



实验报告

单衍喆

2024-9-6

GitHub地址: <https://github.com/Venusss1/course.git>

目录

1	实验内容	iii
2	实验设计	iii
2.1	Shell工具和脚本	iii
2.1.1	Linux系统环境配置	iii
2.1.2	基本命令	iii
2.1.3	环境变量	iv
2.1.4	文件操作	iv
2.1.5	Shell脚本	v
2.2	文本编辑工具Vim	v
2.2.1	编辑模式	v
2.2.2	基础命令	vi
2.2.3	操作符	vi

2.2.4	数量符	vii
2.3	数据整理	vii
2.3.1	sed流编辑工具	vii
2.3.2	正则表达式	viii

1 实验内容

1. Shell工具和脚本
2. 文档编辑工具Vim
3. 数据整理

2 实验设计

2.1 Shell工具和脚本

2.1.1 Linux系统环境配置

1. 下载安装VMware Workstation

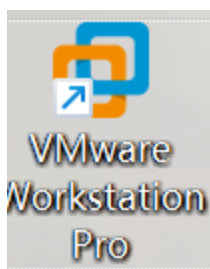


图 1: VMware

2. 下载Ubuntu系统文件
3. 载入系统文件
配置网络和系统设置

2.1.2 基本命令

表 1: 基本命令	
命令	作用
echo	标准输出
man	查看帮助
chmod	修改权限

2.1.3 环境变量

env 可查看当前系统中所有环境变量

env PATH 可以查看特定PATH的值

```

C:\CONFIG_SITE=/etc/config.site
PATH=/c/Users/86199/bin:/mingw64/bin:/usr/local/bin:/usr/bin:/bin:/mingw64/bin:/
usr/bin:/c/Users/86199/bin:/c/Program Files (x86)/VMware/VMware Workstation/bin:/
c/Program Files (x86)/Common Files/Intel/Shared Libraries/redirect/intel64/compil
er:/c/windows/system32:/c/windows:/c/windows/System32/wbem:/c/windows/System32/W
indowsPowerShell/v1.0:/c/windows/System32/OpenSSH:/c/Program Files (x86)/NVIDIA
Corporation/PhysX/Common:/c/Program Files/NVIDIA Corporation/NVIDIA NVDLISR:/c/P
rogram Files/Go/bin:/c/Program Files/mingw64/bin:/c/Program Files/PUTTY:/c/Progr
am Files/nodejs:/cmd:/c/Program Files (x86)/Vim/vim91:/c/Program Files/CMake/bin
:/c/Users/86199/AppData/Local/Microsoft/WindowsApps:/c/Users/86199/go/bin:/c/Use
rs/86199/AppData/Roaming/npm:/c/Users/86199/AppData/Local/Programs/Microsoft VS
Code/bin:/c/texlive/2024/bin/windows:/c/Program Files/nim-2.0.8/bin:/usr/bin/ven
dor_perl:/usr/bin/core_perl
  
```

图 2: PATH

2.1.4 文件操作

- pwd 展示当前目录下所有文件
- cd 切换目录
- ls 查看当前目录下所有文件
- mv 重命名和剪切文件
- cp 拷贝文件
- mkdir 新建文件夹
- touch 创建新文件

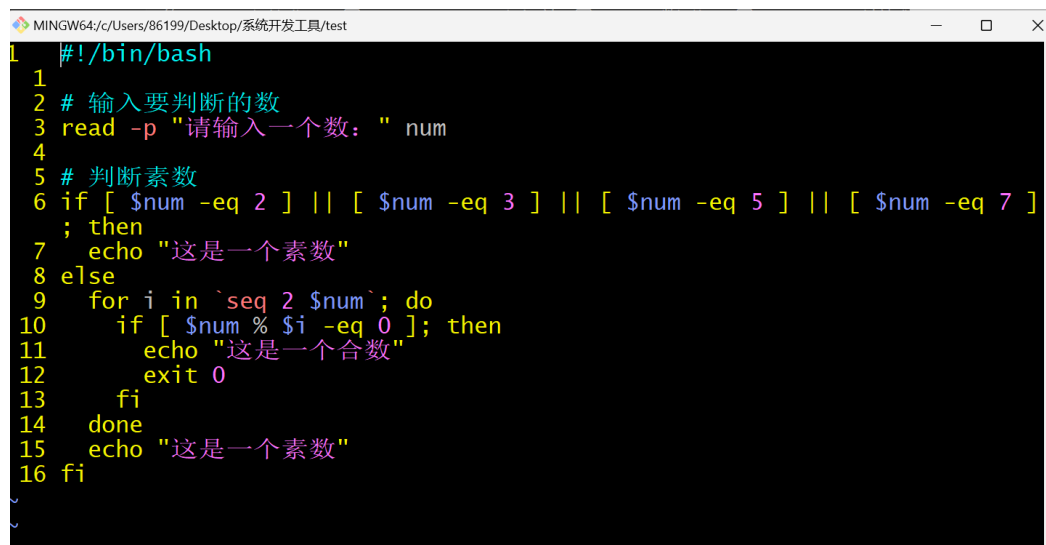
2.1.5 Shell脚本

- 赋值 foo=bar(不能用空格)
- \$ 访问变量
- 通配符

表 2: 基本命令

通配符	作用	示例
?	匹配单个字符	?.tex可通配a.tex
*	匹配任意个字符	*.tex可通配test.tex
{ }	自动合并	{text1,text2}.txt = text1.txt,text2.txt

利用shell脚本判断是否为素数



```
1 #!/bin/bash
2 # 输入要判断的数
3 read -p "请输入一个数: " num
4
5 # 判断素数
6 if [ $num -eq 2 ] || [ $num -eq 3 ] || [ $num -eq 5 ] || [ $num -eq 7 ]
7 ; then
8     echo "这是一个素数"
9 else
10     for i in `seq 2 $num`; do
11         if [ $num % $i -eq 0 ]; then
12             echo "这是一个合数"
13             exit 0
14         fi
15     done
16     echo "这是一个素数"
17 fi
```

图 3: 判断是否为素数

2.2 文本编辑工具Vim

2.2.1 编辑模式

- 正常模式: 默认模式, 输出esc回到正常模式, 移动光标

- 插入模式：在正常模式下输入i，进入插入模式,在插入模式下可输入文本
- 替换模式：在正常模式下输入R，进入替换模式，替换文本
- 可视化模式：在正常模式下输入v，进入可视化模式，可选中文本
- 命令模式：在正常模式下输入:，进入命令模式，可执行命令

2.2.2 基础命令

- 保存文件: w
- 退出: q
- 移动光标: h,j,k,l代表左,上,下,右
- 删除: x
- 追加文本: a,会同时进入插入模式
- 撤销命令: u; 取消撤销: U
- 粘贴命令: p
- 查找配对的括号: %
- 执行外部命令: !
- 在下方插入新行: o;在上方插入新行: O


2.2.3 操作符

- d 删除
- y 复制
- r 替换
- / 查找

2.2.4 数量符

- w 从当前光标到下一个单词起始
- \$ 从当前光标到行末
- 0 从当前光标到行首
- e 从当前光标到单词末尾
- 数字可以与数量符组

利用vim进行批量操作



```
利用vim进行批量操作  
利用vim进行批量操作  
利用vim进行批量操作  
利用vim进行批量操作  
利用vim进行批量操作  
利用vim进行批量操作  
利用vim进行批量操作  
利用vim进行批量操作
```

图 4: vim进行批量操作

2.3 数据整理

2.3.1 sed流编辑工具

- 命令
 - insert 插入
 - append 追加
 - delete 删除
 - copy 复制
 - substitute 替换

- print 打印
- 操作
 - -e 表达
 - -i 直接作用文件，生成备份
 - -n 忽略缓冲区
 - -f 读取多个文件

```
86199@gens MINGW64 ~/Desktop/系统开发工具/test
$ cat test3.txt
i have a dog
i have an apple
i like his dog
i like his apple

86199@gens MINGW64 ~/Desktop/系统开发工具/test
$ sed -r 's/^(^.)/@\1/' test3.txt
@i have a dog
@i have an apple
@i like his dog
@i like his apple
```

图 5: sed和正则替换文件内容

2.3.2 正则表达式

- 限定符
 - ? 前面的字符出现0此或1次
 - * 前面的字符出现0次或多次
 - + 前面的字符出现1次或多次
 - num 前面的字符出现num次，也可以是范围
- 定位符

- . 代表任何一个字符
- ? 可取消贪婪
- \$ 只匹配行尾

- 元字符

- \d 数字字符
- \D 非数字字符
- \w 单词字符（英文、数字、下划线）
- \W 非数字字符
- \s 空白符
- \S 非空白符



图 6: 匹配网址