



信息工程系

教 案

课程名称：智能终端开发技术

教 师：郑博伟

总 学 时：54 学时

理论学时：18

实训学时：36

上课班级：物联 241、物联三加 241、物联自主
241、物联三二 241

授课学期：2025-2026 学年第一学期

实验一 Linux 操作系统

【学习目标】

本课程的总体目标

Linux 为何物

Linux 历史简介及重要人物（思政：重点介绍国产操作系统及发展）

Linux 与 Windows 的不同

如何学习 Linux

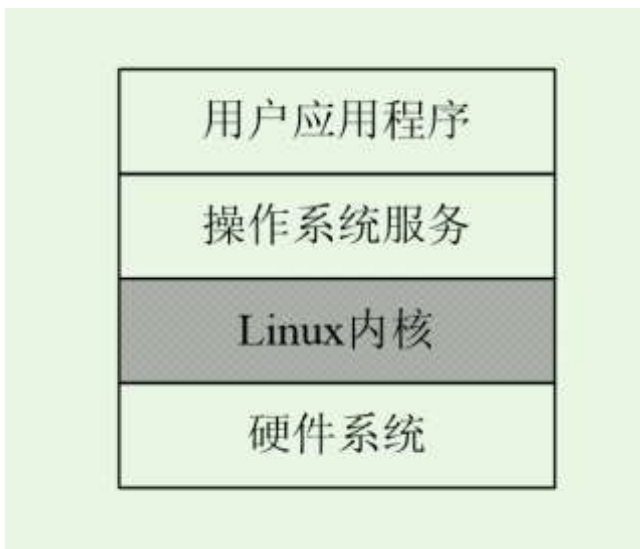
【重点难点】

Linux 的应用场景

【学习过程】

本周介绍 Linux 的入门概念。最基础的，即 Linux 是什么，涉及到操作系统的概念。介绍 Linux 的历史，让学生理解 Linux 的优缺点，以及掌握 Linux 的应用场景。

1. Linux 是一套操作系统。



2. Linux 的诞生

1994 年 Linux 第一个正式版本 1.0 发布。

Linux 内核最初只是由芬兰人李纳斯·托瓦兹（Linus Torvalds）在赫尔辛基大学上学时出于个人爱好而编写的。

Linux 能够在 PC 计算机上实现全部的 UNIX 特性。

3. Linux 的发展

Linux 在桌面应用、服务器平台、嵌入式应用等领域得到了大力发展。

Linux 具有完善的网络功能和较高的安全性，继承了 UNIX 系统卓越的稳定性表现。

在全球排名前 500 名的高性能计算机系统中，Linux 占了 90% 以上的份额。

云计算、大数据作为一个基于开源软件的平台，Linux 占据了核心优势。

4. Linux 的应用领域

今天各种场合都有使用各种 Linux 发行版，从嵌入式设备到超级计算机，并且在服务器领域确定了地位，通常服务器使用 LAMP（Linux + Apache + MySQL + PHP）或 LNMP（Linux + Nginx + MySQL + PHP）组合。

目前 Linux 不仅在家庭与企业中使用，并且在政府中也很受欢迎。

巴西联邦政府由于支持 Linux 而世界闻名。

有新闻报道俄罗斯军队自己制造的 Linux 发布版的，做为 G.H.ost 项目已经取得成果。

印度的 Kerala 联邦计划在向全联邦的高中推广使用 Linux。

中华人民共和国为取得技术独立，在龙芯过程中排他性地使用 Linux。

在西班牙的一些地区开发了自己的 Linux 发布版，并且在政府与教育领域广泛使用，如 Extremadura 地区的 gnuLinEx 和 Andalusia 地区的 Guadalinux。

葡萄牙同样使用自己的 Linux 发布版 Caixa Mágica，用于 Magalhães 笔记本电脑和 e-escola 政府软件。

法国和德国同样开始逐步采用 Linux。

5. Linux 发行版

很多的商业公司或非营利团体，将 Linux 内核(含 tools)与可运行的软件整合起来，加上自己具有创意的工具程序，这个工具程序可以让用户以光盘/DVD 或者透过网络直接安装/管理 Linux 系统。Kernel + Softwares + Tools，我们称之为 Linux distribution（发

行版)。

各大 Linux Distributions 的主要异同：每个 Linux distributions 使用的 kernel 都是 <http://www.kernel.org> 所释出的，而他们所选择的软件，几乎都是目前很知名的软件，重复性相当的高。每个 Linux distributions 除了架构的严谨度与选择的套件内容外，其实差异并不太大。



6. Ubuntu 的父版本 Debian

Debian 是极为精简的 Linux 发行版，操作环境干净，安装步骤简易。

Ubuntu 继承 Debian 的优点，集成在 Debian 下经过测试的优秀自由软件。

7. 自由软件的概念

自由软件的版权 GNU GPL。“Free software” refers to the users’ freedom to run, copy, distribute, study, change, and improve the software.

一个软件挂上了 GPL 版权宣告之后，就成了自由软件：

- 取得软件与原始码：你可以根据自己的需求来执行这个自由软件；
- 复制：你可以自由的复制该软件；
- 修改：你可以将取得的原始码进行程序修改工作，使之适合你的工作；
- 再发行：你可以将你修改过的程序再度自由发行；
- 回馈：你应该将你修改过的程序代码回馈于社群。

你所修改的任何一个自由软件都不应该也不能这样：

- 修改授权：你不能将一个 GPL 授权的自由软件，在你修改后而将他取消 GPL 授权。

➤ 单纯贩卖：你不能单纯的贩卖自由软件。

【学习小结】

通过本周的学习，学生了解 Linux 是什么，有什么应用场景，优缺点是什么。

实验二-2 文字模式界面入门

【实验目标】

学习图形界面和文字界面的切换方法

掌握重要热键的使用方法

学会使用帮助命令 `man`

掌握关机/重启命令的使用方法

思政：嵌入式领域里，国产龙芯用 Linux 开发，强调学习 Linux 的应用与必要性

【重点难点】

文字界面下的简单的命令

关机和重启的方法

【实验内容】

3. 图形模式和文字模式的切换

[Ctrl]+[Alt]+[F1~F6]: 文字接口登入 `tty1~tty6`

[Ctrl]+[Alt]+[F7]: 图形接口桌面

4. 重要热键

[Tab]: 命令补全/文件补全

[Ctrl] + c: 中断目前程序

[Ctrl] + d: 键盘输入结束

5. 帮助命令 `man`

man 命令: 进入该命令的 `man page`。

在 `man page` 界面中:

[空格键]: 向下翻一页

[Page Down]: 向下翻一页

[Page Up]: 向上翻一页
[Home]: 去到第一页
[End]: 去到最后一页
/string: 向下搜索字符串
?string: 向上搜索字符串
N, n: 继续向上/向下搜索字符串
q: 结束这次 man page

6. 关机、重启命令

shutdown [-t 秒] [-arkhncfF] 时间 [警告消息]

参数说明:

-t 秒: -t 后面加秒数, 亦即『过几秒后关机』的意思
-k : 不要真的关机, 只是发送警告信息出去
-r : 在将系统的服务停掉之后就重新启动
-h : 将系统的服务停掉后, 立即关机
-n : 不经过 init 程序, 直接以 shutdown 的功能来关机
-f : 关机并启动之后, 强制略过 fsck 的磁盘检查
-F : 系统重新启动之后, 强制进行 fsck 的磁盘检查
-c : 取消已经在进行的 shutdown 命令内容。

时间: 这是一定要加入的参数, 指定系统关机的时间。

例子:

shutdown -h now: 立刻关机

shutdown -h 20:25: 系统在 20:25 会关机

shutdown -h +10: 系统再过十分钟后自动关机

shutdown -r now: 系统立刻重新启动

shutdown -r +30 'The system will reboot': 再过 30 分钟系统会重新启动, 并显示后面的信息给所有在线的使用者

shutdown -k now 'This system will reboot': 仅发出警告信息, 系统并不会关机

【学习小结】

通过本周的学习，学生学会如何进入文字模式，掌握重要热键的使用方法、帮助命令 `man` 的使用方法，以及如何关机/重启系统。

实验三 目录管理

【实验目标】

了解 Linux 的目录配置标准、Linux 目录树

理解绝对路径和相对路径

掌握基本的目录操作

思政：与常见 Windows 下操作对比优劣，强调优化创新

【重点难点】

目录树原理

绝对路径和相对路径

基本的目录操作

【实验内容】

1. 开机，用户登录。
2. 进入文字界面或使用虚拟终端软件。
3. 查看当前目录。（提示：用 `pwd` 命令）
4. 切换到根目录。（提示：用 `cd /` 命令）
5. 查看根目录下的所有文件和目录。（提示：用 `ls -a` 命令）
6. 查看目录 `/usr` 下的所有文件和目录。（提示：用 `ls -a /usr` 命令）
7. 查看目录 `/var` 下的所有文件和目录。
8. 回到主目录。（提示：用 `cd ~` 命令）
9. 查看主目录下的所有文件和目录。
10. 在主目录下创建一个名为 `test1` 的目录。（提示：用 `mkdir test1` 命令）
11. 在 `test1` 目录下创建一个名为 `test2` 的目录。在 `test2` 目录下创建一个名为 `test3` 的目录。在 `test3` 目录下创建一个名为 `test4` 的目录。（提示：用 `mkdir -p test1/test2/test3/test4` 命令）
12. 切换到目录 `test3`。（提示：用 `cd ~/test1/test2/test3` 命令）

13. 切换到目录 `test2`。(提示：用 `cd ..`命令)
14. 切换到主目录。(提示：用 `cd ~`命令)
15. 把目录 `test4` 移除。(提示：用 `rmdir test1/test2/test3/test4` 命令)
16. 把目录 `test1`, `test2`, `test3` 移除。(提示：用 `rmdir -p test1/test2/test3` 命令)
17. 在主目录下创建分别名为 `test5` 和 `test6` 的两个目录。
18. 把 `test5` 复制到 `test6` 目录下。(提示：使用 `cp -r ~/test5 ~/test6` 命令)
19. 查看目录 `test6` 下的内容。
20. 移除目录 `~/test5`。(提示：使用 `rm -r ~/test5` 命令)
21. 把 `~/test6/test5` 目录移动到主目录。(提示：使用 `mv ~/test6/test5 ~`)
22. 查看主目录和目录 `test6` 下的内容。
23. 把 `test5` 重命名为 `test7`。(提示：使用 `mv ~/test5 ~/test7`)
24. 查看主目录下的内容。
25. 删除 `test6` 和 `test7`。
26. 关机。(提示：用 `shutdown -h now`)

【学习小结】

通过本周的学习，学生了解 Linux 的目录配置标准、Linux 目录树，理解绝对路径和相对路径的概念和使用方法，熟悉掌握基本的目录操作。

实验四 文本文件

【实验目标】

学习使用文本编辑器 nano

掌握各种文本文件查看目录的使用方法

理解文件的时间参数

思政：如何快速整合好自己的文件，增强信心

【重点难点】

文本文件的查看命令

【实验内容】

1. 查看文件/etc/apt/sources.list 的类型。（提示：使用命令 `file /etc/apt/sources.list`）
2. 把文件/etc/apt/sources.list 复制到当前用户的主目录。（提示：使用命令 `cp /etc/apt/sources.list ~`）
3. 使用 man 命令查看 cat、tac、nl、more、less、head、tail 等文本文件查看命令的用法。（提示：如 `man cat`）
4. 分别使用 cat、tac、nl、more、less、head、tail 命令查看~/sources.list 的内容，比较各个命令的异同。
 - a) 如果要显示行号，应该用什么命令？
 - b) 由于文本内容多于一页，要用什么命令翻页查看？
 - c) 如何查找文本中的某个字符串？
 - d) 既要显示行号，又要翻页查看，可以用什么命令？（提示：`nl ~/sources.list | less`）
5. 分别查看/etc/apt/sources.list 和~/sources.list 的 mtime。（提示：用命令 `ls -l /etc/apt/sources.list` 和 `ls -l ~/sources.list`）
6. 使用 Nano 编辑器编辑~/sources.list 的内容，然后保存。
7. 查看~/sources.list 的 mtime 是否发生变化。

【学习小结】

通过本周的学习，学生学会如何使用文本编辑器 `nano`；掌握各种文本文件查看目录的使用方法，比较它们的异同；理解文件的时间参数，学会查看的方法。

实验五 文件权限管理（一）

【实验目标】

理解用户和用户组的概念

掌握查看文件属性的命令

【重点难点】

理解权限

查看和修改权限的方法

【知识点内容】

1. Linux 目录 (文件夹)结构

■ 文件所有者 (owner)

文件属于谁，即用户，谁创建的。

■ 用户组

用户的集合。体现共享的特性。

注意：每个用户可以有多个用户组的支持

■ 其他人

除了用户组之外的集合。注意，root 也属于其他，但是 root 用户可以访问用户组的内容。

■ “所有者、用户组、其他人”可设置功能：

无、只能列出文件、创建和删除文件、访问文件

2. 在目录(文件夹)下的普通文件：

文件所有者 (owner)

● 用户组

● 其他人

● “所有者、用户组、其他人”可设置功能：

■ 读 (r)：读取文件内容或者查看目录。

- 写 (w): 修改文件内容或者创建、删除文件。
- 执行 (x): 执行文件或者允许使用 cd 命令进入目录。

3. 查看文件属性的命令

ls -al [目录/文件名]

```
-rw-rw-r-- 1 zbw zbw 2911 2月 8 13:18 sources.list  
[文件权限] [连接数] [所有者] [所属组] [容量] [修改日期] [文件名]
```

【实验内容】

- 以普通用户身份登录，查看根目录 (/) 下的所有文件和目录的属性。(提示: ls -al /)
 - 有哪些是目录，哪些是文件?
 - 能查询哪些目录下有哪些目录和文件名?
 - 其中哪些目录是可以进入的? 哪些不能?
 - 能在哪些目录下新建文件或目录?
- 查看文件/etc/apt/sources.list 的属性。(提示: ls -l /etc/apt/sources.list)
 - 你能读取它的内容吗?
 - 能修改它的内容吗?
- 把文件/etc/apt/sources.list 复制到用户的主目录 (~) 下，查看~/sources.list 的属性。
 - 你能读取它的内容吗?
 - 能修改它的内容吗?

【学习小结】

通过本周的学习，学生理解用户和用户组的概念，理解目录与文件的权限意义；掌握查看文件属性的命令、改变文件属性与权限的方法。

实验六 文件权限管理（二）

【实验目标】

掌握改变文件属性与权限的方法

理解目录与文件的权限意义

思政：同样管理学生自己的文件，学习保密性

【重点难点】

理解权限

查看和修改权限的方法

【知识点内容】

1.chown : 修改文件所有者

2.chgrp : 修改文件所属组

3.chmod : 修改文件的权限 (r,w,x)

4. chown: 修改文件所有者

用法: chown 账号名称 文件或目录

如果要将目录(文件夹)下所有子目录或文件同时更改所有者的话, 用 chown -R 账号名称
目录

5.chown: 修改文件所有者

用法: chown 账号名称 文件或目录

如果要将目录(文件夹)下所有子目录或文件同时更改所有者的话, 用 chown -R 账号名称
目录

6.chgrp: 修改所属组

用法: chgrp 账号名称 文件或目录

如果要将目录(文件夹)下所有子目录或文件同时更改所有者的话, 用 chgrp -R 账号名称
目录

7.chgrp: 修改所属组

用法: `chgrp` 账号名称 文件或目录

如果要将目录(文件夹)下所有子目录或文件同时更改所有者的话, 用 `chgrp -R` 账号名称 目录

8.chmod : 修改文件的权限 (r,w,x)

用法一: 数字方式

linux 对 r、w、x、三种权限以及无效权限 '-' 分别编码为 4, 2, 1, 0, 所以 `sources.list` 的权限编码为 664。算好后输入命令 `chmod 664` 文件名即可。注意要在 root 权限下修改(即加上 `sudo`)。

如果目标是目录(文件夹), 则输入 `chmod -R` 权限编码 文件名

用法二: 文字方式

取 user, group, others 的首字母, 加个等号, 再输入 `rx` 即可完成权限的给予, 比较直观。

文字方式还有一种特别用法: (a 表示 all, 全部身份的意思)

如果我想让 `sources.list` 文件的初始权限 "`rw-rw-r--`" 设置为不对任何人开放, 为了简单些, 我可以输入命令 "`chmod a-r sources.list`", 这样 `sources.list` 的权限变成了 "`-w--w----`", 也就是所有人都不可读(read), 也可以使用命令 "`chmod u-r,g-r,o-r sources.list`"

反之, 则是 `chmod a+r sources.list` 。

9.umask 是什么: 当我们登录系统之后创建一个文件是设定一个默认权限的, 那么这个权限是怎么来的呢? 这就是 `umask` 做的事情。`umask` 可用于设置用户创建文件或者目录的默认权限。

命令 "`umask -S`" 表示以字符的形式显示当前的掩码。

命令 "`umask -p`" 表示以数字的形式显示当前的掩码。

【实验内容】

- 1.如何修改`~/sources.list` 的属性, 使该文件只能被文件所有者和所在用户组查看, 其它用户无法查看? (提示: `chmod o-r ~/sources.list`)
- 2.如何修改`~/sources.list` 的属性, 使该文件能被所有用户查看? (提示: `chmod a+r ~/sources.list`)

3.查看当前用户的 `umask`。(提示: `umask [-S]`)

4.根据该 `umask`, 新建的目录和文件的权限分别是什么? 并新建一个目录和文件验证你的结论。

【学习小结】

通过本周的学习, 学生理解用户和用户组的概念, 理解目录与文件的权限意义; 掌握查看文件属性的命令、改变文件属性与权限的方法。

实验七 练习课

【实验目标】

一次性创建多个文件夹

以..开头或与..相关 (*)

find 查找超过 1M 的文件

思政：一次性所有便捷与自己鼠标控制的对比

【重点难点】

一次性

【实验内容】

1. 请用命令查出 ifconfig 命令程序的绝对路径。
- 2.(1)请在/下创建目录 abc。
(2)请在/下创建目录/liangjian/liyunlong/weiheshang/duanpeng
(3)请在目录 abc 下一次性创建 test1 test2 test3
(4)请在目录 abc 下一次性创建 1000 个目录，名字以 wenjian 开头，如 wenjian1,wenjian2,wenjian3...wenjian1000。
(5)请用绝对路径方式切换到/liangjian/liyunlong/weiheshang/duanpeng 目录下并用使用命令查看当前的路径，然后用上级目录名".."方式切换到 /liangjian/liyunlong 下
(6)请一次删除目录 abc 下一次创建的 1000 个目录（以 wenjian 开头）。
- 3.请用命令将/boot/目录下以 vmlinuz 开头的文件拷贝到/abc 下
- 4.将其中一个 vmlinuz 开头的文件改名为 kgc，另外一个剪切到/tmp 目录下。
- 5.查看/tmp/目录下以 vmlinuz 开头文件的详细状态信息。（提示：stat）
- 6.用命令查找/tmp 目录下以 vmlinuz 开头及大小超过 1M 的文件。（提示：find）

实验八 用户账户管理

【实验目标】

sudo 账户用法

创建账户

切换登录账户

删除账户

【重点难点】

多用户账户登录

【实验内容】

1. 在 Linux 系统中，用户账户及其相关信息（密码除外）均放在 `/etc/passwd` 文件中。用 `cat` 命令查看这个文件。
2. `/etc/passwd` 解释：该文件的每一行代表一个用户账户的资料，其中用冒号（:）分隔为 7 个域，分别是“用户名：加密口令：UID：GID：用户的描述信息：主目录：命令解释器”，其中某些域可以是空的。请从文件最后几行看出当前登录用户的 UID 是什么，主目录是哪个。
3. 创建一个新用户 `lisa`。
4. 查看 `lisa` 的主目录。
5. 给 `lisa` 设置密码 `123456`。
6. 把 `lisa` 的主目录设置为 `/home/lisa_new`。
7. 切换到 `tty2`，用 `lisa` 登录，看是否能用 `sudo shutdown -h now` 命令关机。
8. 切换回 `tty1`，新建用户 `lily`，其主目录为 `/home/lily`。
9. 给 `lily` 设置密码 `123456`，并切换到切换到 `tty3`，用 `lily` 登录
10. 切换回 `tty1`，用 `who` 命令查看当前有哪些登录用户。
11. 分别切换到 `tty2` 和 `tty3`，用 `exit` 命令退出登录。
12. 切换回 `tty1`，用 `who` 命令查看当前有哪些登录用户。

13. 删除用户 lisa。

14. 给 root 用户设置密码 12345678。

15. 切换到 tty2，用 root 登录，看是否能用 `shutdown -h now` 命令关机。

实验九 vi 文本编辑器

【实验目标】

掌握 vi 文本编辑器的使用方法。

【重点难点】

Vi 的模式切换

【实验内容】

- 保存并退出（“:wq” 或 “:x” 或 “按住 shift，连续按两次 z”）
- 保存但不退出（“:w”）
- 在没有修改这个文件的时候要退出，即只退出（“:q”）
- 只退出（“:q”）前提是没有对文件进行修改，如果进行了修改，则会报错，此时想强制退出，用（“:q!”），即不保存并退出
- 显示行号（“:set nu”）
- 取消显示行号（“:set nonu”）
- 移动光标到某个指定的行（先显示行号“:set nu”，指定到第 3 行，“先按数字 3，然后按大写 G”；移动光标到最后一行，“按大写 G”；移动到第一行，“连续按两下小写 g”）
- 同一行，快速跳转第一个字符（“按 0”）和最后一个字符（“按 shift+4”）
- 光标移动到第 4 行，向右移动 2 个字符。（4G, 2l）
- 替换：（如第 10 行到第 20 行之间的 a 替换成 b，“:10,20s/a/b/g，按回车”）
- 替换（如第 10 行最后一行之间的 c 替换成 a，“:10,\$s/c/a/g，按回车”）
- 替换+确认（如第 10 行最后一行之间的 c 替换成 a，“:10,\$s/c/a/gc，按回车”，“按 y 确定替换”，“按 n”不替换）
- 删除光标所在位置字符（小写“x”）
- 删除光标所在位置前一个字符（大写“X”）
- 删除光标所在一行（连续按两下小写“d”）
- 删除光标往下 3 行，包括光标所在行（先按数字 3，再连续按两下小写“d”）

- 复制粘贴（复制光标所在行，按连续按两下小写“y”，粘贴在光标所在位置的下一行，按小写“p”）
- 复制光标往下 3 行，包括光标所在行（先按数字 3，再连续按两下小写“y”）
- 粘贴在光标所在位置的上一行（按大写“P”）
- 撤销（按小写“u”，或按“ctrl+r”表示重复上一次操作）
- 重复前一个操作（按“.”）
- 所有以“:”输入的命令，都需要按回车
- 查找
- 1、进入 vi 中，先按下“ESC”跳转成编辑模式
- 2、输入斜杠“/”，这时屏幕会跳转到底部，输入栏出现“/”
- 3、输入你需要查找的关键字，回车
- 4、如果要继续查找关键字，输入 n
- 5、向前查找，输入 N（大写）

1. 将/etc/manpath.config 复制到目录/tmp 下。
2. 切换到/tmp 目录，使用 vi 开启本目录下的 manpath.config 文件。
3. 在 vi 中设定显示行号。
4. 光标移动到第 40 行，向右移动 22 个字符。
5. 光标移动到第一行，并且向下搜索“grep”这个字符串，请问它第一次出现在第几行？第三次呢？
6. 将 25 到 50 行之间的“man”改为“MAN”，并且一个一个挑选是否需要修改，如何下达指令？如果在挑选过程中一直按 y，结果会在最后一行出现改变了几个 man 呢？
7. 上一步修改完之后，要全部复原，有哪些方法？
8. 复制 66 到 71 这几行的内容，并且贴到最后一行之后。
9. 删除 95 到 99 这几行的内容。
10. 将这个文件另存成一个名为 manpath.test.config 的文件。
11. 在第一行新增一行，该行内容输入“I am a student...”。
12. 保存文件后退出。

【学习小结】

通过本周的学习，掌握 vi 文本编辑器的使用方法。

实验十 vim 编辑器

【实验目标】

掌握 vim 文本编辑器的使用方法。

【重点难点】

Vim 的功能

【知识要点】

vim 的功能（相对 vi）

- a) 颜色显示，支持许多程序语法
- b) 块选择
- c) 多文件编辑
- d) 多窗口功能

以上各功能的使用方法参考课本 P285-287。

【实验内容】

a) 颜色显示，支持许多程序语法

b) 块选择

vim 的快速选择操作：

快捷键	功能
v	字符选择，会将光标经过的地方着色选择
V	行选择，将光标经过的的行着色选择
[ctrl] + v	块选择，用方形的方式着色选择数据
y	将以上四种方式选择的数据复制
d	将着色选择的地方删除

c) 多文件编辑

d) 多窗口功能

vim -o

vim -O

13. 在用户主目录下，用 **vim** 分别新建两个文件，一个名为 **test1.txt**，内容为：

“Hello world!

I am a student.”

另一个名为 **test2.txt**，内容为：

“Test”

保存后退出。

14. 用 **vim** 同时打开 **test1.txt** 和 **test2.txt**。把 **test1.txt** 的第二行复制到 **test2.txt** 的最后一行后面。保存后退出。

15. 将 **/etc/manpath.config** 复制到用户主目录下。

16. 用 **vim** 打开 **~/manpath.config**，设置显示行号。

17. 新增一个窗口，打开文件 **test2.txt**。

18. 在 **manpath.config** 的第 29 到 41 行，三列的表格中，选取中间列，复制，粘贴到 **test2.txt** 最后一行后面。

19. 保存后结束 **test2.txt** 的窗口。

20. 退出 **vim**。

实验十一-1 GCC（GNU 编译器套件）

【实验目标】

掌握 gcc 的安装和使用方法。

【重点难点】

Gcc 的简单使用方法

【知识要点】

1. GCC（GNU 编译器套件）简介

GCC（GNU Compiler Collection，GNU 编译器套件），是由 GNU 开发的编程语言编译器。它是以 GPL 许可证所发行的自由软件，也是 GNU 计划的关键部分。GCC 原本作为 GNU 操作系统的官方编译器，现已被大多数类 Unix 操作系统（如 Linux、BSD、Mac OS X 等）采纳为标准的编译器。

GCC 原名为 GNU C 语言编译器（GNU C Compiler），因为它原本只能处理 C 语言。GCC 很快地扩展，变得可处理 C++。后来又扩展能够支持更多编程语言，如 Fortran、Pascal、Objective-C、Java、Ada、Go 以及各类处理器架构上的汇编语言等，所以改名 GNU 编译器套件（GNU Compiler Collection）。

2. GCC 安装（在 Ubuntu 中）

```
sudo apt-get install gcc
```

3. GCC 基本用法

GCC 最基本的用法是：`gcc [options] [filenames]`

其中 `options` 就是编译器所需要的参数，`filenames` 给出相关的文件名称。

`-c`，只编译，不链接成为可执行文件，编译器只是由输入的 `.c` 等源代码文件生成 `.o` 为后缀的目标文件，通常用于编译不包含主程序的子程序文件。

-o output_filename, 确定输出文件的名称为 output_filename, 同时这个名称不能和源文件同名。

-g, 产生符号调试工具所必要的符号资讯, 要想对源代码进行调试, 我们就必须加入这个选项。

-O, 对程序进行优化编译、链接, 采用这个选项, 整个源代码会在编译、链接过程中进行优化处理, 这样产生的可执行文件的执行效率可以提高, 但是, 编译、链接的速度就相应地要慢一些。

-O2, 比-O 更好的优化编译、链接, 当然整个编译、链接过程会更慢。

-I dirname, 将 dirname 所指出的目录加入到程序头文件目录列表中, 是在预编译过程中使用的参数。

4. gcc 所遵循的部分约定规则

.c 为后缀的文件, C 语言源代码文件;

.a 为后缀的文件, 是由目标文件构成的档案库文件;

.C, .cc 或.cxx 为后缀的文件, 是 C++源代码文件且必须要经过预处理;

.h 为后缀的文件, 是程序所包含的头文件;

.i 为后缀的文件, 是 C 源代码文件且不应该对其执行预处理;

.ii 为后缀的文件, 是 C++源代码文件且不应该对其执行预处理;

.m 为后缀的文件, 是 Objective-C 源代码文件;

.mm 为后缀的文件是 Objective-C++源代码文件;

.o 为后缀的文件, 是编译后的目标文件;

.s 为后缀的文件, 是汇编语言源代码文件;

.S 为后缀的文件, 是经过预编译的汇编语言源代码文件。

【实验内容】

1. 用 vim 编辑器新建一个 C 语言源代码文件 test.c, 内容如下:

```
#include <stdio.h>
```

```
main()
```

```
{  
    printf("Hello world!\n");  
}
```

保存后退出。

2. 把这个源文件直接编译成可执行程序：

```
gcc -o test test.c
```

3. 尝试执行 `test`（命令 `./test`），能执行吗？如果不行，可能是什么原因，如何解决呢？

实验十-2 文件压缩和打包（一）

【实验目标】

掌握文件压缩和打包的命令的使用方法。

思政：国产压缩软件与 Linux 压缩对比，还有差距。

【重点难点】

文件压缩和打包的命令

掌握 zip 和 unzip 命令

【实验内容】

zip 命令

是一个应用广泛的跨平台的压缩工具，压缩文件的后缀为 zip 文件。

基本用法

zip [参数] [压缩包名] [压缩的目录或者文件的路径]

常用参数：

-o 将压缩文件内的所有文件的最新变动时间设为压缩的时间

-q 安静模式，在压缩的时候不显示指令执行的过程

-r 递归压缩，将自定目录下的所有子文件以及文件一起处理

-x “文件列表”，压缩时排除文件列表中的文件

xxx.dat deflated (98%)

xxx.dat 文件有 98%的空间瘪下去了，即压缩率为 2%。

实例：

a.压缩目录，将/目录下的 bin 目录压缩为名为 bin.zip 压缩包：

```
zip -r bin.zip /bin
```

正常压缩，不加-q 选项，显示压缩的过程，加上-q 选项，安静模式输出，不显示压缩的过程：zip -rq bin.zip /bin

b. 压缩多个文件，将/目录下的 `anaconda3/LICENSE.txt` 文件和 `bin/grep` 文件 压缩为名为 `jihe.zip` 的压缩包： 因为压缩的全是文件，所以可以不用加上-r 选项

```
zip jihe.zip /anaconda3/LICENSE.txt /bin/grep
```

c. 压缩文件跟目录，将/下的 `anaconda3/LICENSE.txt` 文件和 `bin` 目录压缩成名为 `dajihe.zip` 的压缩包： `zip -qr dajihe.zip /anaconda3/LICENSE.txt /bin`

d.将/目录下的 `anaconda3` 目录压缩为 `anaconda31.zip` 压缩包，排除 `anaconda3` 目录下的 `LICENSE.txt` 文件：

```
zip -r anaconda31.zip /anaconda3 -x "./anaconda3 / LICENSE.txt"
```

e. 设置压缩级别为 9 和 1（9 最大，1 最小），1 表示最快压缩但体积大，9 表示最小但耗时最久

```
zip -r -9 xxx.zip xxx
```

f. 测试压缩包：

```
zip -T xxx.zip
```

g. 查看压缩包内容：

```
zip -sf xxx.zip
```

`unzip` 命令

将 `xxx.zip` 解压到当前目录：

```
unzip xxx.zip
```

使用安静模式，将文件解压到指定目录：

```
unzip -q xxx.zip -d ziptest
```

上述指定目录不存在，将会自动创建。如果你不想解压只想查看压缩包的内容你可以使用 -l 参数：

```
unzip -l xxx.zip
```

如解压中文压缩文件，使用 -O（英文字母，大写 o）参数指定编码类型：

```
unzip -O GBK 中文压缩文件.zip
```

【学习小结】

通过本周的学习，学生掌握文件压缩和打包的命令的使用方法。

实验十二 文件压缩和打包（二）

【实验目标】

掌握文件压缩和打包的命令的使用方法。

思政：国产压缩软件与 Linux 压缩对比，还有差距。

【重点难点】

文件压缩和打包的命令

掌握 tar 命令

【实验内容】

语法：tar 选项 文件或目录

l 选项 c，表示创建新的 tar 包

l 选项 v，表示显示详细 tar 执行过程

l 选项 f，指定压缩的文件名字

l 选项 t，不解压查看 tar 包的内容

l 选项 p，表示保持文件的原有属性

l 选项 j，通过 bzip2 命令压缩或解压

l 选项 z，通过 gzip 压缩或解压

l 选项 x，解压 tar 包

l 选项 C（大写），指定解压的目录路径

1. 切换到文字模式，以一般用户身份登录。
2. 把/etc/manpath.config 复制到用户主目录下，并查看文件大小。
3. 用 gzip 压缩~/manpath.config。（提示：gzip ~/manpath.config）
4. 查看原文件是否还存在？
5. 查看压缩后的文件 manpath.config.gz 的大小，和压缩前的文件大小对比。

6. 读取压缩文件 `manpath.config.gz` 的内容，用翻页查看的模式。（提示：`zcat manpath.config.gz | more`）
7. 解压文件 `manpath.config.gz`。（提示：`gzip -d manpath.config.gz`）
8. 查看压缩文件是否还在？
9. 用 `bzip2` 压缩 `~/manpath.config`，并保留原文件。（提示：`bzip2 -k ~/manpath.config`）
10. 查看原文件是否还存在？
11. 查看压缩后的文件 `manpath.config.bz2` 的大小，和压缩前的文件大小对比。
12. 读取压缩文件 `manpath.config.bz2` 的内容，用翻页查看的模式。（提示：`bzcat manpath.config.bz2 | more`）
13. 解压文件 `manpath.config.bz2`。（提示：`bzip2 -d manpath.config.bz2`）
14. 把整个 `/etc` 目录打包到用户主目录下，并通过 `bzip2` 进行压缩。（提示：`tar -jcv -f ~/etc.tar.bz2 /etc`）
15. 查看打包文件 `etc.tar.bz2` 的内容，使用翻页查看的方式。（提示：`tar -jtv -f ~/etc.tar.bz2 | more`）
16. 在用户主目录下新建一个目录 `testdir`。
17. 把打包文件 `etc.tar.bz2` 解开到 `~/testdir`。（提示：`tar -jxv -f ~/etc.tar.bz2 -C ~/testdir`）
18. 查看 `~/testdir` 的内容。
19. 把打包文件 `etc.tar.bz2` 中的 `etc/environment` 文件解压出来。（提示：`tar -jxv -f ~/etc.tar.bz2 etc/environment`）
20. 把以上各步骤中在用户主目录中新增的文件和目录删除。
21. 关机。（提示：`sudo shutdown -h now`）

【学习小结】

通过本周的学习，学生掌握文件压缩和打包的命令的使用方法。

实验十三 shell 编程（一）

【实验目标】

掌握 shell 的概念

【重点难点】

- 1.shell 以及 shell 脚本的概念
- 2.shell 脚本的创建及执行
- 3.echo 用法

【知识要点】

1.Shell 是系统的用户界面，提供了用户与内核进行交互操作的一种接口。它接收用户输入的命令并把它送入内核去执行。

2.Shell 脚本（shell script），是一种为 shell 编写的脚本程序。

Shell 是与 Linux 交互的基本工具，有两种执行命令的方式。一种是交互式，用户每输入一条命令，Shell 就解释执行一条。另一种是批处理（Batch），需要事先编写一个 Shell 脚本，其中包含若干条命令，让 Shell 一次将这些命令执行完，编写 Shell 脚本的过程就是 Shell 编程。
(封装/汇集命令)

3. Shell 环境

Shell 编程跟 JavaScript、php 编程一样，只要有一个能编写代码的文本编辑器和一个能解释执行的脚本解释器就可以了。

Linux 的 Shell 种类众多，常见的有：

Bourne Shell（/usr/bin/sh 或/bin/sh）

Bourne Again Shell（/bin/bash）

C Shell（/usr/bin/csh）

本教程关注的是 Bash，也就是 Bourne Again Shell，由于易用和免费，Bash 在日常工作中被广泛使用。同时，Bash 也是大多数 Linux 系统默认的 Shell。

在一般情况下，人们并不区分 Bourne Shell 和 Bourne Again Shell，所以，像 `#!/bin/sh`，它同样也可以改为 `#!/bin/bash`。

`#!` 告诉系统其后路径所指定的程序即是解释此脚本文件的 Shell 程序。

4.打开文本编辑器(可以使用 `vi/vim` 命令来创建文件)，新建一个文件 `test.sh`，扩展名为 `sh` (`sh` 代表 shell)

输入一些代码，第一行一般是这样：

```
#!/bin/bash
```

```
echo "Hello World !"
```

`#!` 是一个约定的标记，它告诉系统这个脚本需要什么解释器来执行，即使用哪一种 Shell。

`echo` 命令用于向窗口输出文本。

5.运行 shell

两种方法：

①作为可执行程序

将上面的代码保存为 `test.sh`，并 `cd` 到相应目录：

```
chmod 777 ./test.sh #使脚本具有执行权限
```

```
./test.sh #执行脚本
```

②作为解释器参数

这种运行方式是，直接运行解释器，其参数就是 shell 脚本的文件名，如：

```
/bin/sh test.sh
```

```
bash test.sh
```

● 13.4.echo

1、显示普通字符串

```
echo "It is a test"
```

这里的双引号完全可以省略：

```
echo It is a test
```

2、显示转义字符

```
echo "\It is a test\"
```

结果将是：

```
"It is a test"
```

3、显示变量

```
name="nihao"  
echo "${name}beijing"
```

4、显示转义字符

```
echo -e "OK!\n" # -e 开启转义  
echo "It is a test"
```

```
# \n换行符  
# \t制表符  
# \c不换行
```

实验十四 shell 编程（二）

【实验目标】

掌握 shell 的使用

【重点难点】

- 1.注释
- 2.shell 变量及使用
- 3.交互式 shell 脚本
- 4.数值计算
- 5.浮点型用法
- 6.数组

【知识要点】

- 1.注释

法一：<<EOF 和 EOF 之间（多行）

```
#!/bin/bash
```

```
<<EOF
```

这是一个打印“helloworld”的程序

```
EOF
```

```
echo "helloworld"
```

法二：每一行代码之前加#（单行）

```
#!/bin/bash
```

```
#EOF
```

```
#这是一个打印“helloworld”的程序
```

```
#EOF
```

```
echo "helloworld"
```

2.定义变量时，如

```
your_name="runoob.com"
```

```
your_name=1
```

```
your_name=1.4
```

注意，变量名和等号之间不能有空格，这可能和你熟悉的所有编程语言都不一样。同时，变量名的命名须遵循如下规则：

命名只能使用英文字母，数字和下划线，首个字符不能以数字开头。

中间不能有空格，可以使用下划线 _。

不能使用标点符号。

不能使用 `bash` 里的关键字（可用 `help` 命令查看保留关键字）。

3. 获取变量的值，只要在变量名前面加美元符号即可，如：：

```
your_name="qinjx"
```

```
echo $your_name
```

```
echo ${your_name}
```

```
ehco "my name is ${your_name}"
```

变量名外面的花括号是可选的，加不加都行，加花括号是为了帮助解释器识别变量的边界。

建议大家养成习惯，加上花括号。

已定义的变量，可以被重新定义，如：

```
your_name="tom"
```

```
echo $your_name
```

```
your_name="alibaba"
```

```
echo $your_name
```

这样写是合法的，但注意，第二次赋值的时候不能写 `$your_name="alibaba"`，使用变量的时候才加美元符（`$`）。

4. `read -p`

`p` 是 `prompt` 的缩写，即由这个开关项给出提示信息

`shell` 仅支持整型，数值计算使用 `$(表达式)`或 `$[]`，不要求空格，示例：

```
#!/bin/bash
```

```
read -p "please input operand and number: " operand number
```

```
echo "$operand + $number = $((($operand+$number))"
```

```
echo "$operand - $number = $((($operand - $number))"
```

```
echo "$operand * $number = $((($operand * $number))"
```

```
divided=$((($operand/$number))    #定义变量等号间不能有空格
```

```
echo "$operand / $number = $divided"
```

实验十五 shell 编程（三）

【实验目标】

掌握 shell 的条件判断、循环

【重点难点】

1. if 条件判断
2. 多命令执行
3. for 循环
4. while 循环

【知识要点】

1. ①if 语句语法格式：

```
if condition
```

```
then
```

```
    command1
```

```
    command2
```

```
    ...
```

```
    commandN
```

```
fi # 结束语句
```

- ②if else 语法格式：if condition

```
then
```

```
    command1
```

```
    command2
```

```
    ...
```

```
    commandN
```

```
else
```

```
command
```

```
fi # 结束语句
```

③if else-if else 语法格式:

```
if condition1
```

```
then
```

```
command1
```

```
elif condition2
```

```
then
```

```
command2
```

```
else
```

```
commandN
```

```
fi # 结束语句
```

2.多命令

使用逻辑与符号“&&”连接多条命令

```
命令1 && 命令2
```

使用逻辑或符号“||”连接多条命令

```
命令1 || 命令2
```

联合使用符号“&&”和“||”

```
命令1 && 命令2 || 命令3
```

```
命令1 || 命令2 && 命令3
```

使用括号()组合多条命令

3.for 循环

```
for var in item1 item2 ... itemN
```

```
do
```

```
command1
```

```
command2
```

```
...
```

```
commandN
```

```
done
```

4.while 循环

while condition

do

 command

done

实验十七 数据流重定向和管道命令

【实验内容】

1. 查看根目录/下的所有内容、权限和属性，并把它记录在文件/tmp/test1.txt 里。(ls -al /> /tmp/test1.txt)
2. 查看目录/tmp 下的所有内容、权限和属性，并把它接着记录在文件/tmp/test1.txt 里，注意不要清空/tmp/test1.txt 原先的内容。(ls -al /tmp >> ~/test1.txt)
3. 查看/tmp/test1.txt 的内容。
4. 读取来自键盘的输入 (hello world)，作为变量 a 的内容，要用什么命令？(read a)
5. 如果要用/tmp/test1.txt 的内容代替键盘输入，作为变量 b 的内容呢？(read b < /tmp/test1.txt)
6. 依次查看变量 a 和变量 b 的内容。(echo \$a; echo \$b)
7. 如果目录/tmp/abc 不存在，则把它创建出来。(ls /tmp/abc || mkdir /tmp/abc)
8. 如果目录/tmp/abc 存在，则用 touch 命令创建/tmp/abc/test2。(ls /tmp/abc && touch /tmp/abc/test2)
9. 查看 touch /tmp/abc/test2 的属性。
10. 翻页查看 ls -al /etc 的输出内容。(ls -al /etc | less)
11. 把/etc/manpath.config 中含 MANPATH 的几行显示出来，顺便输出行号。(grep -n 'MANPATH' /etc/manpath.config)
12. 把 /etc/manpath.config 中含 MANPATH 的几行排序。(grep 'MANPATH' /etc/manpath.config | sort)

实验十八 综合实验

【实验目标】

综合运用本课程的各知识点。

【实验内容】

1. 以普通用户（即非 root）身份登录。
2. 在用户主目录下，新建一个目录（我们称它为“目录 1”），名字为你的学号，并切换到该目录下。
3. 在当前目录下，用 vi 编辑器新建一个 shell 脚本，名字为“学号_script.sh”，内容要求如下：提示用户输入 first name 和 last name，最后把 full name 显示出来。
4. 修改“学号_script.sh”文件的权限，让文件所有者有执行权限，并运行该文件。
5. 在当前目录下新建一个名为 flash 的目录。插入 U 盘，把 U 盘挂载到该目录。用 ls 命令查看 U 盘有什么内容，并把结果输出到“目录 1”下的文件 flash.txt 中。卸载 U 盘。
6. 把“目录 1”打包到用户主目录下，并通过 bzip2 进行压缩，打包后的文件命名为“学号.tar.bz2”。
7. 新建一个用户，用户名为你的学号，注意在新建用户的时候强制创建用户主文件夹。
8. 切换到 tty2，用你的学号的用户登录，把刚才的“学号.tar.bz2”文件复制到当前目录下，并把它解开到当前目录下。