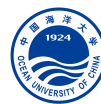


海纳百川
取则行远



中国海洋大学
OCEAN UNIVERSITY OF CHINA

Week2

左昊天

2024-08-30

Github 仓库地址

1 Shell

1.1 新建文件

1.1.1 创建任意类型的文件：

```
1 touch 文件名
```

1.1.2 创建可写入内容的文件：

```
1 echo "要写入的内容">test.txt
```

```
19355@-:~$ touch create.txt
19355@-:~$ echo "test">create2.txt
```

create.txt	2024-08-30 10:21	文本文档	0 KB
create2.txt	2024-08-30 10:21	文本文档	1 KB

1.2 新建文件夹

1.2.1 新建单个文件夹：

```
1 mkdir 文件夹名
```

1.2.2 新建多个文件夹：

```
1 mkdir 文件夹名1 文件夹名2 文件夹名3
```

```
19355@-:~$ mkdir 1 2 3
```

系统开发工具基础 > 1 > tool_class_2024_sum > week2 > 在 week2 中搜索

排序 查看

名称	修改日期	类型	大小
1	2024-08-30 10:48	文件夹	
2	2024-08-30 10:48	文件夹	
3	2024-08-30 10:48	文件夹	

1.2.3 新建多级文件夹:

```
1 mkdir -p 文件夹名1/文件夹名2/文件夹名3
```

```
19355@- 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum/week2/1 (main)
$ mkdir -p 1/2/3

week2 > 1 > 1 > 2 > 3
```

1.3 将命令一行一行写入脚本文件并执行

用 echo 逐行添加命令，再用 ./semester 执行文件。

注: > 会覆盖文件内容，» 用于在文件末尾追加内容

```
19355@- 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum/week2 (main)
$ echo '#!/bin/sh' > semester

19355@- 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum/week2 (main)
$ echo curl --head --silent https://missing.csail.mit.edu >> semester

19355@- 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum/week2 (main)
$ ./semester
HTTP/2 200
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: Mon, 26 Aug 2024 19:04:44 GMT
cache-control: max-age=600
x-proxy-cache: MISS
x-github-request-id: 4A7C:81A35:72A77:769D8:66CCCF74
accept-ranges: bytes
age: 0
date: Fri, 30 Aug 2024 03:10:17 GMT
via: 1.1 varnish
x-served-by: cache-nrt-rjtf7700053-NRT
x-cache: HIT
x-cache-hits: 0
```

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

1.4.1 管道操作符 |

| 用于将左边的命令的输出内容传递给右边的命令

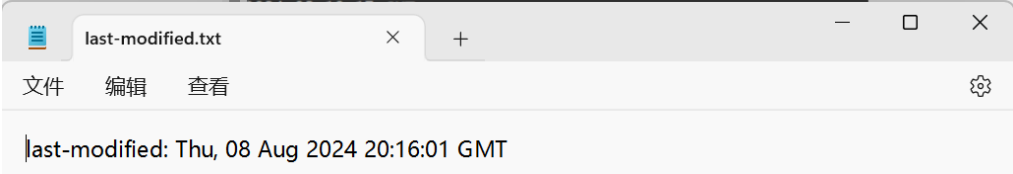
1.4.2 行过滤工具 grep

grep 用于查找符合条件的行并输出

grep last-modified 可以查找有 last-modified 字样的行并输出

1.4.3 实现

```
19355@Mingw64: ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum/week2 (main)
$ ./semester | grep last-modified > last-modified.txt
```



1.5 输入输出

read 变量名 可将用户输入内容赋值给变量

echo 输出的字符串中用\$变量名表示变量

```
#!/bin/bash
echo "请输入用户名:"
read name
echo "您好, $name"
```

echo -n 为不换行输出

示例:

```
1      #!/bin/bash
2
3      echo "请输入用户名:"
4      read name
5      echo -n "您好, $name"
6      echo "!!!"
```

```
$ ./test.sh
请输入用户名:
eric
您好, eric!!!
```

遇到的问题:

一开始会报错

```
$ ./test.sh
请输入用户名:
eric
./test.sh: line 5: echo您好, eric: command not found
```

原因: shell 脚本中, 命令和参数之间必须有空格。即 echo 和"输出的内容"间需有空格。

1.6 循环

1.6.1 for 循环

写法一:

```
1      for (( 表达式1;表达式2;表达式3 )) ; do
2          命令
3      done
```

写法二:

```
1  for i in {1..10}
2  do echo $i
3  done
```

```
$ ./for.sh
0
1
2
3
4
5
6
7
8
9
10
```

1.6.2 while 循环

```
1      while [[ $变量1 -lt $变量2 ]] do
2          命令
3      done
```

shell 中比较数值的操作符:

操作符	含义
-eq	等于
-ne	不等于
-ge	大于等于
-le	小于等于
-gt	大于
-lt	小于

1.7 猜字游戏

```
1      #!/bin/bash
2
```

```
3     number=$(shuf -i 1-10 -n 1)
4     while [[ $num -ne $number ]] do
5         echo "请输入1-10之间的数字"
6         read num
7         if [[ $num -eq $number ]];then
8             echo "恭喜你猜对了"
9         elif [[ $num -lt $number ]];then
10            echo "猜小了"
11        else
12            echo "猜大了"
13        fi
14    done
```

```
$ ./game.sh
请输入1-10之间的数字
5
猜大了
请输入1-10之间的数字
2
猜小了
请输入1-10之间的数字
4
恭喜你猜对了
```

1.8 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。

代码：

```
1     #!/bin/bash
2
3     marco(){
4         echo "$(pwd)" > C:/Users/19355/Desktop/大二暑假/
          系统开发工具基础/1/tool_class_2024_sum/week2/
          history.log
5         echo "已保存目录"
6     }
7
8     polo(){
9         cd "$(cat C:/Users/19355/Desktop/大二暑假/系统
```

```
    开发工具基础/1/tool_class_2024_sum/week2/  
    history.log)"  
10 }
```

解释:

- 1、\$()为命令替换符，可将内容替换为括号中命令执行后的内容。
- 2、cat 文件地址 用于打开文件并输出文件内容

运行截图:

```
19355@- ~ 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum  
/week2 (main)  
$ source marco.sh  
  
19355@- ~ 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum  
/week2 (main)  
$ marco  
已保存目录  
  
19355@- ~ 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum  
/week2 (main)  
$ cd ../  
  
19355@- ~ 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum  
(main)  
$ cd ../  
  
19355@- ~ 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1  
$ polo  
  
19355@- ~ 9 MINGW64 ~/Desktop/大二暑假/系统开发工具基础/1/tool_class_2024_sum  
/week2 (main)  
$ |
```

2 Vim

2.1 vim 中的不同模式以及进入方式

模式	进入方式
normal	vim 打开文件后默认进入的模式
insert	在 normal 下有多重方式进入。如输入 i, 在光标之前开始输入
	a 光标之后开始输入
	o 下方插入一行开始输入
	s 删除当前光标字符，开始输入
	I 从本行开头输入
	A 从本行结尾输入
	O 上方插入一行开始输入
	S 删除当前行并输入
command	在 normal 下输入: 进入
visual	在 normal 下输入 v 进入

2.2 vim 中的不同模式以及进入方式

2.3 退出 vim 的方式

在 normal 模式下输入:wq 为保存并退出

:q 为直接退出