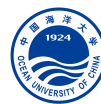


海纳百川  
取则行远



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

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

### 1.3 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@~$ echo '#!/bin/sh' > semester
19355@~$ echo curl --head --silent https://missing.csail.mit.edu >> semester
19355@~$ ./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

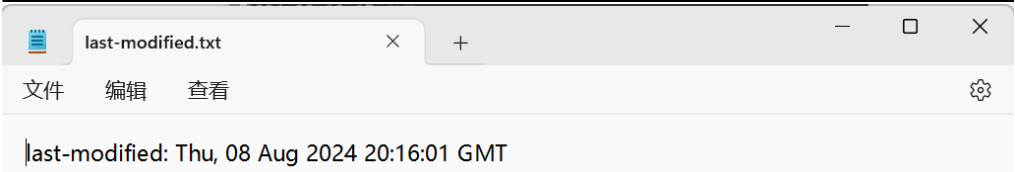
#### 1.4.2 行过滤工具 grep

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@~$ ./semester | grep last-modified > last-modified.txt
```



The screenshot shows a text editor window titled 'last-modified.txt'. The content of the file is 'last-modified: Thu, 08 Aug 2024 20:16:01 GMT'.

## 1.5 变量的输入输出

`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`$name`表示变量

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

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 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
```