

# 实验报告

程传哲

2024 年 9 月 1 日

## 1 学习成果

### 1.1 Shell

#### 1. echo \$SHELL

- 使用echo \$SHELL它去告诉系统这个脚本需要什么解释器来执行

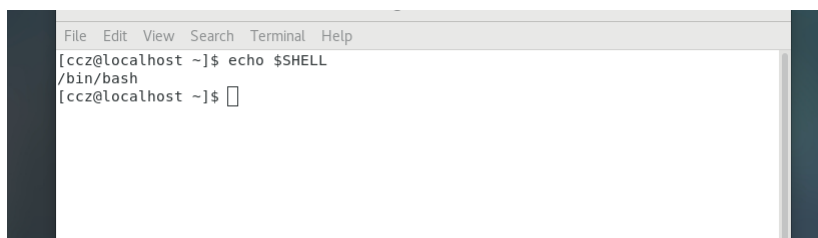
A terminal window with a menu bar (File, Edit, View, Search, Terminal, Help) and a dark background. The prompt is [ccz@localhost ~]\$. The command 'echo \$SHELL' is entered, and the output '/bin/bash' is displayed on the next line. The prompt returns to [ccz@localhost ~]\$.

图 1: figure title

- 显示为/bin/bash的时候即shell可以使用

#### 2. cd tmp – mkdir missing

- 使用cd tmp 去切换到tmp的目录下，在使用mkdir missing 去创建一个名为missing的文件夹

#### 3. touch semester

```
ccz@localhost:/tmp
File Edit View Search Terminal Help
[ccz@localhost ~]$ cd tep
bash: cd: tep: No such file or directory
[ccz@localhost ~]$ cd tmp
bash: cd: tmp: No such file or directory
[ccz@localhost ~]$ cd ..
[ccz@localhost home]$ ls
ccz
[ccz@localhost home]$ cd /
[ccz@localhost /]$ cd tmp
[ccz@localhost tmp]$ mkdir missing
[ccz@localhost tmp]$ ls
hsperfdata_root
ks-script-B3XPwk
ks-script-h5glzx
missing
ssh-nTRUv7dAehxW
ssh-u606AN3aYCjp
systemd-private-0130d1b9535c45b98f2656b4faf9e654-bolt.service-VGH9oS
systemd-private-0130d1b9535c45b98f2656b4faf9e654-chronyd.service-eR0r5n
systemd-private-0130d1b9535c45b98f2656b4faf9e654-colord.service-g0p7fE
systemd-private-0130d1b9535c45b98f2656b4faf9e654-cups.service-YGigbP
systemd-private-0130d1b9535c45b98f2656b4faf9e654-fwupd.service-gtXmJ3
systemd-private-0130d1b9535c45b98f2656b4faf9e654-rtkit-daemon.service-WzCEsa
systemd-private-8a4ab3016d2c4ffaa8025c15e8e69047-bolt.service-MxhzW0
```

图 2: figure title

- 使用semester可以在missing的目录下创建一个名为semester的文档

#### 4. man指令

- man指令的作用是查看各种命令的手册，man是manual的缩写
- 下面是使用man touch的结果

```
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
    too.

    -a      change only the access time

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

图 3: figure title

#### 5. cd ..或 cd /

- 使用cd ..是返回至上一目录，cd /是返回至根目录

## 6. pwd

- 使用pwd是查看当前所处的文件位置

## 7. ls

- 使用ls是去查看当前文件夹下的所有文件

当然ls后面可以接很多指令

-a	all显示所有文件及目录 (. 开头的隐藏文件也会列出)
-A	同-a但不列出 “.” (目前目录) 及 “...” (父目录)
-l	以长格式显示目录下的内容列表，包括文件的权限、链接数等
-d	仅列出目录本身，而不是列出目录内的文件数据
-r	reverse，将排序结果反向输出，例如：原本文件名由小到大
-S	sort by file size。根据文件大小排序，而不是文件名
-t	sort by modification time，以文件修改时间排序（从最新开始排）
-f	直接列出结果，而不进行排序（ls 默认以文件名排序）

## 8. echo hello >hello.txt

- 使用echo hello >hello.txt是将hello这个内容输入到hello.txt中

## 9. cat hello.txt

- 使用cat hello.txt可以显示出hello.txt的内容

## 10. cp拷贝文件

- 使用cp去拷贝文件有集中区分
  - 1.在当前目录下拷贝一个文件：cp <现存文件名><新文件名>
  - 2.复制文件到指定目录：cp <现存文件名><目标目录>
  - 3.复制目录及其内容：cp -r <现存目录名><目标目录>

## 11. mv

- 重命名文件

## 12. pipes管道

- |操作符允许我们将一个程序的输出和另外一个程序的输入连接起来

## 1.2 vim文本编辑器

### 1. ESC

- 使用ESC是为了退出编辑模式切换到命令模式

### 2. vi/vim

- 通过vi/vim进入命令模式

### 3. -i

- 切换到输入模式，在光标当前位置开始输入文本

### 4. -x

- 删除当前光标所在处的字符

### 5. -u

- 撤销上一次操作

### 6. -:

- 进入底线命令模式

### 7. -ctrl + r

- 重做上一次撤销的操作

### 8. : wq

- 保存文件并退出

## 1.3 shell脚本

### 1. 赋值语句

- `num = 100` 回车即可操作

### 2. 打印语句

- `echo "num = $num"` 回车即可操作

### 3. 输入读取语句

- `read num` 回车即可操作
- `read -p "请输入num的值" num` 其中加入-p可以同时显示与读取

### 4. 只读变量

- `readonly num = 10`回车即可操作后续将无法通过赋值语句改变num的量

## 2 Github链接

## 3 学习感悟

在本次的系统工具开发实践课中我受益颇多，在深入学习Shell与Vim的过程中，我经历了从初识的困惑到逐渐熟练运用的转变，这一过程不仅提升了我的技术能力，也让我对编程和命令行操作有了更深的理解和感悟，在学习shell中逐渐的了解了各个指令的用处以及shell脚本的使用，在vim的学习中学会了如何去使用vim键盘以及编写程序，这次的课程使我收益匪浅，也会继续在系统操作中继续学习下去