**Linux应用开发**

**（2018年春）期末大作业报告**

组 长 姓 名： 何志颖 .

组 长 学 号： 20151120237 .

任 课 教 师 ： 朱艳萍 .

编 制 时 间 ： 2018/6/23 .

小组报告成绩：

**云南大学软件学院**

**2018年6月制表**

**小组成员及工作情况表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **姓 名（组长填第一行）** | **学 号** | **大作业工作量比例（%）** | **报告中主要参与的工作** | **成绩** |
| 何志颖 | 20151120237 | 50 | 完成了留言板程序部分及其该部分的文档 |  |
| 王秋莉 | 20151120247 | 50 | 完成了shell程序部分及其该部分的文档 |  |

目录

[一 程序描述与要求 5](#_Toc247763649)

[1 第一部分 shell编程 5](#_Toc1008556410)

[1.1 程序描述 5](#_Toc185107497)

[1.2 程序要求 6](#_Toc760078369)

[2 第二部分 linux下的程序设计部分 7](#_Toc1346797129)

[2.1 程序描述 7](#_Toc1078302545)

[2.2 程序要求 7](#_Toc746467686)

[二 程序测试调试的过程及结果记录 8](#_Toc1878185652)

[1 第一部分 shell编程的结果记录 8](#_Toc1951262974)

[1.1 选择usb功能 9](#_Toc1080603306)

[1.2 压缩功能 15](#_Toc1786363445)

[1.3 编辑文件属性功能 16](#_Toc854432699)

[1.4 系统时间功能 17](#_Toc1914522088)

[1.5 搜索功能 19](#_Toc1273397790)

[1.6 退出程序功能 20](#_Toc772647094)

[2 第二部分 linux下的程序设计的结果记录 21](#_Toc595227326)

[2.1 登录注册功能 22](#_Toc589113886)

[2.2 留言功能 26](#_Toc1616374749)

[三 程序部分关键源码展示 28](#_Toc1395293524)

[1 第一部分 shell编程的部分关键代码 28](#_Toc948090149)

[1.1 main模块的完整代码 28](#_Toc337408359)

[1.2 usb模块中的7个函数 30](#_Toc2053101294)

[1.3 压缩功能模块中的部分关键代码 32](#_Toc2060416090)

[1.4 编辑文件属性模块的部分关键代码 33](#_Toc1001680941)

[1.5 系统时间模块的部分关键代码 34](#_Toc511041493)

[1.6 搜索模块的部分关键代码 34](#_Toc1446519119)

[2 第二部分 linux下程序设计的部分关键代码 35](#_Toc2081865139)

[2.1 服务器端nodejs核心代码 35](#_Toc1741934562)

[2.2 客户端html/css/javascript核心代码 39](#_Toc432394423)

[四 期末课程小结 46](#_Toc181348699)

[1 王秋莉的课程学习小结 46](#_Toc250725126)

[2 何志颖的课程学习小结 48](#_Toc680158073)

# 一 程序描述与要求

## 1 第一部分 shell编程

### 1.1 程序描述

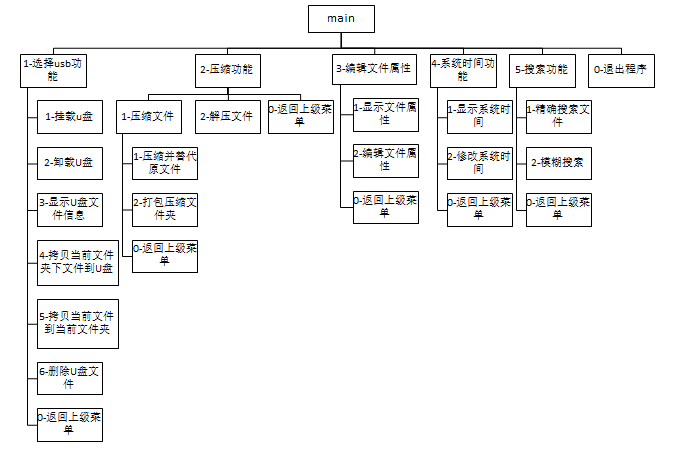
设计一个main程序，程序由多个文件组成，文件之间可以相互调用，并完成以下所示的模块功能，程序主要分成六个模块，主要功能如图1-1所示。

图1-1 程序模块图

### 1.2 程序要求

（1）程序总共由6个脚本文件构成：main(主程序)，usb部分，压缩部分，编辑文件属性部分，修改日期部分，搜索文件部分。

（2）main脚本

通过chmod +x filename实现对几个功能模块脚本文件属性的改变；

总的模块，通过./filename来实现对其他几个模块脚本程序的调用。

（3）其余功能的脚本文件将要实现的各功能用函数和相应的shell命令实现。

例如：usb脚本文件中基本应该包含7个函数

（4）编写各脚本的代码，并标注必要的注释或者说明；

（5）记录测试调试的过程及结果。

## 2 第二部分 linux下的程序设计部分

### 2.1 程序描述

设计一个程序通过用本地文件存储留言信息实现留言内容的保存，使得对象用户在下次打开时就能查看他人的留言信息。此留言板的功能有：用户可以对指定用户名进行留言、用户在输入用户名后查看自己的留言箱、用户可以选择清空留言箱内容、留言箱支持显示用户名、留言时间。

### 2.2 程序要求

（1）功能包括：用户可以对指定用户名进行留言、用户在输入用户名后查看自己的留言箱、用户可以选择清空留言箱内容、留言箱支持显示用户名、留言时间。

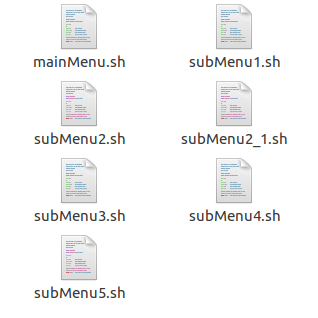
（2）程序中需标注必要的注释或者说明。

（3）记录测试调试的过程及结果。

# 二 程序测试调试的过程及结果记录

## 1 第一部分 shell编程的结果记录

程序总共由六个模块组成，即总共由6个脚本文件构成，分别为：main(主程序)，usb部分，压缩部分，编辑文件属性部分，修改日期部分，搜索文件部分。程序组成如图2-1所示：

图2-1 程序组成文件图

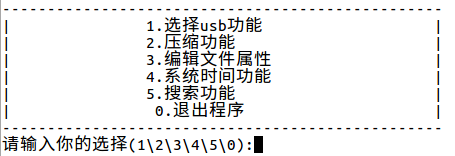
通过chmod +x filename实现对几个功能模块脚本文件属性的改变。命令如图2-2所示：



图2-2 修改脚本文件属性

### 1.1 选择usb功能

（1）通过./mainMenu.sh运行程序，初始运行结果如图2-3所示。

图2-3 初始运行结果

（2）输入“1”，选择usb功能，进入usb操作功能页面，如图2-4所示。

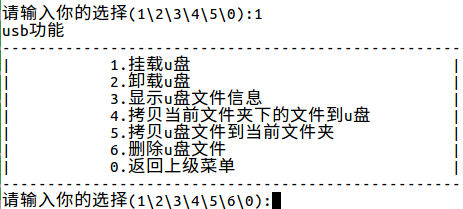


图2-4 usb功能目录

（3）在usb输入功能目录下，输入“1”可挂载u盘到/mnt/usb目录下；输入“2”，可卸载u盘。操作结果如图2-5所示。

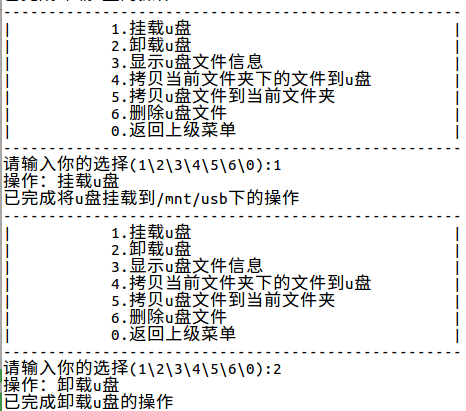


图2-5 挂载以及卸载u盘

（4）在将u盘挂载以后，输入“3”，即可显示u盘文件信息。操作结果如图2-6所示。

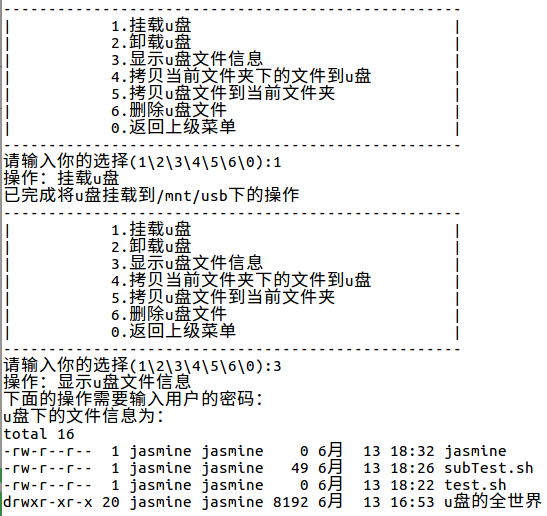


图2-6 显示u盘文件信息

（5）输入“4”，可拷贝当前文件夹下的指定文件或文件夹到u盘中。程序先在屏幕上输出当前文件夹下的文件信息，以供用户选择；然后询问用户要拷贝的文件或文件夹的名字；最后，告知用户操作结果，并输出操作完成后的u盘下的文件信息。整个操作结果如图2-7所示。

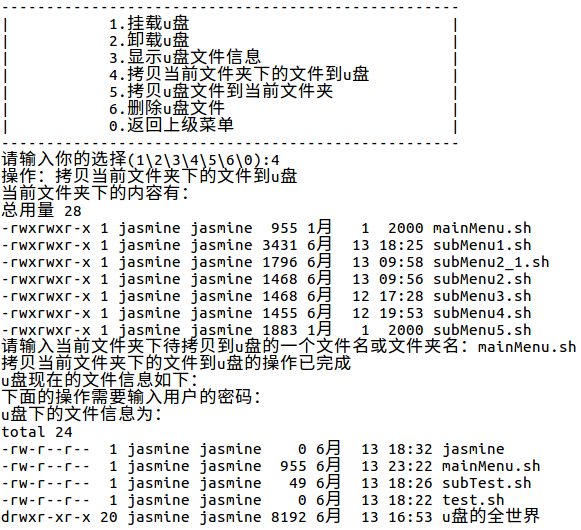


图2-7 拷贝当前文件夹下的文件到u盘

（6）输入“5”，可进行拷贝当前文件夹下的指定文件到u盘的操作。操作结果如图2-8所示。

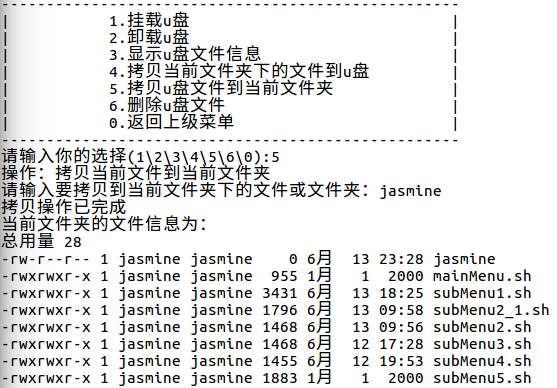


图2-8 拷贝当前文件夹下的文件到u盘

（7）输入“6”，可实现删除u盘中的指定文件或文件夹的操作。操作结果如图2-9所示。

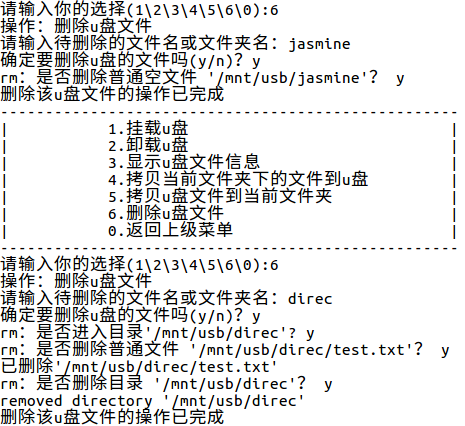


图2-9 删除u盘文件和文件夹

（8）输入“0”，即可放回上级菜单。操作结果如图2-10所示。

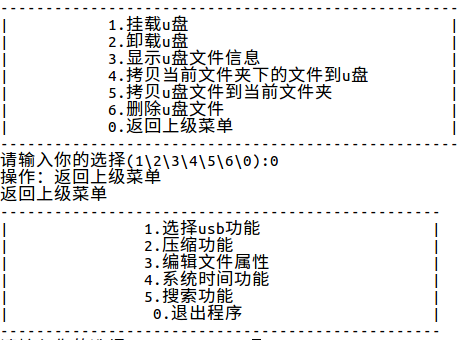


图2-10 返回上级菜单

### 1.2 压缩功能

输入“2”，即可进入压缩功能的菜单。如图2-11所示。

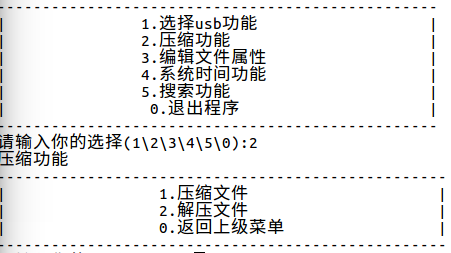


图2-11 进入压缩菜单

（1）输入“1“，即可进入具体压缩功能的子菜单。如图2-12所示。

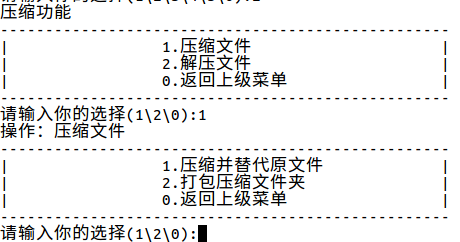


图2-12 具体压缩功能子菜单

（2）输入“1“，可进行压缩指定文件并替代原文件的操作。操作结果如图2-13所示。

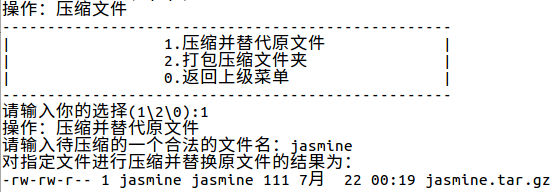


图2-13 压缩文件并替代原文件

（3）输入“2“，可进行压缩指定打包文件夹的操作，如图2-14所示。

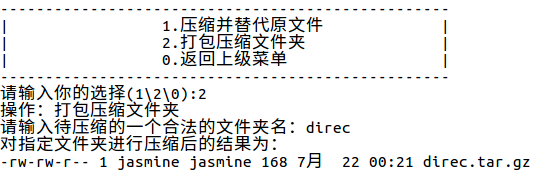


图2-14 压缩打包文件夹

### 1.3 编辑文件属性功能

输入“3”，即可进入编辑文件属性的菜单，如图2-15所示。

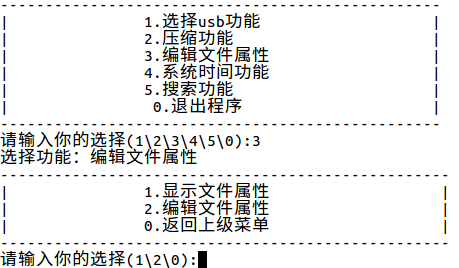


图2-15 进入编辑文件属性菜单

（1）输入“1“，可显示指定文件或文件夹的属性，如图2-16所示。

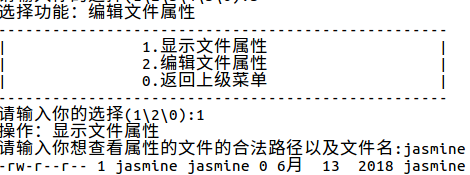


图2-16 显示文件属性

（2）输入“2“，可进行编辑指定文件属性的操作。操作结果如图2-17所示。

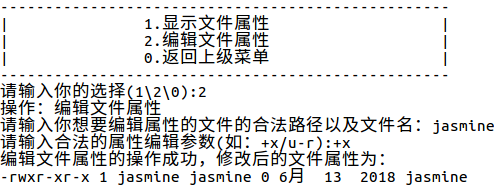


图2-17 编辑文件属性

### 1.4 系统时间功能

输入“4”，即可进入系统时间功能的菜单，如图2-18所示。

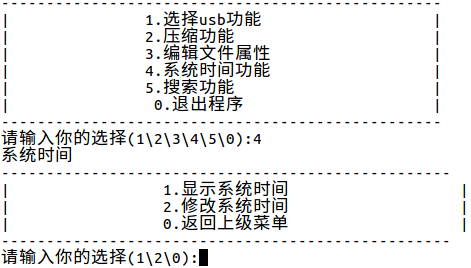


图2-18 进入系统时间功能菜单

（1）输入“1”，即可显示系统时间，如图2-19所示。

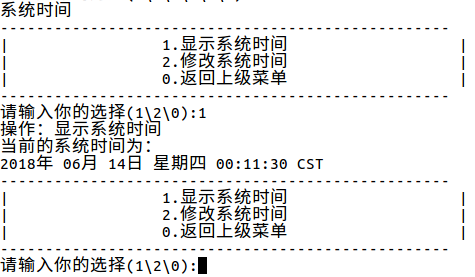


图2-19 显示系统时间

（2）输入“2”，即可修改系统时间，如图2-20所示。

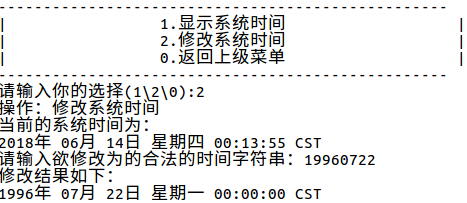


图2-20 修改系统时间

### 1.5 搜索功能

输入“5”，即可进入搜索功能的菜单，如图2-21所示。

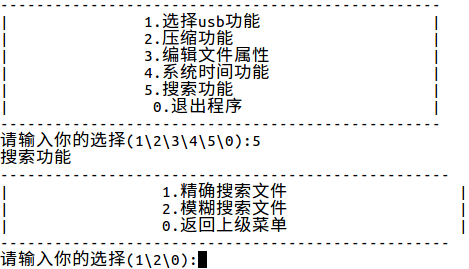


图2-21 进入搜索功能菜单

1. 输入“1”，可进行精确搜索的操作。通过指定要搜索的文件的文件名和搜索路径，进行搜索。操作结果如图2-22所示。

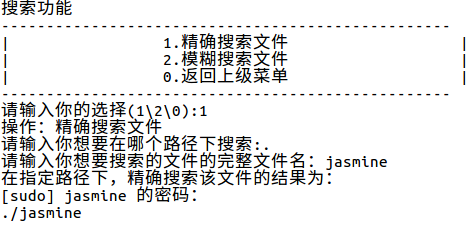


图2-22 精确搜索文件

1. 输入“”2，可进行模糊搜索的操作。通过指定要搜索的文件的属性、特征、搜索路径，进行搜索。操作结果如图2-23所示。

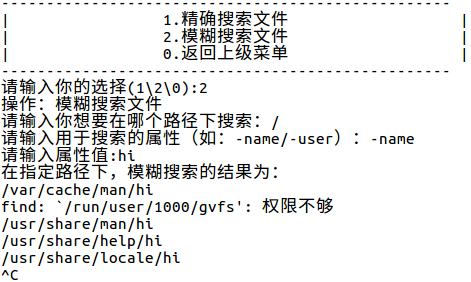


图2-23 模糊搜索文件

### 1.6 退出程序功能

输入“0”，即可退出程序，选择退出功能结果如图2-24所示。

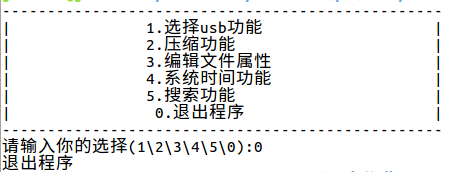


图2-24 退出程序结果

## 2 第二部分 linux下的程序设计的结果记录

项目：在线留言板

技术路线：前端：html/css/javascript/jquery

后端: node.js

数据: json格式文件

### 2.1 登录注册功能

1. 在线留言板提供登录注册的用户管理功能，如图2-25.



图2-25 留言板首页

1. 用户注册页面，如图2-26.



图2-26 用户注册页面

1. 用户注册用户名或密码为空时，提示错误信息，如图2-27.

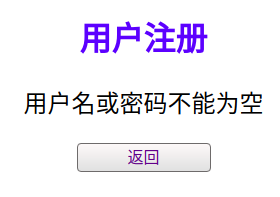


图 2-27 用户名密码不能为空

（4）注册成功，如图2-28.



图2-28 注册成功

1. 用户名已存在，提示错误信息，如图2-29.

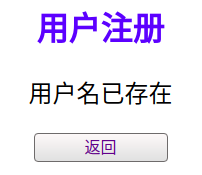


图2-29 用户名已存在

1. 用户登录界面，如图2-30.

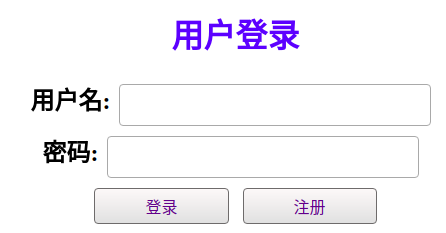


图2-30 用户登录界面

1. 用户名或密码错误，如图2-31.

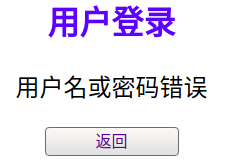


图2-31 用户名或密码错误

1. 成功登录，提示前往个人中心，如图2-32.

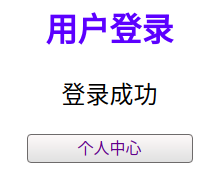


图2-32 登录成功，前往个人中心。

1. 个人中心界面，如图2-33.



图2-33 个人中心。

1. 注销功能，点击注销按钮后跳转到登录界面，如图2-34.

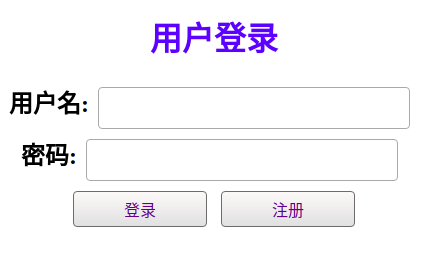


图2-33 注销登录后跳转到登录界面

### 2.2 留言功能

（1）用户登录后，在个人中心填写留言内容及其收信人，即可留言。如图2-34.



图2-34 留言功能

（2）若未填写收信人或留言内容，提示错误。如图2-35.

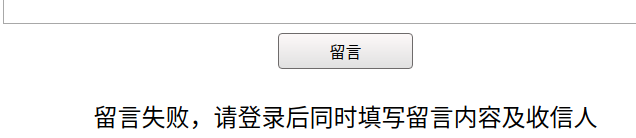


图2-35 留言错误提示

（3）点击我的留言，即可查看别人给用户的留言内容。如图2-36.



图2-36 我的留言

# 三 程序部分关键源码展示

## 1 第一部分 shell编程的部分关键代码

### 1.1 main模块的完整代码

#!/bin/bash

mainMenu(){

echo "-------------------------------------------------"

echo "| 1.选择usb功能 |"

echo "| 2.压缩功能 |"

echo "| 3.编辑文件属性 |"

echo "| 4.系统时间功能 |"

echo "| 5.搜索功能 |"

echo "| 0.退出程序 |"

echo "-------------------------------------------------"

read -p "请输入你的选择(1\2\3\4\5\0):" option

if [ $option -eq 0 ]

then

echo "退出程序"

return -1

exit 0

fi

case $option in

1)

echo "usb功能"

./subMenu1.sh

;;

2)

echo "压缩功能"

./subMenu2.sh

;;

3)

echo "选择功能：编辑文件属性"

./subMenu3.sh

;;

4)

echo "系统时间"

./subMenu4.sh

;;

5)

echo "搜索功能"

./subMenu5.sh

;;

\*)

echo "没有这个选项，请重新选择(1\2\3\4\5\0)！"

;;

esac

}

mainMenu

### 1.2 usb模块中的7个函数

# 1、挂载u盘

loadUsb(){

sudo mkdir /mnt/usb

sudo mount -t vfat /dev/sdb4 /mnt/usb

echo "已完成将u盘挂载到/mnt/usb下的操作"

}

# 2、卸载u盘

unloadUsb(){

sudo umount /dev/sdb4

sudo rmdir /mnt/usb

echo "已完成卸载u盘的操作"

}

# 3、显示u盘文件信息

showUsbFileInf(){

echo "下面的操作需要输入用户的密码："

sudo su - root <<EOF

cd

cd /mnt

cd usb

echo "u盘下的文件信息为："

ls -l

EOF

}

# 4、拷贝u盘文件到当前文件夹

copyFiles(){

read -p "请输入要拷贝到当前文件夹下的文件或文件夹：" name

path="/mnt/usb/"

cp -r "$path$name" .

echo "拷贝操作已完成"

echo "当前文件夹的文件信息为："

ls -l .

}

# 5、删除u盘文件

deleteUsbFile(){

read -p "请输入待删除的文件名或文件夹名：" name

read -p "确定要删除u盘的文件吗(y/n)？" isDelete

path="/mnt/usb/"

if [ $isDelete == "y" ]

then

if [ -f $name ]

then

rm -i "$path$name"

echo "删除该u盘文件的操作已完成"

else

rm -irv "$path$name"

echo "删除该u盘文件的操作已完成"

fi

else

echo "中止了删除文件的操作"

fi

}

#6、返回上级菜单

goBack(){

echo "返回上级菜单"

./mainMenu.sh

}

# 7、拷贝当前文件夹下的文件到u盘

copyFilesToUsb(){

echo "当前文件夹下的内容有："

ls -l .

read -p "请输入当前文件夹下待拷贝到u盘的一个文件名或文件夹名：" name

echo "拷贝当前文件夹下的文件到u盘的操作已完成"

cp -r $name /mnt/usb

echo "u盘现在的文件信息如下："

showUsbFileInf

}

### 1.3 压缩功能模块中的部分关键代码

# 1、解压文件

unzipFile(){

read -p "请输入带解压缩的压缩包名:" gizName

echo "对指定压缩包解压缩的结果为："

chmod +xrw $gizName

tar -xzf $gizName

ls -l

}

# 1、压缩并替代原文件

zip1(){

read -p "请输入待压缩的一个合法的文件名：" fileName

echo "对指定文件进行压缩并替换原文件的结果为："

suffix=".tar.gz"

tar -czf "${fileName}${suffix}" $fileName --remove-files

ls -l $fileName.\*

}

# 2、打包压缩文件夹

zip2(){

read -p "请输入待压缩的一个合法的文件夹名：" direc

echo "对指定文件夹进行压缩后的结果为："

suffix=".tar.gz"

tar -czf "${direc}${suffix}" $direc

ls -l $direc.\*

}

# 3、返回上级菜单

goBack(){

echo "返回上级菜单"

./subMenu2.sh

}

### 1.4 编辑文件属性模块的部分关键代码

# 1、显示文件属性

lsFileAttr(){

read -p "请输入你想查看属性的文件的合法路径以及文件名:" filePath

ls -l $filePath

}

# 2、编辑文件属性

chmodFile(){

read -p "请输入你想要编辑属性的文件的合法路径以及文件名：" filePath

read -p "请输入合法的属性编辑参数(如：+x/u-r):" signal

chmod $signal $filePath

echo "编辑文件属性的操作成功，修改后的文件属性为："

ls -l $filePath

}

### 1.5 系统时间模块的部分关键代码

showTime(){

echo "当前的系统时间为："

date

}

changeTime(){

showTime

read -p "请输入欲修改为的合法的时间字符串：" timeStr

echo "修改结果如下："

sudo date -s $timeStr

}

### 1.6 搜索模块的部分关键代码

# 精确搜索文件

exactFileSearch(){

read -p "请输入你想要在哪个路径下搜索:" path

read -p "请输入你想要搜索的文件的完整文件名：" fileName

echo "在指定路径下，精确搜索该文件的结果为："

sudo find $path -name $fileName

}

# 2、模糊搜索

fuzzyFileSearch(){

read -p "请输入你想要在哪个路径下搜索：" path

read -p "请输入用于搜索的属性（如：-name/-user）：" key

read -p "请输入属性值:" value

echo "在指定路径下，模糊搜索的结果为："

sudo find $path $key $value

}

## 2 第二部分 linux下程序设计的部分关键代码

### 2.1 服务器端nodejs核心代码

let express = require('express')

let app = express()

let bodyParser = require('body-parser')

let fs = require('fs')

let cookieParser = require('cookie-parser')

let users = []

let msgs = []

app.use(bodyParser.json())

app.use(bodyParser.urlencoded({

extended: true

}))

app.use(cookieParser())

fs.readFile('./user.json', function (err, data) {

if (err) {

console.log('read user.json err:', err)

return

}

if (data != false) {

console.log('user.json:', data)

users = JSON.parse(data)

console.log('parsed user.json:', users)

}

})

fs.readFile('./msg.json', function (err, data) {

if (err) {

console.log('read msg.json err:', err)

return

}

if (data != false) {

console.log('msg.json:', data)

msgs = JSON.parse(data)

console.log('parsed msg.json:', msgs)

}

})

app.use(function (req, res, next) {

res.set({

'Access-Control-Allow-Origin': '\*'

})

res.set({

'Access-Control-Allow-Headers': '\*'

})

res.set({

'Access-Control-Allow-Credentials': 'true'

})

next()

})

app.post('/login', function (req, res) {

let login\_success = false

let username = req.body.username

let password = req.body.password

users.forEach(one => {

if (one['username'] === username &&

one['password'] === password) {

login\_success = true

}

})

res\_content = login\_success ? {

success: true,

message: '登录成功',

username: username

} : {

success: false,

message: '用户名或密码错误'

}

console.log('res\_content', res\_content)

return res.json(res\_content)

})

app.post('/signup', function (req, res) {

let samename = false

let username = req.body.username

let password = req.body.password

users.forEach(one => {

if (one['username'] === username) {

samename = true

}

})

if (!samename) {

users.push({

username: username,

password: password

})

fs.writeFile('./user.json', JSON.stringify(users), function (err) {

if (err) {

console.log(err)

}

})

}

res\_content = samename ? {

success: false,

message: '用户名已存在'

} : {

success: true,

message: '注册成功'

}

console.log('res\_content', res\_content)

return res.json(res\_content)

})

app.post('/message', function (req, res) {

let from = req.body.from

let to = req.body.to

let msg = req.body.msg

let time = req.body.time

let res\_content = {}

if (from != false && to != false && msg != false) {

msgs.push({

from: from,

to: to,

msg: msg,

time: time

})

fs.writeFile('./msg.json', JSON.stringify(msgs), function (err) {

if (err) {

console.log('write msg.json err:', err)

return

}

})

res\_content = {

success: true,

message: '留言成功'

}

} else {

res\_content = {

success: false,

message: '留言失败，请登录后同时填写留言内容及收信人'

}

}

console.log(from + ' say ' + msg + ' to ' + to + ' in ' + new Date(time).toLocaleString())

return res.json(res\_content)

})

app.post('/mymsg', function (req, res) {

let username = req.body.username

let res\_content = {

success: true,

results: []

}

msgs.forEach(function (value) {

if (value.to === username) {

res\_content.results.push(value)

}

})

return res.json(res\_content)

})

let server = app.listen(3000, function () {

let host = server.address().address

let port = server.address().port

console.log('app listening at http://%s:%s', host, port)

})

### 2.2 客户端html/css/javascript核心代码

(1)主页html

<body>

<h1>留言板</h1>

<hr>

<div class="center">

<p>这是一个在线留言版，登录后即可查看来自网络上小伙伴们给你的留言哦！</p>

<a href="login.html">登录</a>

<a href="signup.html">注册</a><br>

<img src="images/lyb.png" alt="留言板" width="800" height="300"><br>

<p style="color: #999;">作者: 何志颖 王秋莉</p>

</div>

</body>

1. 登录页面html/javascript

<body>

<h1>用户登录</h1>

<div class="center login">

<label for="username">用户名:</label>

<input type="text" id="username" name="username">

<br>

<label for="password">密码:</label>

<input type="password" id="password" name="password">

<br>

<a href="#" id="submit">登录</a>

<a href="signup.html">注册</a>

</div>

<div class="center">

<a style="display: none;" id="personal-center" href="msg.html">个人中心</a>

<a style="display: none;" id="back" href="login.html">返回</a>

</div>

<script>

$(function () {

$('#submit').click(function () {

$.ajax({

url: 'http://127.0.0.1:3000/login',

type: 'post',

contentType: 'application/json',

dataType: 'json',

data: JSON.stringify({

'username': $('#username').val(),

'password': $('#password').val()

}),

success: function (data) {

if (data.success === true) {

$('.login').html('<p>'+data.message+'</p>')

$('#personal-center').css('display', 'inline')

document.cookie = 'username=' + data.username

} else if (data.success === false) {

$('.login').html('<p>'+data.message+'</p>')

$('#back').css('display', 'inline')

}

}

})

})

})

</script>

</body>

1. 注册页面html/javascript

<body>

<h1>用户注册</h1>

<div class="center signup">

<label for="username">用户名:</label>

<input type="text" id="username" name="username" required>

<br>

<label for="password">密码:</label>

<input type="password" id="password" name="password" required>

<br>

<a href="#" id="submit">注册</a>

<a href="index.html">返回</a>

</div>

<div class="center">

<div id="result"></div>

<a style="display: none;" id="back" href="signup.html">返回</a>

</div>

<script>

$(function () {

$('#submit').click(function () {

if ($('#username').val() === '' || $('#password').val() === '') {

$('.signup').html('<p>用户名或密码不能为空</p>')

$('#back').css('display', 'inline')

return

}

$.ajax({

url: 'http://127.0.0.1:3000/signup',

type: 'post',

contentType: 'application/json',

dataType: 'json',

data: JSON.stringify({

'username': $('#username').val(),

'password': $('#password').val()

}),

success: function (data) {

if (data.success === true) {

alert(data.message)

} else if (data.success === false) {

$('.signup').html('<p>' + data.message + '</p>')

$('#back').css('display', 'inline')

}

}

})

})

})

</script>

</body>

1. 个人中心html/javascript

<body>

<a href="login.html">注销</a>

<a href="mymsg.html">我的留言</a>

<h1 id="title"></h1>

<div class="center">

<hr>

<label for="say-to">留言给:</label>

<input type="text" id="say-to" name="say-to">

<br>

<textarea name="msg" id="msg" cols="150" rows="20"></textarea>

<br>

<a id="submit">留言</a>

<div id="result"></div>

</div>

<script>

$(function () {

let cookies = document.cookie.split('; ')

let cookie\_dict = {}

cookies.forEach(function (value) {

let name\_value = value.split('=')

cookie\_dict[name\_value[0]] = name\_value[1]

})

$('#title').html(document.cookie.split('=')[1] + ' 的留言板')

$('#submit').click(function () {

$.ajax({

url: 'http://127.0.0.1:3000/message',

type: 'post',

dataType: 'json',

contentType: 'application/json',

data: JSON.stringify({

from: cookie\_dict['username'],

to: $('#say-to').val(),

msg: $('#msg').val(),

time: new Date().getTime()

}),

success: function (data) {

if (data.success === true) {

alert(data.message)

} else {

$('#result').html('<p>'+data.message+'</p>')

}

}

})

})

})

</script>

</body>

1. 我的留言 html/javascript

<body>

<div id="title"></div>

<div id="content"></div>

<script>

$(function () {

let cookies = document.cookie.split('; ')

let cookie\_dict = {}

cookies.forEach(function (value) {

let cookie\_name = value.split('=')[0]

let cookie\_value = value.split('=')[1]

cookie\_dict[cookie\_name] = cookie\_value

})

$.ajax({

url: 'http://127.0.0.1:3000/mymsg',

type: 'post',

contentType: 'application/json',

dataType: 'json',

data: JSON.stringify({

username: cookie\_dict['username']

}),

success: function (data) {

if (data.success === true) {

$('#title').html('<h1>' + cookie\_dict['username'] + ' 的留言记录</h1>' +

'<div class="center">' +

'<a href="login.html">注销</a>' +

'<a href="msg.html">返回</a><hr>' + '</div>')

let content =

'<table class="center" border="1">\n<tr>\n<th>发送者</th>\n<th>内容</th>\n<th>时间</th>\n</tr>\n'

data.results.forEach(function (value) {

content += '<tr>\n<td>' + value['from'] + '</td>\n<td>' +

value['msg'] + '</td>\n<td>' +

new Date(value['time']).toLocaleString() + '</td>\n</tr>\n'

})

content += '</table>'

$('#content').html(content)

} else {

$('#title').html('<h1>404</h1>')

}

}

})

})

</script>

</body>

1. 样式代码 main.css

h1 {

text-align: center;

font-size: 200%;

color: blue;

}

a,

button {

display: inline-block;

text-decoration: none;

border-radius: 4px;

background-image: linear-gradient(rgb(253, 249, 249) 0%, rgb(226, 226, 226) 100%);

padding: 5px 50px;

margin: 5px;

border: 1px rgb(110, 107, 107) solid;

}

.center {

text-align: center;

}

p {

font-size: 150%

}

label {

font-size: 150%;

font-weight: 700;

}

input {

min-width: 300px;

min-height: 30px;

border-radius: 4px;

border: 1px #aaa solid;

padding: 5px;

margin: 5px;

}

table {

border-radius: 4px;

border: 2px rgb(68, 78, 173) solid;

margin: 0 auto;

}

th,

td {

padding: 20px;

font-size: 150%;

}

# 四 期末课程小结

## 1 王秋莉的课程学习小结

本人在本次大作业中，编码部分负责的是第一部分的shell编程。在完成这部分的脚本程序的编写的过程中，复习了本学期学到的很多的shell编程的知识点，同时，也遇到了许多的困难和问题。通过与组员何志颖讨论和利用网络资源，最终都解决了遇到的问题，完成了第一部分的代码的编写。对本次编程过程遇到的一些难点以及本学期的linux的学习的一些知识点做一下总结。

1. 在挂载u盘前，需要先搞清楚，u盘的自动挂载点。在挂载u盘前后，分别使用”fdisk -l”命令，就可从前后列出的设备的不同之处，找出u盘的自动挂载点。之后需要完成的挂载和卸载u盘，也就很简单了。
2. 在编写程序的过程中，出现了这样一个疑问：如何切换为root用户完成一段命令以后，再切换为原来的用户？在解决这个问题的过程中，学习到了很多知识。

首先是，搞清楚了关于su和su -的区别： 切换用户是可以使用su tom或者su - tom来实现，但是两者有区别。 su只是切换身份，但shell环境仍然是原用户的shell；su -是连用户的shell环境一起切换，切换了shell环境，PATH的环境变量才有效。

其次，搞清楚了su命令的不同参数的不同用法。切换用户只执行一条命令的可以用: su - user -c command;切换用户执行一个shell文件可以用:su - user -s /bin/bash shell.sh。

最后，从网上发现<<EOF这样的符号，了解了它的作用，学习了它的用法。

EOF本意是 End Of File，表明到了文件末尾。  
使用格式基本是这样的：   
命令 << EOF  
 内容段  
EOF  
将“内容段”整个作为命令的输入。其实，不一定要用EOF，只要是“内容段”中没有出现的字符串，都可以用来替代EOF，只是一个起始和结束的标志罢了。

需要特别注意的是，结尾的EOF一定要定格写！

1. 还弄清楚了压缩文件与归档文件的区别与联系。归档文件的作用是，将很多的文件及文件夹归为一个文件；而压缩文件的作用是，将一个大的文件或文件夹通过某种压缩算法，将它变成一个小的文件。它们之间的联系就是：在使用时，一般是归档后压缩的。
2. 忘记了root的密码时，通过“su -root”,“passwd”两条命令，就可以重设密码了。

## 2 何志颖的课程学习小结

本人在本次大作业中，主要编写了“在线留言板”程序，在Linux环境Ubuntu下采用nodejs的后台技术，html/css/javascript的前台技术，构建了一个在线可联网的留言系统，用户可以在注册登录后相互留言。在编程的过程中遇到一些问题，最终在查阅网络资料的帮助下克服，现在总结如下：

1. nodejs在后台接受前台发送过来的数据时，总是收不到数据。

这个问题产生的原因是在前台发送ajax请求给后台时，采用的是post请求，并且数据类型是application/json，而后台默认是不允许json类型的post请求直接访问的，通过查阅资料，最终了解到需要在后台发给前台的options响应报文中添加Access-Control-Allow-Headers: “post”;该http请求头允许post类型的所有数据请求，所以最后才能接收到前台发来的json数据。

1. 前台发送数据给后台时，总是产生诸如404,500等的错误。

这个问题产生的原因是在进行浏览器跨域访问资源时，浏览器有着跨域保护，最后通过查阅网络资料，了解到可以通过CORS技术进行解决，细节上是在后台发送给前台的http请求头中加入Access-Control-Allow-Origin: “\*”;允许所有的域名访问该服务，最终得以解决。

1. 如何保存用户的登录状态

通过查阅资料，最终采用浏览器的cookie保存用户的登录状态，来保证进行页面间跳转后，可以记录下用户的登录状态，保持登录状态。