新建文件夹 . . . . .	1
1.2	查看程序使用手册 . . . . .	1
1.3	使用程序 . . . . .	2
1.4	写入文件 . . . . .	2
1.5	列出文件信息 . . . . .	2
1.6	\sys . . . . .	2
1.7	运算符 . . . . .	3
1.8	分支语句 . . . . .	3
1.9	循环语句 . . . . .	3
1.9.1	for 循环 . . . . .	3
1.9.2	while 循环 . . . . .	3
<b>2</b>	<b>vim 部分</b>	<b>4</b>
2.1	vimtutor . . . . .	4
2.2	基本操作模式 . . . . .	4
2.2.1	概述 . . . . .	4
2.2.2	转换方式 . . . . .	4
2.3	移动操作 . . . . .	5
2.4	搜索操作 . . . . .	5
2.5	编辑操作 . . . . .	6
2.5.1	保存与退出 . . . . .	6
2.5.2	删除文本 . . . . .	6
2.5.3	替换操作 . . . . .	6
2.5.4	重复与撤销操作 . . . . .	6
<b>3</b>	<b>数据整理</b>	<b>6</b>
3.0.1	cut 列提取命令 . . . . .	6
3.0.2	grep 命令 . . . . .	7
3.0.3	sed 命令 . . . . .	7
<b>4</b>	<b>总结</b>	<b>8</b>

## 1 Shell 部分

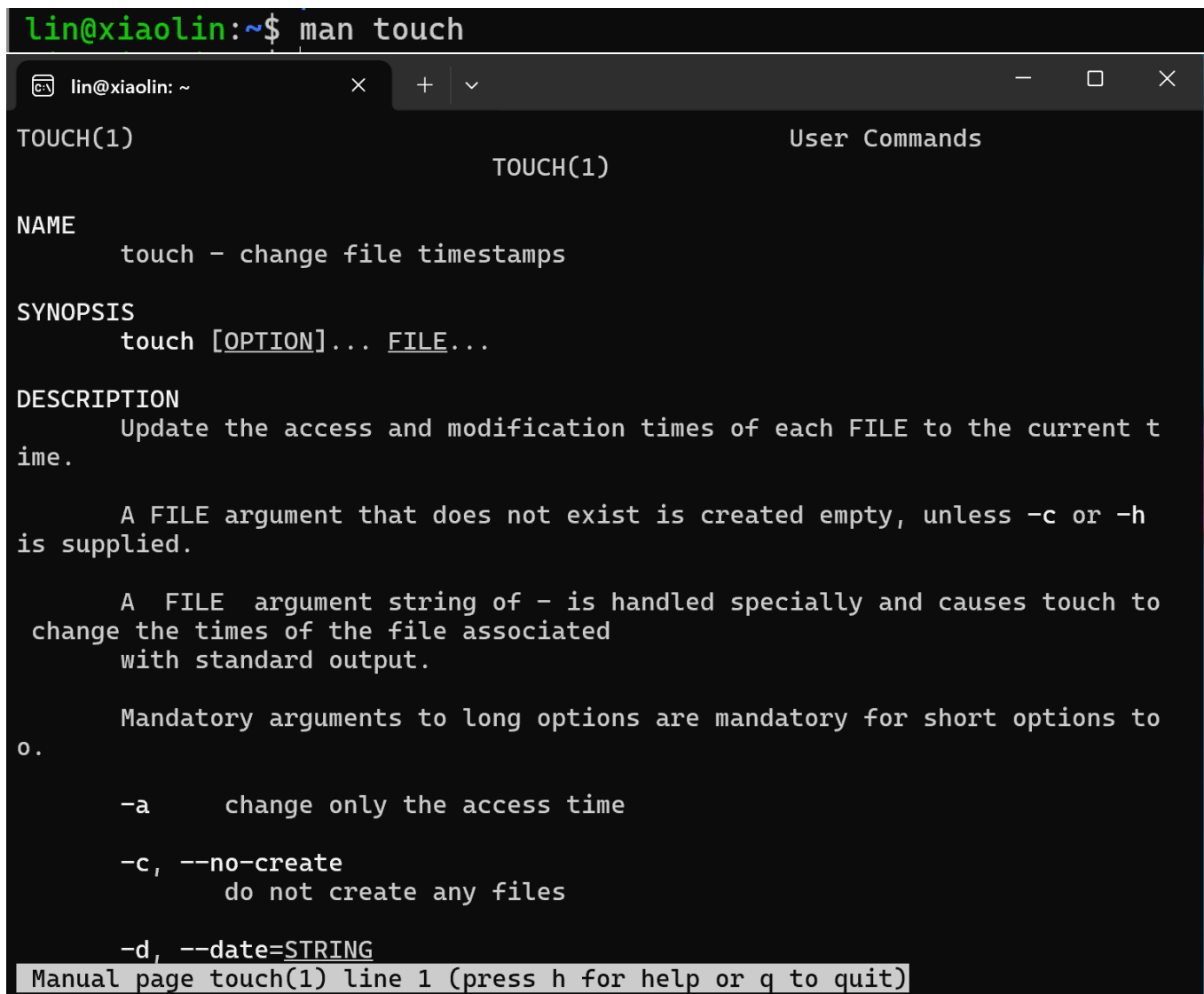
### 1.1 新建文件夹

```
lin@xiaolin:/tmp$ mkdir missing1
lin@xiaolin:/tmp$ cd
lin@xiaolin:~$ cd /tmp
lin@xiaolin:/tmp$ ls
missing          systemd-private-6d829499d79240df85a3b94b511d2d7c-systemd-lo
gind.service-PMgC0V
missing1         systemd-private-6d829499d79240df85a3b94b511d2d7c-systemd-re
solved.service-B7lZik
snap-private-tmp
```

图 1: 在/tmp 下新建一个名为 missing1 的文件夹

### 1.2 查看程序使用手册

```
lin@xiaolin:~$ man touch
```



TOUCH(1) User Commands

TOUCH(1)

NAME

touch - change file timestamps

SYNOPSIS

touch [OPTION]... FILE...

DESCRIPTION

Update the access and modification times of each FILE to the current time.

A FILE argument that does not exist is created empty, unless -c or -h is supplied.

A FILE argument string of - is handled specially and causes touch to change the times of the file associated with standard output.

Mandatory arguments to long options are mandatory for short options to o.

-a change only the access time

-c, --no-create do not create any files

-d, --date=STRING

Manual page touch(1) line 1 (press h for help or q to quit)

图 2: man 查看程序 touch 的使用手册

### 1.3 使用程序

```
lin@xiaolin:/tmp/missing1$ touch semester
lin@xiaolin:/tmp/missing1$ ls
semester
```

图 3: 用 touch 在 missing1 文件夹中新建一个叫 semester 的文件

### 1.4 写入文件

```
lin@xiaolin:/tmp/missing1$ echo '#!/bin/sh' > semester
lin@xiaolin:/tmp/missing1$ cat semester
#!/bin/sh
lin@xiaolin:/tmp/missing1$ echo curl --head --silent https://missing.csail.mit.edu > semester
lin@xiaolin:/tmp/missing1$ cat semester
curl --head --silent https://missing.csail.mit.edu
```

图 4: 一行一行写入 semester 文件

注: '#' 和双引号具有特殊含义, 因此需加单引号以阻止转义。

### 1.5 列出文件信息

```
lin@xiaolin:/tmp/missing1$ ./semester
-bash: ./semester: Permission denied
lin@xiaolin:/tmp/missing1$ ls
semester
lin@xiaolin:/tmp/missing1$ ls -l
total 4
-rw-r--r-- 1 lin lin 51 Aug 30 10:24 semester
```

图 5: ./semester 命令无法执行, 显示没有可执行 x 权限

### 1.6 \sys

```
lin@xiaolin:/tmp/missing1$ cat /sys/class/power_supply/BAT1/capacity
68
```

图 6: 显示笔记本电量信息

## 1.7 运算符

-eq	检测两个数是否相等
-ne	检测两个数是否不相等
-gt	检测左边的数是否大于右边的
-lt	检测左边的数是否小于右边的
-ge	检测左边的数是否大于等于右边的
-le	检测左边的数是否小于等于右边的

```
lin@xiaolin:~$ [ 10 -eq 10 ]
lin@xiaolin:~$ echo $?
0
```

图 7: -eq

## 1.8 分支语句

```
#!/bin/bash
read -p "请输入您的年龄:" AGE
if [ $AGE -le 18 ];then
    echo "未成年"
elif [ $AGE -le 30 ];then
    echo "年轻人"
else
    echo "老年人"
fi

lin@xiaolin:~$ vim test.sh
lin@xiaolin:~$ bash test.sh
请输入您的年龄:19
年轻人
```

图 8: if 语句

## 1.9 循环语句

### 1.9.1 for 循环

### 1.9.2 while 循环

```
#!/bin/bash

s=0
for (( i=1;i<=100;i=i+1 ))
do
    s=$(( $s+$i ))
done
echo "The sum of 1+2+...+100 is : $s"

lin@xiaolin:~$ bash test.sh
The sum of 1+2+...+100 is : 5050
```

图 9: for 语句求 1-100 的和

```
#!/bin/bash
i=1
s=0
while [ $i -le 100 ]
do
    s=$(( $s+$i ))
    i=$(( $i+1 ))
done
echo $s

lin@xiaolin:~$ vim test.sh
lin@xiaolin:~$ bash test.sh
5050
```

图 10: while 语句求 1-100 的和

## 2 vim 部分

### 2.1 vimtutor

### 2.2 基本操作模式

#### 2.2.1 概述

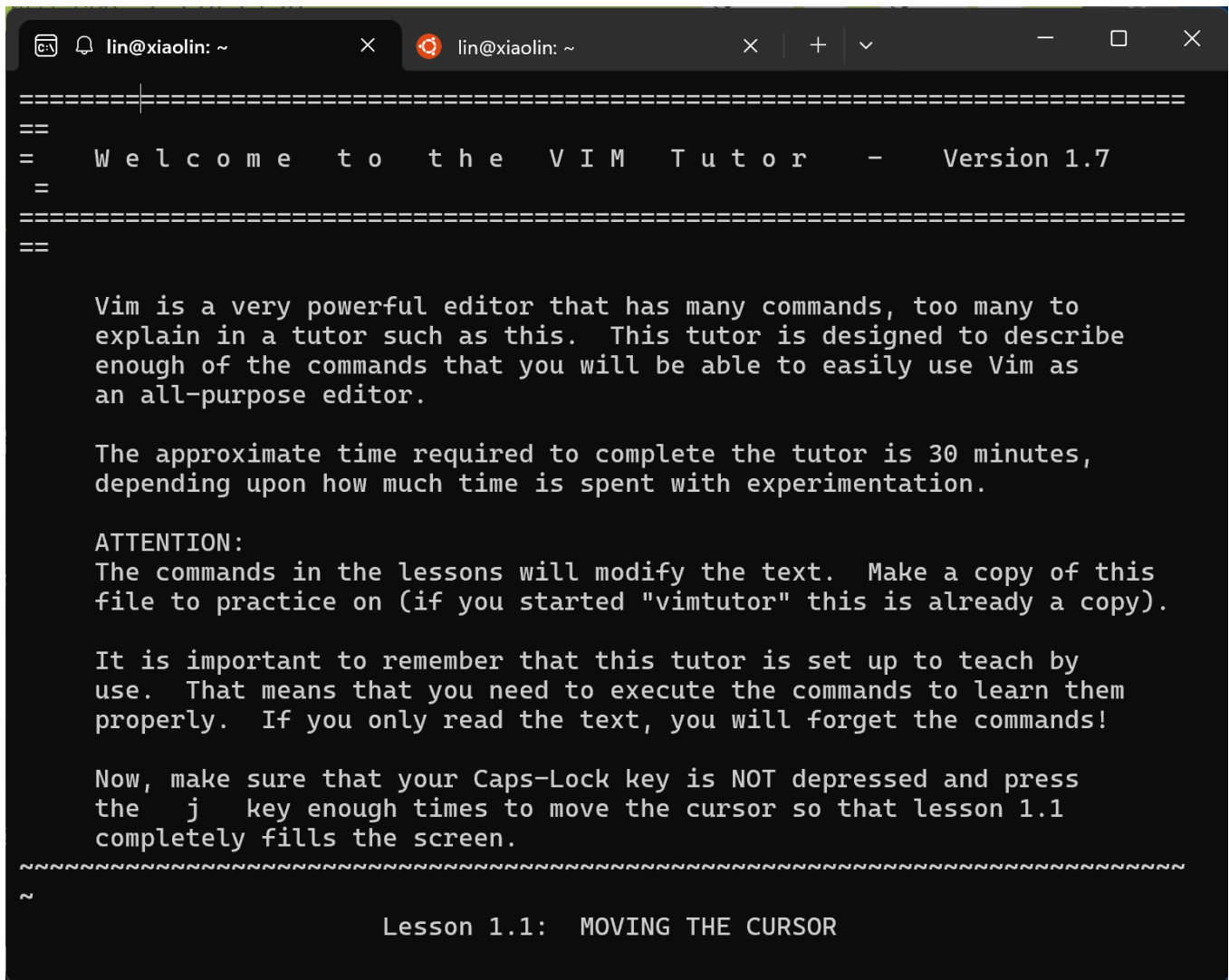
命令行模式：最初进入 vi 之后的模式，该模式下可以移动光标进行浏览，整行删除，但无法输入文字。

插入模式：在该模式下才能进行文字的编辑输入。

最底行模式：该模式下，光标位于屏幕底行，可以进行文件保存或退出操作，也可以设置编辑环境，如寻找字符串、列出行号。

#### 2.2.2 转换方式

在命令模式下按下 i 就进入了输入模式，使用 Esc 键可以返回到普通模式。在命令模式下按下: 就进入了底线命令模式；按 ESC 键可随时退出底线命令模式。



```
=====  
==  
=   W e l c o m e   t o   t h e   V I M   T u t o r   -   V e r s i o n   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 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 practice 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 Caps-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.  
~~~~~  
~  
  
Lesson 1.1:  MOVING THE CURSOR
```

图 11: ESc 进入命令模式后键入:q! 退出

## 2.3 移动操作

hkjl 四个命令用于移动光标: h 向左, l 向右, j 向下, k 向上。光标上下行切换的时候会尽可能保证光标位置不动。

## 2.4 搜索操作

在命令行模式下, 用 \ 要搜索的单词 使光标跳转到该单词前。

## 2.5 编辑操作

### 2.5.1 保存与退出

:w	保存
:w!	强制保存
:q	退出
:q!	强制退出
:wq or :x	保存退出
:wq! or ZZ	强制保存退出
:w /path/filename	将内容保存到指定位置
:r /path/filename	将内容从指定文件读取过来

### 2.5.2 删除文本

x or Delete	删除游标所在的字符
X	删除游标所在前一个字符
dd	删除整行
dw	删除一个单词（不适用中文）
d\$ or D	删除至行尾
d	删除至行首
dG	删除到文档结尾处
d1G	删至文档首部

### 2.5.3 替换操作

在命令行模式下输入 {作用范围}s/{目标}/{替换}/{替换标志}，例如%s/sort/array/g.

%	全局替换
. 或不输入	替换当前行
1,2	替换指定行

### 2.5.4 重复与撤销操作

重复：在命令行模式下输入.

撤销：在命令行模式下输入 u

## 3 数据整理

### 3.0.1 cut 列提取命令

cut [选项] 文件名

选项:

-f 列号: 提取第几列

-d 分隔符: 按照指定分隔符分割列

-n 取消分割多字节字符

-c 字符范围: 不依赖分隔符来区分队，而是通过字符范围（行首为 0）来进行字段提取。“n-”表示从第 n 个字符到行尾；“n-m”从第 n 个字符到第 m 个字符；“-m”表示从第 1 个字符到第 m 个字符



```
id      name    gender  mark
1       liming  m       86
2       sc     m       67
3       tg     n       90

lin@xiaolin:~$ vim stu.txt
lin@xiaolin:~$ cut -f 3 stu.txt
gender
m
m
n
```

图 12: 提取第 3 列内容

### 3.0.2 grep 命令

grep [选项] [正则表达式] [文件]

```
lin@xiaolin:~$ egrep "l" stu.txt
1       liming  m       86
```

图 13: 查询以 l 开头的字符（输出完整的行）

### 3.0.3 sed 命令

sed [选项] ‘[动作]’ 文件名

选项：-n：一般 sed 命令会把所有数据都输出到屏幕，如果加入此选项则只会把经过 sed 命令处理的行输出到屏幕

-e：允许对输入数据应用多条命令编辑

-i：用 sed 命令修改结果直接修改读取数据的文件，而不是由屏幕输出。

动作：

a：追加，在当前行后添加一行或多行

c：行替换，用 c 后面的字符串替换原数据行

i：插入，在当期行前插入一行或多行

d：删除，删除指定的行

p：打印，输出指定的行

s：字符串替换，用一个字符串替换另外一个字符串。格式为“行范围 s/旧字符串/新字符串/g”（和 vim 中的替换格式类似）

```
lin@xiaolin:~$ sed '2i test' stu.txt
id      name      gender  mark
test
1       liming    m       86
2       sc       m       67
3       tg       n       90
```

图 14: 插入一行

## 4 总结

本周学习了 shell 的基本命令、vim 的使用方法以及数据整理、正则表达式的简单操作，我感受到 shell 脚本运行开发效率很高，指令的堆积是可以看到的，好用精悍；第一次学习 vim，vim 的键盘快捷方式很难记忆，感觉很难上手，希望在后面的实践中能越用越熟练。