

中国海洋大学

系统开发工具基础实验报告

学生姓名：兰春光 学号：23020007053 指导老师：周小伟

实验时间：2024 年 9 月 5 日

目录

一 实验名称：shell 工具和脚本，Vim，数据整理	2
二 实验实例（内容和结果）	2
2 shell 工具和脚本	2
2.1 使用 echo \$SHELL 命令可以查看 shell	2
2.2 新建一个名为 missing 的文件夹	2
2.3 用 touch 在 missing 文件夹中新建一个叫 semester 的文件	2
2.4 将以下内容一行一行地写入 semester 文件：	3
2.5 尝试执行这个文件。	3
2.6 查看 chmod 的手册	4
2.7 使用 chmod 命令改变权限，使 ./semester 能够成功执行，不要使用 sh semester 来执行该程序。	4
2.8 使用 和 >，将 semester 文件输出的最后更改日期信息，写入主目录下的 last-modified.txt 的文件中	5
2.9 命令：ls -ahtl -color=auto，可实现所有文件（包括隐藏文件）、以人类可以理解的格式输出、以最近访问顺序排序、以彩色文本显示输出结果：	5
2.10 编写两个 bash 函数 marco 和 polo 执行下面的操作。	5
2.11 编写一段 bash 脚本，运行如下的脚本直到它出错，将它的标准输出和标准错误流记录到文件，并在最后输出所有内容。	6
2.12 按照最近的使用时间列出文件的命令：find . -type f -print xargs -d " " ls -t	7
2 Vim	7
2.1 vim 基本操作	7
2.2 替换	8
2 数据处理	8
2.1 mkdir	8
2.2 cat	8
2.3 grep	9
2.4 awk	9
2.5 sed	9

一 实验名称：SHELL 工具和脚本，VIM，数据整理2

2.6 rm 10

2.7 mv 10

三 解题感悟10

一 实验名称：shell 工具和脚本，Vim，数据整理

github 链接: <https://github.com/lan200503/System-Development-Tools->

二 实验实例（内容和结果）

2 shell 工具和脚本

2.1 使用 echo \$SHELL 命令可以查看 shell

```
lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell
$ echo $SHELL
/usr/bin/bash
```

图 1: echo \$shell

可以看到当前使用的是 bash

2.2 新建一个名为 missing 的文件夹

```
lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell
$ mkdir missing1
lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell
$ ls
file22.txt file3.txt missing/ missing-semester/ missing1/ semester1/
```

图 2: mkdir missing

2.3 用 touch 在 missing 文件夹中新建一个叫 semester 的文件

```
lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell
$ cd missing
lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell/missing
$ touch semester
lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell/missing
$ ls
semester
lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell/missing
$
```

图 3: touch semester

2.4 将以下内容一行一行地写入 semester 文件：

```
#!/bin/sh
curl -head -silent https://missing.csail.mit.edu
```

```
lan@LAPTOP-DVEQDM3F MINGW64 ~/learn-shell/missing
$ echo '#!/bin/sh' >> semester
lan@LAPTOP-DVEQDM3F MINGW64 ~/learn-shell/missing
$ echo 'curl -head --silent https://missing.csail.mit.edu' >> semester
bash: echo: command not found
lan@LAPTOP-DVEQDM3F MINGW64 ~/learn-shell/missing
$ echo 'curl -head --silent https://missing.csail.mit.edu' >> semester
bash: echo: command not found
lan@LAPTOP-DVEQDM3F MINGW64 ~/learn-shell/missing
$ cat semester
#!/bin/sh
lan@LAPTOP-DVEQDM3F MINGW64 ~/learn-shell/missing
$ echo 'curl -head --silent https://missing.csail.mit.edu' >> semester
bash: echo: command not found
lan@LAPTOP-DVEQDM3F MINGW64 ~/learn-shell/missing
$ echo 'curl -head --silent https://missing.csail.mit.edu' >> semester
bash: echo: command not found
lan@LAPTOP-DVEQDM3F MINGW64 ~/learn-shell/missing
$ echo 'curl -head --silent https://missing.csail.mit.edu' >> semester
lan@LAPTOP-DVEQDM3F MINGW64 ~/learn-shell/missing
$ cat semester
#!/bin/sh
curl -head --silent https://missing.csail.mit.edu
```

图 4: echo ” » <file>

2.5 尝试执行这个文件。

```
lan@LAPTOP-DVEQDM3F MINGW64 ~/learn-shell/missing
$ ./semester
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 8285
Server: GitHub.com
Content-Type: text/html; charset=utf-8
Last-Modified: Thu, 08 Aug 2024 20:16:01 GMT
Access-Control-Allow-Origin: *
ETag: "66b52781-205d"
expires: Tue, 03 Sep 2024 04:29:41 GMT
Cache-Control: max-age=600
x-proxy-cache: MISS
X-GitHub-Request-Id: 3DA6:3257C3:11AA297:1209311:66D68E5D
Accept-Ranges: bytes
Age: 560
Date: Thu, 05 Sep 2024 08:28:24 GMT
Via: 1.1 varnish
X-Served-By: cache-nrt-rjtf7700041-NRT
X-Cache: HIT
X-Cache-Hits: 0
X-Timer: S1725324905.634566,VSO,VE1
Vary: Accept-Encoding
X-Fastly-Request-ID: 4e922055e31efcda18d405dc6df2e237671ad301

lan@LAPTOP-DVEQDM3F MINGW64 ~/learn-shell/missing
$ ls -l
total 1
-rwxr-xr-x 1 lan 197121 61 Sep  5 16:26 semester*
```

图 5: ./semester

git bash 和 Linux 的结果不太一样。Linux 会显示权限不足

2.6 查看 chmod 的手册

```
lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell/missing
$ chmod --help
Usage: chmod [OPTION]... MODE[,MODE]... FILE...
or: chmod [OPTION]... OCTAL-MODE FILE...
or: chmod [OPTION]... --reference=FILE FILE...
Change the mode of each FILE to MODE.
With --reference, change the mode of each FILE to that of RFILE.

-c, --changes          like verbose but report only when a change is made
-f, --silent, --quiet  suppress most error messages
-v, --verbose          output a diagnostic for every file processed
    --no-preserve-root  do not treat '/' specially (the default)
    --preserve-root     fail to operate recursively on '/'
    --reference=RFILE   use RFILE's mode instead of MODE values
-R, --recursive        change files and directories recursively
-h, --help             display this help and exit
--version              output version information and exit

Each MODE is of the form '[ugoa]*([-+=]([rwxkst]*|[ugo]))+|[-+=][0-7]+'.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Report any translation bugs to <https://translationproject.org/team/>
Full documentation <https://www.gnu.org/software/coreutils/chmod>
or available locally via: 'info '(coreutils) chmod invocation'
```

图 6: chmod -help

2.7 使用 chmod 命令改变权限，使 ./semester 能够成功执行，不要使用 sh semester 来执行该程序。

```
lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell/missing
$ chmod +x semester

lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell/missing
$ ls -l
total 1
-rwxr-xr-x 1 lan 197121 61 Sep  5 16:26 semester*

lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell/missing
$ ./semester
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 8285
Server: GitHub.com
Content-Type: text/html; charset=utf-8
Last-Modified: Thu, 08 Aug 2024 20:16:01 GMT
Access-Control-Allow-Origin: *
ETag: "66b52781-205d"
Expires: Tue, 03 Sep 2024 04:29:41 GMT
Cache-Control: max-age=600
X-proxy-cache: MISS
X-Github-Request-Id: 3DA6:3257C3:11AA297:1209311:66D68E5D
Accept-Ranges: bytes
Age: 0
Date: Thu, 05 Sep 2024 08:37:59 GMT
Via: 1.1 varnish
X-Served-By: cache-nrt-rjtf7700059-NRT
X-Cache: HIT
X-Cache-Hits: 0
X-Timer: S172525479.098153,V50,V223
Vary: Accept-Encoding
X-Fastly-Request-ID: 4c7cdf46d633438208899cb677164c80971bf3ad
```

图 7: chmod +x

2.8 使用 | 和 >，将 semester 文件输出的最后更改日期信息，写入主目录下的 last-modified.txt 的文件中

```
lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell/missing
$ ./semester
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 8285
Server: GitHub.com
Content-Type: text/html; charset=utf-8
Last-Modified: Thu, 08 Aug 2024 20:16:01 GMT
Access-Control-Allow-Origin: *
ETag: "66b52781-205d"
expires: Tue, 03 Sep 2024 04:29:41 GMT
Cache-Control: max-age=600
X-proxy-cache: MISS
X-GitHub-Request-Id: 3DA6:3257C3:11AA297:1209311:66D68E5D
Accept-Ranges: bytes
Age: 560
Date: Thu, 05 Sep 2024 08:28:24 GMT
Via: 1.1 varnish
X-Served-By: cache-nrt-rjtf7700041-NRT
X-cache: HIT
X-cache-Hits: 0
X-Timer: S1725524905.634566,V50,V1
Vary: Accept-Encoding
X-Fastly-Request-ID: 4e922055e31efcda18d405dc6df2e237671ad301

lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell/missing
$ ls -l
total 1
-rwxr-xr-x 1 lan 197121 61 Sep  5 16:26 semester*
```

图 8: ./semester | grep "last-modified" > /last-modified.txt

2.9 命令：ls -ahtl -color=auto，可实现所有文件（包括隐藏文件）、以人类可以理解的格式输出、以最近访问顺序排序、以彩色文本显示输出结果：

```
lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell
$ ls -ahtl --color=auto
total 26K
drwxr-xr-x 1 lan 197121  0 Sep  5 22:57 ./
drwxr-xr-x 1 lan 197121  0 Sep  5 22:57 missing1/
drwxr-xr-x 1 lan 197121  0 Sep  5 22:57 semester1/
drwxr-xr-x 1 lan 197121  0 Sep  5 22:39 ../
-rw-r--r-- 1 lan 197121 17 Sep  5 22:39 file22.txt
drwxr-xr-x 1 lan 197121  0 Sep  5 21:56 missing-semester/
-rw-r--r-- 1 lan 197121 4.0K Sep  5 20:46 .swp
-rw-r--r-- 1 lan 197121  8 Sep  5 20:29 file3.txt
drwxr-xr-x 1 lan 197121  0 Sep  5 17:36 missing/
```

图 9: ls -ahtl -color=auto

2.10 编写两个 bash 函数 marco 和 polo 执行下面的操作。

每当你执行 marco 时，当前的工作目录应当以某种形式保存，当执行 last polo 时，无论现在处在什么目录下，都应当 cd 回到当时执行 marco 的目录。为了方便 debug，你可以把代码写在单独的文件 marco.sh 中，并通过 source marco.sh 命令，（重新）加载函数。

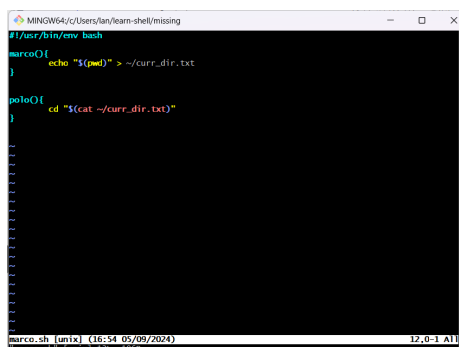


图 10: vim 编辑

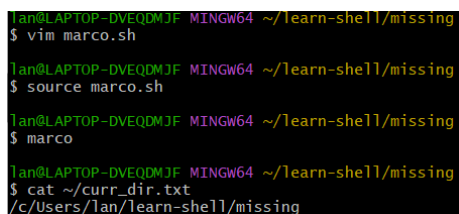
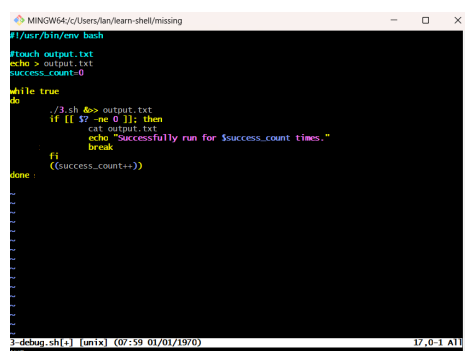


图 11: source marco.sh

2.11 编写一段 bash 脚本，运行如下的脚本直到它出错，将它的标准输出和标准错误流记录到文件，并在最后输出所有内容。

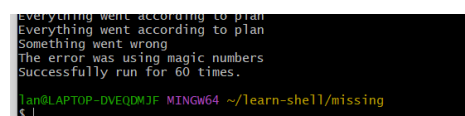
```
#!/usr/bin/env bash
n=$(( RANDOM % 100 ))
if [[ n -eq 42 ]]; then
    echo "Something went wrong"
    >2 echo "The error was using magic numbers"
    exit 1
fi
echo "Everything went according to plan"
```

将该脚本写入 3.sh 中



```
MINGW64~/Users/lan/learn-shell/missing
# /usr/bin/env bash
touch output.txt
echo > output.txt
success_count=0
while true
do
    ./3.sh &&> output.txt
    if [[ $? -ne 0 ]]; then
        cat output.txt
        echo "Successfully run for $success_count times."
        break
    fi
    ((success_count++))
done
3-debug.sh (*) [unix] (07:59 01/01/1970) 17.0-1 All
```

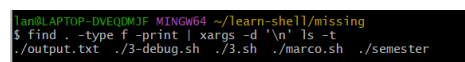
图 12: 3-debug



```
Everything went according to plan
Everything went according to plan
Something went wrong
The error was using magic numbers
Successfully run for 60 times.
lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell/missing
$
```

图 13: output

2.12 按照最近的使用时间列出文件的命令: `find . -type f -print | xargs -d "\n" ls -t`



```
lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell/missing
$ find . -type f -print | xargs -d '\n' ls -t
./output.txt ./3-debug.sh ./3.sh ./marco.sh ./semester
```

图 14: `find . -type f -print | xargs -d "\n" ls -t`

2 Vim

2.1 vim 基本操作

普通模式：这是 vim 的默认模式，用于输入命令。

插入模式：用于输入文本。

命令行模式：用于输入命令，如保存文件、退出等。

x: 删除光标所在位置的字符

dd: 删除整行

d\$: 删除从光标位置到行尾的内容

:w 保存

:q 退出

:wq 保存并退出

2.2 替换

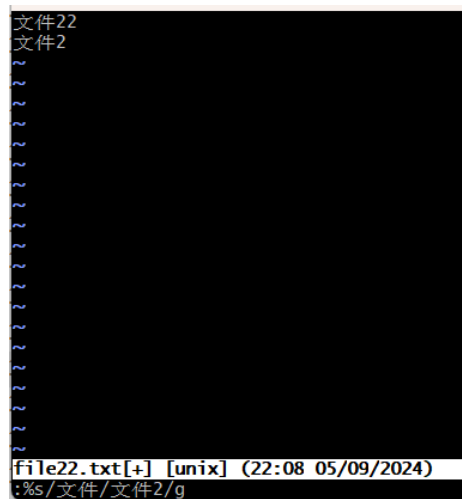


图 15: :%s/文件 2/文件 22/g

2 数据处理

2.1 mkdir

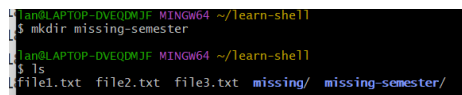


图 16: mkdir missing-semester

2.2 cat

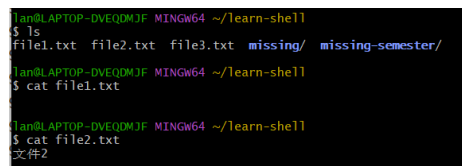


图 17: cat file2.txt

2.3 grep

```
lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell
$ cat file2.txt
文件2

lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell
$ echo "文件" >> file2.txt

lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell
$ cat file2.txt
文件2
文件

lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell
$ grep "文件" file2.txt
文件2
文件
```

图 18: grep "文件" file2.txt

2.4 awk

```
lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell
$ cat file2.txt
文件2
文件

lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell
$ awk 'NR==2' file2.txt
文件
```

图 19: awk 'NR==2' file2.txt

2.5 sed

```
lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell
$ echo " " > file1.txt

lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell
$ cat file2.txt
文件2
文件

lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell
$ sed 's/文件2/文件1/g' file2.txt > file1.txt

lan@LAPTOP-DVEQDMJF MINGW64 ~/learn-shell
$ cat file1.txt
文件1
文件
```

图 20: sed 's/old/new/g' file2.txt > file1.txt

2.6 rm

```
lan@LAPTOP-DVEQDM3F MINGW64 ~/learn-shell
$ ls
file1.txt file2.txt file3.txt missing/ missing-semester/
lan@LAPTOP-DVEQDM3F MINGW64 ~/learn-shell
$ rm file1.txt
lan@LAPTOP-DVEQDM3F MINGW64 ~/learn-shell
$ ls
file2.txt file3.txt missing/ missing-semester/
```

图 21: rm file1.txt

2.7 mv

```
lan@LAPTOP-DVEQDM3F MINGW64 ~/learn-shell
$ ls
file2.txt file3.txt missing/ missing-semester/
lan@LAPTOP-DVEQDM3F MINGW64 ~/learn-shell
$ mv file2.txt file22.txt
lan@LAPTOP-DVEQDM3F MINGW64 ~/learn-shell
$ ls
file22.txt file3.txt missing/ missing-semester/
```

图 22: mv file2.txt file22.txt

三 解题感悟

学习 Shell 工具和脚本、Vim 编辑器以及数据整理，让我深刻体会到了命令行的强大和灵活性。Shell 脚本自动化处理任务的能力极大地提高了工作效率。Vim 作为高效的文本编辑器，在代码编辑时更加灵活便捷。数据整理则让我更加熟悉对文件等一系列操作。希望以后能系统性地学习 Linux 系统。