



系统开发工具基础实验报告

题目：
Shell 工具和脚本; 编辑器（Vim）; 数据整理;

学生姓名 周洋迅 学号 24020007175
学部、学院 (中心) 信息科学与工程学部
专业 计算机科学与技术
日期 2025 年 8 月 29 日
github 链接 <https://github.com/zysgusg/SysDevelopmentTools>

中国海洋大学

目录

1	练习内容	2
1.1	Shell 命令	2
1.2	Shell 工具和脚本	2
1.3	vim	3
2	练习结果	3
2.1	结果截图	3
3	心得体会	10

1 练习内容

1.1 Shell 命令

1. 使用 `echo $SHELL` 命令可以查看您的 shell 是否满足要求。
2. 在 `/tmp` 下新建一个名为 `missing` 的文件夹。
3. 用 `man` 查看程序 `touch` 的使用手册。
4. 用 `touch` 在 `missing` 文件夹中新建一个叫 `semester` 的文件。
5. 将以下内容一行一行地写入 `semester` 文件：

```
1      #!/bin/sh
2      curl --head --silent https://missing.csail.mit.edu
3
```

6. 尝试执行这个文件。例如，将该脚本的路径 (`./semester`) 输入到您的 shell 中并回车。如果程序无法执行，请使用 `ls` 命令来获取信息并理解其不能执行的原因。
7. 查看 `chmod` 的手册 (例如，使用 `man chmod` 命令)
8. 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。您的 shell 是如何知晓这个文件需要使用 `sh` 来解析呢？
9. 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
10. 写一段命令来从 `/sys` 中获取笔记本的电量信息，或者台式机 CPU 的温度。

1.2 Shell 工具和脚本

11. 阅读 `man ls`，然后使用 `ls` 命令进行如下操作：-所有文件（包括隐藏文件）-文件打印以人类可以理解的格式输出 (例如，使用 `454M` 而不是 `454279954`) -文件以最近修改顺序排序 -以彩色文本显示输出结果
12. 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 `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。
13. 假设您有一个命令，它很少出错。因此为了在出错时能够对其进行调试，需要花费大量的时间重现错误并捕获输出。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。加分项：报告脚本在失败前共运行了多少次。

```
1      #!/usr/bin/env bash
2
3      n=$(( RANDOM % 100 ))
4
5      if [[ n -eq 42 ]]; then
6          echo "Something went wrong"
7          >&2 echo "The error was using magic numbers"
```

```

8      exit 1
9
10     fi
11
12     echo "Everything went according to plan"

```

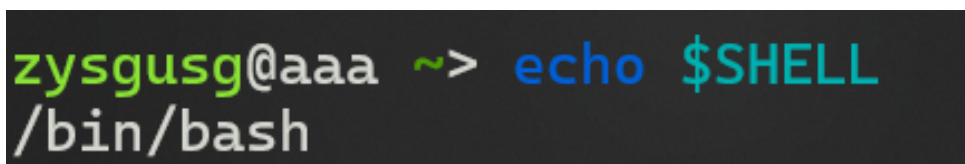
14. 您的任务是编写一个命令，它可以递归地查找文件夹中所有的 HTML 文件，并将它们压缩成 zip 文件。注意，即使文件名中包含空格，您的命令也应该能够正确执行
15. 编写一个命令或脚本递归的查找文件夹中最近修改的文件。更通用的做法，你可以按照最近的修改时间列出文件吗？

1.3 vim

16. 完成 vimtutor
17. 下载我们提供的 vimrc，然后把它保存到 ~/.vimrc。通读这个注释详细的文件（用 Vim!），然后观察 Vim 在这个新的设置下看起来和使用起来有哪些细微的区别。
18. 安装和配置一个插件：ctrlp.vim
19. 自定义 CtrlP：添加 configuration 到你的 ~/.vimrc 来用按 Ctrl-P 打开 CtrlP
20. 用 Vim 宏将 XML 转换到 JSON (例子文件)。尝试着先完全自己做，但是在你卡住的时候可以查看上面宏章节。

2 练习结果

2.1 结果截图

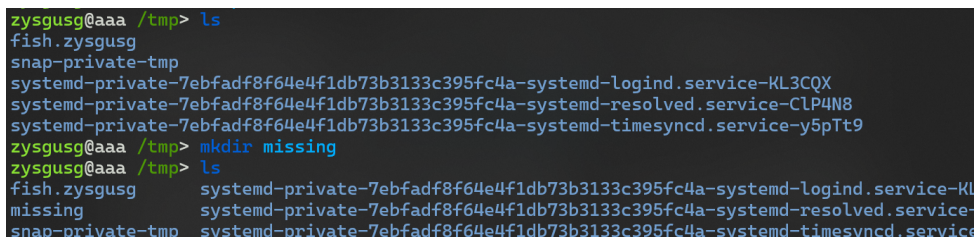


```

zysgusg@aaa ~> echo $SHELL
/bin/bash

```

图 1: echo \$SHELL



```

zysgusg@aaa /tmp> ls
fish.zysgusg
snap-private-tmp
systemd-private-7ebfadf8f64e4f1db73b3133c395fc4a-systemd-logind.service-KL3CQX
systemd-private-7ebfadf8f64e4f1db73b3133c395fc4a-systemd-resolved.service-CLP4N8
systemd-private-7ebfadf8f64e4f1db73b3133c395fc4a-systemd-timesyncd.service-y5pTt9
zysgusg@aaa /tmp> mkdir missing
zysgusg@aaa /tmp> ls
fish.zysgusg      systemd-private-7ebfadf8f64e4f1db73b3133c395fc4a-systemd-logind.service-KL
missing           systemd-private-7ebfadf8f64e4f1db73b3133c395fc4a-systemd-resolved.service-
snap-private-tmp  systemd-private-7ebfadf8f64e4f1db73b3133c395fc4a-systemd-timesyncd.service-

```

图 2: 新建一个名为 missing 的文件夹

```
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 as-
    sociated with standard output.

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

    -a      change only the access time

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

    -d, --date=STRING
            parse STRING and use it instead of current time

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

图 3: 用 man 查看程序 touch 的使用手册

```
zysgusg@aaa /tmp> touch missing/semester
zysgusg@aaa /tmp> cd missing
zysgusg@aaa /t/missing> ls
semester
zysgusg@aaa /t/missing> _
```

图 4: 新建一个叫 semester 的文件

```
vim semester /tmp/miss x + v
#!/bin/sh
curl --head --silent https://missing.csail.mit.edu
~
~
~
~
```

图 5: 写入 semester 文件

```
zysgusg@aaa /t/missing> ./semester
fish: The file './semester' is not executable by this user
zysgusg@aaa /t/missing [126]> ls -l
total 4
-rw-r--r-- 1 zysgusg zysgusg 61 Sep  5 09:41 semester
```

图 6: 程序无法执行, 没有执行权限

```

zyzgusg@aaa /t/missing> tldr chmod
chmod
Change the access permissions of a file or directory. More information: https://www.gnu.org/software/coreutils/chmod.

- Give the [u]ser who owns a file the right to e[x]ecute it:
  chmod u+x {{path/to/file}}

- Give the [u]ser rights to [r]ead and [w]rite to a file/directory:
  chmod u+rw {{path/to/file_or_directory}}

- Remove e[x]ecutable rights from the [g]roup:
  chmod g-x {{path/to/file}}

- Give [a]ll users rights to [r]ead and e[x]ecute:
  chmod a+rx {{path/to/file}}

- Give [o]thers (not in the file owner's group) the same rights as the [g]roup:
  chmod o=g {{path/to/file}}

- Remove all rights from [o]thers:
  chmod o= {{path/to/file}}

- Change permissions recursively giving [g]roup and [o]thers the ability to [w]rite:
  chmod -R g+w,o+w {{path/to/directory}}

- Recursively give [a]ll users [r]ead permissions to files and e[x]ecute permissions to sub-directories within a directory:
  chmod -R a+rx {{path/to/directory}}
zyzgusg@aaa /t/missing>

```

图 7: 查看 chmod 的手册

```

zyzgusg@aaa /t/missing> chmod a+rx semester
zyzgusg@aaa /t/missing> ./semester
HTTP/2 200
server: GitHub.com
content-type: text/html; charset=utf-8
last-modified: Thu, 28 Aug 2025 13:37:00 GMT
access-control-allow-origin: *
etag: "68b05b7c-2002"
expires: Thu, 04 Sep 2025 15:52:05 GMT
cache-control: max-age=600
x-proxy-cache: MISS
x-github-request-id: 29FC:3C4A54:29DDD0:2C3AC4:68B9B343
accept-ranges: bytes
age: 395
date: Fri, 05 Sep 2025 01:43:48 GMT
via: 1.1 varnish
x-served-by: cache-tyo11921-TY0
x-cache: HIT
x-cache-hits: 0
x-timer: S1757036629.623348,VS0,VE1
vary: Accept-Encoding
x-fastly-request-id: 83a888bc1e96597e28da04dd10a495f6120f576a
content-length: 8194

```

图 8: 使用 chmod 命令改变权限

```

zyzgusg@aaa /t/missing> ./semester | grep -F "last-modified" >~/last-modified.txt
zyzgusg@aaa /t/missing> cat ~/last-modified.txt
last-modified: Thu, 28 Aug 2025 13:37:00 GMT

```

图 9: 写入主目录下的 last-modified.txt 的文件

```

zyzgusg@aaa ~> cat /sys/class/power_supply/BAT1/capacity
100

```

图 10: 获取笔记本的电量信息

```
zysgusg@aaa /s/c/p/BAT1> ls -a -h --color=auto -lt
total 0
-rw-r--r-- 1 root root 4.0K Sep 5 09:56 alarm
-r--r--r-- 1 root root 4.0K Sep 5 09:56 capacity
-r--r--r-- 1 root root 4.0K Sep 5 09:56 capacity_level
-r--r--r-- 1 root root 4.0K Sep 5 09:56 cycle_count
lrwxrwxrwx 1 root root 0 Sep 5 09:56 device -> ../../PNP0C0A:00
-r--r--r-- 1 root root 4.0K Sep 5 09:56 energy_full
-r--r--r-- 1 root root 4.0K Sep 5 09:56 energy_full_design
-r--r--r-- 1 root root 4.0K Sep 5 09:56 energy_now
-r--r--r-- 1 root root 4.0K Sep 5 09:56 manufacturer
-r--r--r-- 1 root root 4.0K Sep 5 09:56 model_name
drwxr-xr-x 2 root root 0 Sep 5 09:56 power
-r--r--r-- 1 root root 4.0K Sep 5 09:56 power_now
-r--r--r-- 1 root root 4.0K Sep 5 09:56 present
-r--r--r-- 1 root root 4.0K Sep 5 09:56 serial_number
-r--r--r-- 1 root root 4.0K Sep 5 09:56 status
-r--r--r-- 1 root root 4.0K Sep 5 09:56 technology
-r--r--r-- 1 root root 4.0K Sep 5 09:56 voltage_min_design
-r--r--r-- 1 root root 4.0K Sep 5 09:56 voltage_now
-r--r--r-- 1 root root 4.0K Sep 5 08:36 type
drwxr-xr-x 4 root root 0 Sep 5 08:36 .
drwxr-xr-x 3 root root 0 Sep 5 08:36 ..
drwxr-xr-x 3 root root 0 Sep 5 08:36 hwmn1
lrwxrwxrwx 1 root root 0 Sep 5 08:36 subsystem -> ../../../../../../class/power_supply
-rw-r--r-- 1 root root 4.0K Sep 5 08:36 uevent
```

图 11: 使用 ls 命令

```
zysgusg@aaa ~/1> marco
/home/zysgusg/1
zysgusg@aaa ~/1> cd /sys/class/
zysgusg@aaa /s/class> polo
zysgusg@aaa ~/1> _
```

图 12: marco.sh 效果


```

zysgusg@aaa:~$ source catch_error.sh
error at 1 times
output
Something went wrong
error
The error was using magic numbers
zysgusg@aaa:~$ source catch_error.sh
error at 7 times
output
Something went wrong
error
The error was using magic numbers
zysgusg@aaa:~$ source catch_error.sh
error at 136 times
output
Something went wrong
error
The error was using magic numbers
zysgusg@aaa:~$ █

```

图 13: 运行次数

```

zysgusg@aaa:/tmp/html$ find . -type f -name "*.html" -print0 | xargs -d '\0' zip html.zip
adding: new_folder/2.html (stored 0%)
adding: 1.html (stored 0%)
zysgusg@aaa:/tmp/html$ ls
1.html  html.zip  new_folder

```

图 14: 递归地查找文件夹中所有的 HTML 文件，并将它们压缩成 zip 文件

```

zysgusg@aaa:~$ my_ls 5
-rw-r--r-- 1 zysgusg zysgusg      99 Sep  7 15:48 ./my_ls.sh
-rw-r--r-- 1 zysgusg zysgusg     21 Sep  7 14:51 ./output.log
-rw-r--r-- 1 zysgusg zysgusg    219 Sep  7 14:18 ./marco.sh
-rwxr-xr-x 1 zysgusg zysgusg    211 Sep  5 11:50 ./error.sh
-rw----- 1 zysgusg zysgusg     20 Sep  5 10:36 ./lessht
xargs: ls: terminated by signal 13

```

图 15: 按照最近的修改时间列出文件


```

This concludes the Vim Tutor. It was intended to give a brief overview of
the Vim editor, just enough to allow you to use the editor fairly easily.
It is far from complete as Vim has many many more commands. Read the user
manual next: ":help user-manual".

For further reading and studying, this book is recommended:
    Vim - Vi Improved - by Steve Oualline
    Publisher: New Riders
The first book completely dedicated to Vim. Especially useful for beginners.
There are many examples and pictures.
See http://iccf-holland.org/click5.html

This book is older and more about Vi than Vim, but also recommended:
    Learning the Vi Editor - by Linda Lamb
    Publisher: O'Reilly & Associates Inc.
It is a good book to get to know almost anything you want to do with Vi.
The sixth edition also includes information on Vim.

This tutorial was written by Michael C. Pierce and Robert K. Ware,
Colorado School of Mines using ideas supplied by Charles Smith,
Colorado State University. E-mail: bware@mines.colorado.edu.

Modified for Vim by Bram Moolenaar.

```

图 16: 完成 vimtutor

```

zysgusg@aaa:~$ mv vimrc ~/.vimrc
zysgusg@aaa:~$ vim ~/.vimrc

```

图 17: 完成 vimrc 设置

```

zysgusg@aaa:~$ mkdir -p ~/.vim/pack/vendor/start
zysgusg@aaa:~$ cd ~/.vim/pack/vendor/start
zysgusg@aaa:~/.vim/pack/vendor/start$ git clone https://github.com/ctrlpvim/ctrlp.vim
Cloning into 'ctrlp.vim'...
^C
zysgusg@aaa:~/.vim/pack/vendor/start$ git clone https://github.com/ctrlpvim/ctrlp.vim
Cloning into 'ctrlp.vim'...
remote: Enumerating objects: 4316, done.
remote: Counting objects: 100% (185/185), done.
remote: Compressing objects: 100% (113/113), done.
remote: Total 4316 (delta 78), reused 164 (delta 71), pack-reused 4131 (from 1)
Receiving objects: 100% (4316/4316), 1.71 MiB | 3.50 MiB/s, done.
Resolving deltas: 100% (1668/1668), done.

```

图 18: 安装插件

```

1 " 设置默认映射为 Ctrl+p
2 let g:ctrlp_map = '<c-p>'
3
4 " 搜索模式, 0 = 文件名, 1 = buffer, 2 = MRU (最近使用文件)
5 let g:ctrlp_cmd = 'CtrlP'
6 let g:ctrlp_working_path_mode = 'ra'
7
8 " 排除某些文件、文件夹
9 let g:ctrlp_custom_ignore = {
10 \ 'dir': '\v[\/]\.(git|hg|svn)$',
11 \ 'file': '\v\.(exe|so|dll)$',
12 \ 'link': 'some_bad_symbolic_links',
13 \ }
14
15 " 只在项目根目录下查找
16 let g:ctrlp_root_markers = ['.git', '.hg', '.svn', 'Makefile', 'package.json']
17 let g:ctrlp_working_path_mode = 'ra'
18
19 " 缓存文件列表, 加快打开速度
20 let g:ctrlp_use_caching = 1

```

图 19: 添加 configuration

```

c "a ggddGddo^[<80><fd>agg
c "b dd0{^[<80><fd>aj
c "c 0@da,^[<80><fd>aj
c "d f<r"f>r"a:"^[f<r"^[<80><fd>ald$
c "e 0@dj
c "f dd0}^[<80><fd>aj
c "g @b@c@e@f
c "h @a250@gi<80>kb^[<80><fd>a

```

图 20: Vim 宏

3 心得体会

- 通过本次实验，我学到了 shell 命令和 vim 相关的知识，加深了对 shell 脚本实现机制的理解。