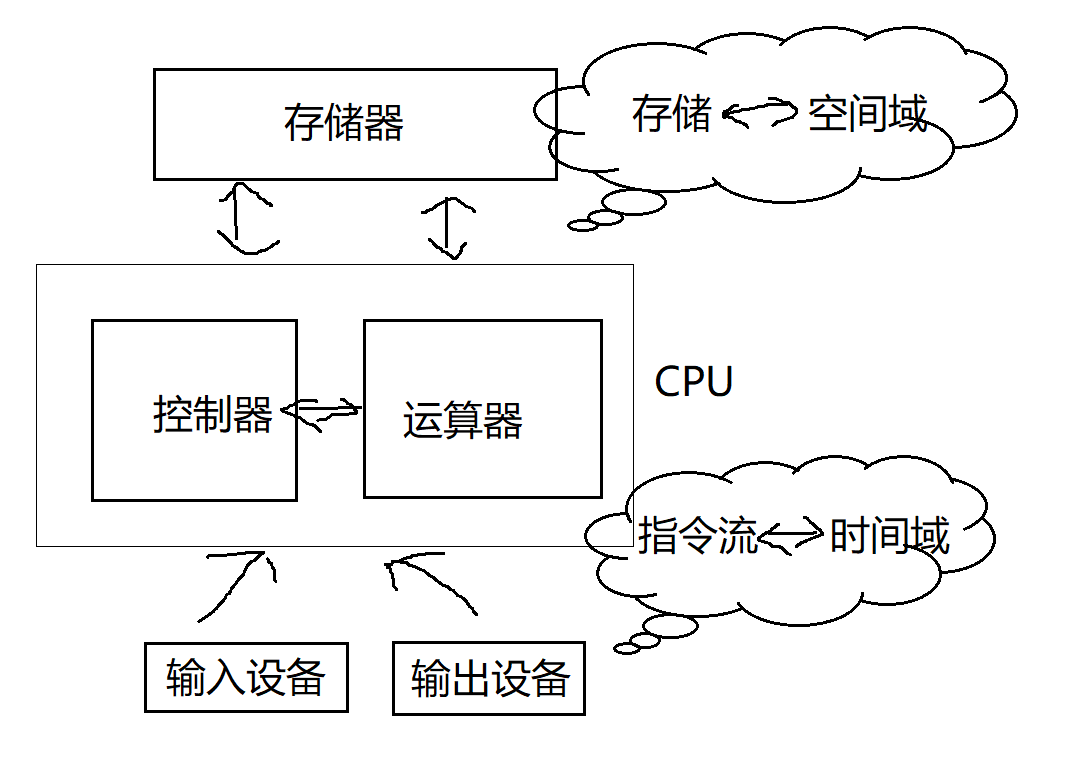
**第1章 计算机体系结构与操作系统**

1.1 计算科学与计算机系统

1. 冯诺依曼体系的结构要点：

计算机数制采用二进制，程序指令和数据统一存储，计算机应按照程序顺序执行。按照冯诺依曼结构设计的计算机由 **控制器，运算器，存储器和输入设备，输出设备**五部分组成。

1. 控制器和运算器 构成现代通用计算机系统的中央处理器 （CPU，central processing unit），即传统的单核CPU。
2. 在主存储器中实现存入要执行的程序和要处理的数据，运行时 CPU将自动并按顺序从主存储器中取出指令一条一条执行。冯诺依曼结构的计算机系统映射现实世界的时空观，存储体系对应空间域，指令顺序执行过程对应时间域。



1.2 计算机体系结构的基本概念

1.2.2 计算机系统的层次结构

1. 计算机系统是由软件和硬件组成的一个复杂的有机整体

2. 计算机系统的层次结构：

把计算机系统按功能可分为多级层次结构。第0层：数字逻辑层，第1层：微体系结构层，第2层：传统机器层，第3层：操作系统层，第4层：汇编语言层，第5层：高级语言层，第6层：应用程序层。

3. 计算机系统结构设计者所关心的是（机器语言或汇编语言程序员）所看到的计算机结构

1.2.3 指令集

1. 指令集是一个计算机系统支持的所有机器指令的集合，它常常被看作软硬件之间的分界面。
2. 计算机中最普通的指令格式是：操作码+操作数（地址）。用二进制数据来表示。
3. 根据操作数的数量不同，可以将指令格式分为：零地址指令，一地址指令、二地址指令、三地址指令和多地址指令。
4. 寻址方式按指令的寻址空间来分，可分为立即数寻址、寄存器寻址、主存寻址和堆栈寻址4种。
5. 指令的功能分类：

从指令的从操用计算机的指令系统中常见的指令类型包括：数据传输指令、算术运算指令、逻辑运算指令、程序控制指令、输入输出指令、处理器控制和调试指令。

1. 指令集因其系统性和复杂性，也被称为指令集体系结构ISA（instruction set architecture）
2. 按照指令系统复杂程度不同，ISA可分为CISC和RISC两大阵营。CISC是指复杂指令系统计算机complex instruction set computer; RISC是指精简指令系统计算机 reduced instruction set computer.
3. CISC结构追求的目标是强化指令功能，减少程序的指令条数，以达到提高性能的目的。RISC通过减少指令种类、规范指令格式和简化寻址方式等方法，方便处理器内部的并行处理，提高超大规模集成电路VLSI期间的使用效率，从而大幅度的提高处理器的性能。

1.2.4 存储系统

1. 现代通用计算机中存储系统的层次结构：

第一层寄存器组，第二层高速缓冲存储器（cache），第三层主存储器，第四层辅助存储器，第五层脱机存储器。存储介质越往上速度越快、价格越高，越往下，容量越大。

1. 程序访问的局部性包含时间局部性和空间局部性两方面。

时间局部性是指最近访问过的内容很可能会在短期内被再次访问。如程序中的循环，循环体的指令要被反复执行。空间局部性是指某个存储单位被访问，短时间内其附近的存储单元也会被访问。

1. “Cache-主存-辅存”是存储体系里最重要的三级层次，通常将其分为“Cache-主存”和“主存-辅存”两个两级存储系统来讨论。“主存-辅存”两级存储系统也称为虚拟存储系统。
2. “Cache-主存”层次的理论依据是程序局部性。
3. Cache与主存储器间最基本的地址映射方式有三种：全相联映射、直接映射和组相联映射。
4. 在采用全相联映射和组相联映射的系统中，当主存向Cache传送一个新块而Cache已满时，就需要使用替换算法将Cache中的某一块换出。常用替换算法有随机法（RAND），先进先出（FIFO）、最近最少使用法（LRU）。
5. Cache读写过程设计数据一致性的问题，一致性是指同一份数据在不同存储位置的多个拷贝应保持一致。Cache的一致性体现在连个方面，一是Cache和主存之间的数据一致性，二是各个处理器内核私有的Cache之间的数据一致性。
6. Cache与主存之间的数据一致性通过Cache特定的写操作方式来保证：写直达（write through），写回（write back）
7. 多核处理器中，处理器内核之间的缓存一致性通过缓存一致性协议类维护，MESI（Modified Exclusive Shared or Invalid）最为经典的缓存一致性协议
8. MESI协议将Cache Line 的状态分成 修改（Modify）、独占（Exclusive）、共享（Shared）和失效（Invalid）。
9. 虚拟存储系统的提出是为了满足应用程序对高速大容量主存的需求，扩大存储系统容量，速度慢。“主存-辅存”层次的存储管理技术称为虚拟存储管理技术。

1.3 微处理器体系结构

1. 微处理器，即中央处理器CPU 是计算机系统中最重要、最核心的部件
2. CPU 单核处理器基本逻辑结构，主要由控制器、运算器和存储单元三部分组成，这三部份由CPU内部总线相连。
3. 目前，通用处理器芯片基本上都采用了多核处理器（Chip Multi-Processor，CMP）架构。

按照计算内核的对等与否，CMP可分为同构多核和异构多核：计算内核相同、地位对等的称为同构多核；计算内核不同、地位不对等的称为异构多核，多采用“主处理核+协处理核”的设计。P19

1. CPU指令执行过程基本上包含5个阶段：取指令、指令译码、执行指令、访存取数、结果写回

1.4 操作系统的基本概念

1. 操作系统是通用计算机系统中不可或缺的系统软件
2. 计算机系统的资源按照其作用可大致分为四类：处理器、存储器、I/O设备和文件。
3. 对这四类资源的管理就是操作系统内核的主要功能。

1.4.2 操作系统的基本特征：

现代通用操作系统一般具有**并发、共享、虚拟和异步**这四个基本特征。并发是指两个或多个事件在同一时间间隔内发生。共享是指计算机系统中的软硬件资源可被多个并发执行的进程共同使用，而不是被一个进程所独占。虚拟是指把一个物理上的实体变为一个或多个逻辑上的对应物。异步又称随机性或不确定性，异步是指在相同的计算机环境和初始条件下，在同一数据集上运行的同一程序每次执行的顺序和所需时间都不一定相同。（避免混淆：并行是指两个或多个事件在同一时刻发生）

1.5 操作系统的基本功能和原理

1. 从资源管理的角度来看，操作系统具备五大基本功能：处理器管理、存储管理、设备管理、文件管理和作业管理。

1.5.1 处理器管理

1. 处理器管理的功能主要有两个：中断处理和进程管理。

-1.1 中断处理

1）.在处理器上运行的指令，根据权限可分为两种：

1.特权指令：只能被操作系统内核使用的指令，这些指令设计改变机器状态、修改寄存器内容、启动设备等敏感操作。

2.非特权指令：所有程序都能使同的指令。

2） 处理器的运行状态一般分为内核态和用户态，在内核态下能执行所有指令，在用户态下只能执行非特权指令。

3）用户态切换到内核态3种方式：系统调用，异常，外围设备的中断。3种方式都通过中断机制发生。

中断概念延伸拓展 https://blog.csdn.net/weixin\_36278982/article/details/113411238

中断是指CPU对系统发生的某个事件做出的一种反应，它使CPU暂停正在执行的程序，保留现场后自动执行相应的处理程序，处理该事件后，如被中断进程的优先级最高，则返回断点继续执行被“打断”的程序。

引起中断的事件或发出中断请求的来源称为中断源。中断源向CPU提出的处理请求称为中断请求。发生中断时，被打断程序的暂停点称为断点。

中断概念的另一个发展是访管(即访问管理）指令(或系统调用）的使用。

中断类型 按功能划分

这种分类法类似于IBM 360/370系统的分类法，所有中断源分为5类。

(1）机器故障中断。它是机器发生错误时产生的中断，用来反映硬件在执行过程中出现的故障，以便进入诊断程序，做出处理。例如，机器电路检验错、电源故障和内存读数错，等等。

(2）I/O中断。这是来自通道或各种外部设备的中断，用于反映通道或设备的工作情况。例如，打印机打印结束、磁盘传输完成、利用终端进行输入/输出等。

(3）外部中断。它是来自计算机系统外部装置的中断，用来反映外界对本系统的要求。例如，计时器到时溢出，操作员操纵控制台按钮，在多机系统中它机送来信号，等等。

(4）程序性中断。这是因错误地使用指令或数据而引起的中断，用于反映程序执行过程中发现的例外情况，例如，非法操作码，无效地址和运算溢出，等等。

(5）访管中断。由于执行“访问管理程序”指令而产生的中断，用来使CPU的状态从用户态转入管理态，由操作系统根据不同的编号引进不同的处理。这样，操作系统为用户态程序提供对系统资源使用请求的服务。

@ 在配置了通道的计算机系统中，用户程序需要输人/输出时，引起的中断是（访管

-1.2 进程管理

1）进程是计算机中的程序在某数据集合上的一次运行活动，是系统进行资源分配和调度的基本单位。

2）进程和程序是两个既有联系又有区别的概念。

进程是一个动态的概念，而程序则是一个静态的概念。进程具有并发特征，程序则没有；进程有生命期，它只在计算机运行期间才有可能存在，程序可以在外存上长期保存；进程与程序间并不总是一一对应，一个程序执行在不同的数据集上就称为不同的进程；进程与程序的组成不同，进程实体的组成包括程序、数据和控制块。

3）为了描述进程的动态变化过程，采用了一个与进程相关联的进程控制块（PCB）系统，根据它感知进程的存在。

4） 进程控制块 是进程存在的唯一标志。

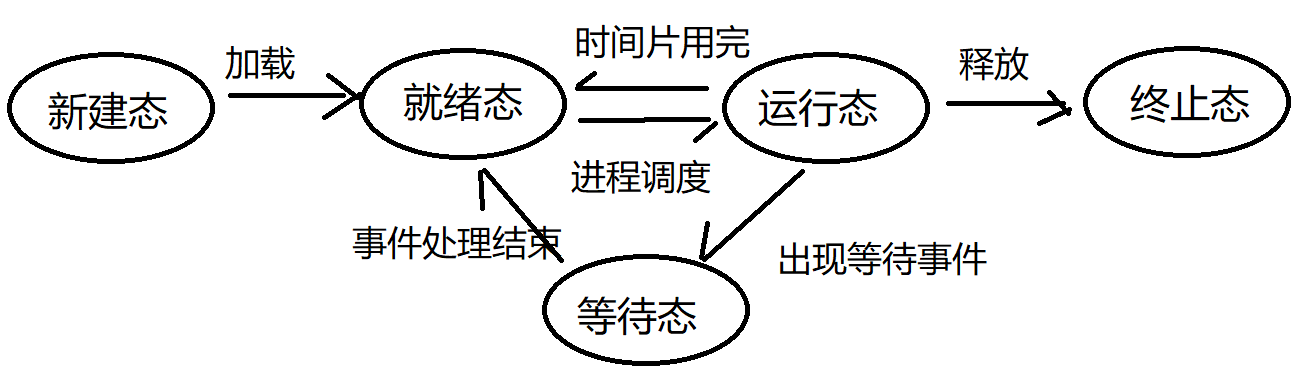
5）线程作为被处理器调度运行的最小单位。

线程被包含在进程之中，一条线程就是进程中一个单一顺序的控制流，在一个进程中可以并发多个线程，每条线程并行执行不同的任务。

6）进程的状态及转换

3种基本状态是：运行态(running），就绪态(ready），等待态(wait）：又称阻塞态(blocked）或睡眠态(sleep）。实际系统中增加两个进程状态：新建态(new），终止态(exit）

图片中的对应关系 要记住！



7）已经获得除 CPU 以外的所有运行所需资原的进程处于就堵状态

。。。。P30

1.5.2 存储管理

将用户源程序变为可在内存中执行的程序步骤：编译，链接，装入

存储器管理主要功能：内存空间的分配与回收，地址转换，内存空间的扩充和存储保护

1.5.3 设备管理

设备管理模块主要负责处理器内核与外部设备间的数据交互，包括把设备分配给某个请求使用该设备的进程，处理CPU和设备间I/O通信，响应设备的中断请求，维护和回收设备等。

1.5.4 文件管理

将负责文件管理的部分称为文件系统。这部分功能设计文件的逻辑组织和物流组织、目录结构和管理等

常用文件物理结构有顺序结构、链表结构、索引结构 。。。。P41 还有一些有时间再看

1.5.5 作业管理

作业时用户在一次计算过程中或一个事务处理中要求计算机系统所作工作的总和，也是用互相计算机系统提交任务的基本单位。

包括作业的输入和输出，以及作业的调度和控制。

一个作业从交给计算机系统到执行结束退出系统，一般经历提交、后备、执行和完成4个状态

**第2章 Linux系统概述**

1. Linus Torvalds 负责开发的Linux只是一个内核。内核指的是一个提供设备驱动、文件系统、进程管理、网络通信等功能的系统软件，内核并不是一套完整的操作系统，只是操作系统的核心。

2. Linux发行版本：Red Hat、Debian Linux、Fedora Core、CentOS、Ubuntu Linux、SuSE Linux、Gentoo Linux、Slackware

3. 2019年截止，世界超级计算机500强上运行的都是Linux。安卓系统是基于Linux的开源系统，是Google公司为移动终端打造的真正开放和完整的移动软件。亚马逊云阿里云谷歌云都部署在Linux系统上。人工智能和区块链的底层也都是Linux。

**第3章 Linux系统使用基础**

3.3 常用Linux命令

3.3.1. Linux命令的基本格式

“开始“菜单中选择应用程序-附件-终端，打开shell，看到命令提示符：

[root@localhost ~]#

root：当前的登陆用户

localhost：当前系统的简写主机名（完整名localhost.localdommain）

~：用户当前所在目录

#：命令提示符，如果是超级用户#，提示符是#；普通用户，提示符是$

家目录又称为主目录，用户在登陆后，要有一个初始登录位置，这个位置就成为用户的家，超级用户的家目录为/root，普通用户的家目录为/home/用户名。初始登录状态下，用户创建的文件都放入家目录。切换目录的命令是cd。

Linux命令的基本格式是

[root@localhost ~]# 命令 [选项] [参数]

Linux选项又分为短格式选项（-l），为英文简写和长格式选项（--all），为完整英文单词

命令的选项用于调整命令的执行方式，而命令的参数是这个命令的操作对象。

3.2.2. Linux简单命令

2.1 w和who命令

二者都可查看服务器上目前已登陆的用户信息，区别是w命令除了能知道目前已登陆的用户信息，还可知道每个用户执行任务的情况。

w命令基本格式：

[root@localhost ~]# w [选项] [用户名]

w命令后只跟用户名，表示只显示此用户的信息

-h 不显示输出信息的标题

-l 用详细格式输出

-s 用简洁格式输出

who命令基本格式：

[root@localhost ~]# who [选项] [file]

若无指定文件，则who命令默认通过/var/run/utmp文件来获取登陆用户信息

-a 列出所有信息，相当于所有选项

-b 列出系统最近启动时间

-l 列出所有可登录的终端信息

-m 仅列出当前终端的信息，相当于 who am I

-r

-s 仅显示名称线路和时间字段信息，是who命令的默认选项

-u

2.2 echo 命令

可将命令行中的参数显示到标准输出（屏幕）上，命令格式：

[root@localhost ~]# echo [选项] [输出内容]

-n 取消输出后行末的换行符号（内容输出后不换行）

-e 支持反斜线控制的字符转换

-e选项支持控制字符：

\\ 输出\本身

\a 输出警告音

\b 退格键，向左删除键

\c 取消输出行末的换行符，和-n 一样

\e Esc键

\f 换页符

\n 换行符

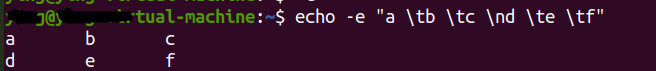
\r 回车键

\t 制表符，就是Tab键

\v 垂直制表符

\0nnn 按八进制ASCII码表输出字符

\xhh 按十六进制ASCII码表输出字符



2.3 date 命令

显示或设定系统的日期与时间

[root@localhost ~]# date -s 可以设定当前时间，只有root权限才能设置

2.4 passwd命令

密码配置指令，基本格式如下

[root@localhost ~]# passwd [选项] 用户名

普通用户仅能修改自己的密码，超级用户root可以修改自己和普通用户的密码。

普通用户修改自己的密码需要先输入自己的旧密码，超级用户不用。

3.4 软件包

Linux的软件包可细分为两种：源码包和二进制包。

Linux常用打包压缩格式“tar.gz”，因此源码包又被称为Tarball，Tarball是一款打包工具。

两大主流的二进制包管理系统：RPM包管理系统，DPKG包管理系统。

3.5 软件安装、运行与卸载

3.5.1 RPM包安装、卸载和升级

RPM包默认安装路径，最好不要手动指定安装路径

/etc/

/usr/bin/

/usr/lib/

/usr/share/doc/

/usr/share/man/

安装RPM包的命令格式：

[root@localhost ~]# rpm -ivh 包全名

涉及包全名的命令，一定要注意路径

此命令中各选项参数的含义为：

-i 安装（install）

-v 显示更详细的信息（verbose）

-h 打印，显示安装进度（hash）

命令还可以一次性安装多个软件包，仅需空格隔开，如下：

[root@localhost ~]# rpm -ivh a.rpm b.rpm c.rpm

安装完成后，可尝试启动：

[root@localhost ~]# service 服务名 start |stop |restart |status

各参数含义：启动服务|停止服务|重启|查看状态

RPM包的升级：

[root@localhost ~]# rpm -Uvh 包全名

-U 若该软件没安装过则直接安装，若已安装则升级至最新版本。

[root@localhost ~]# rpm -Fvh 包全名

-F 若该软件没有安装，则不会安装，必须安装较低版本才能升级。

RPM包的卸载要考虑包之间的依赖性，如先安装httpd软件包，后安装httpd的功能模块mod\_ssl包，那么卸载时，必须先卸载mod\_ssl包，然后卸载httpd，否则会报错。

卸载命令格式：

[root@localhost ~]# rpm -e 包名

-e 卸载，erase，卸载命令支持“-nocteps”选项，既可以不检测依赖性直接卸载，但不推荐

软件包的查询

[root@localhost ~]# rpm 选项 查询对象

-q 包名 ，表示查询是否安装，querry

-qa ，查询所有已安装软件包，可以用管道符查找出需要的内容，如

[root@localhost ~]# rpm -qa | grep httpd，采用这种方式可以找到含有包名的所有软件包。

-qi 包名， 查询软件包的详细信息，-i表示查询软件信息，information。

-qip 包全名，-p表示查询未安装的软件包，package。未安装的软件包需使用“绝对路径+包全名”的方式才能确定。

-ql 包名， 查询已安装软件包中包含的所有文件及各自安装位置。-l 表示列出软件包所有文件的安装目录。

-qlp 包全名，未安装软件包中包含的所有文件以及打算安装的路径。

还支持反向查询，查询文件属于哪个RPM软件包：

[root@localhost ~]# rpm -qf 系统文件名

-f 查询文件所属那个软件包，file。只有使用RPM安装的文件才能使用该命令，手动方式建立的文件无法使用

-qf /bin/ls

-qR

-qRp P93

第四章 shell脚本编程基础

4.1 shell 编程概述

1. shell本身是一个用C语言编写的程序，是连接 用户 和 UNIX/Linux内核 的桥梁。

2. #！/bin/bash , “#!” 说明该脚本使用哪种shell编写的，通常放在脚本第一行，从而调用相应的解释程序予以执行

3. # 是当行注释符，表示这是一行注释。注释行用来说明程序的功能、结构、算法和变量的作用等，增加程序的可读性。

4. Linux中，创建文件默认是不可执行的文件，需要修改文件的权限，为文件增加可执行权限，后在命令行模式下直接输入该文件的相对路径 ./文件名，即可运行

[root@localhost ~]# touch 文件名，创建一个新文件。

5. [root@localhost ~]# cat /etc/shell，可以看到shell的种类

6. 命令和用户的可行性脚本文件是否能够执行，与环境变量PATH有关

环境变量PATH的值的作用是 决定了shell将到哪些目录中寻找命令或可执行程序，当用户运行一个命令或可执行程序时，在这些目录下按照顺序依次搜寻

7. 脚本运行3种方式：脚本权限设置为可执行，在提示符下直接运行；将脚本名作为参数传递给shell程序；将脚本以输入重定向的方式传递给bash. P97

4.2 shell 基础

4.2.1 通配符

文件名的扩展称为通配（Globbing）

\* 匹配任意字符的0次或多次出现

？匹配任意单个字符

[ ] 匹配该字符组所限定的任何一个字符

[ ^ ] 或 [ ! ] 匹配不在该字符组中的任何一个字符

{ string1，string2，……} 匹配其中一个指定的字符串

4.2.2 引号

关于转义字符“\”，按下转义字符前，遇特殊字符，按特殊含义执行，按下转义字符后，按普通含义执行。

shell中引用字符有3种：单引号，双引号和倒引号。

1. 单引号引用一串字符，所有字符都是普通字符

2. 双引号引用一串字符，但保留 “$” ”\” 和倒引号 “ ` ” 的特殊含义。

$ 代表引用变量的值， 倒引号代表引用命令

3. 倒引号引用一串字符，被shell解释为命令行，shell会先执行该命令，并以其输出结果取代整个倒引号部分

4.2.3 输入/输出重定向符

输入/输出方向就是数据的流动方向。

1. 标准输入重定向

1.1 command<file 将文件作为命令的输入

[root@localhost ~]# wc -l<file 表示统计file文件中有多少行文本

1.2 command<<分界符 从标准输入中读入，直到遇见分界符才停止。使用特定的分界符作为命令输入的结束标志，而不使用默认的Ctril+D组合键。

2. 标准输出重定向

2.1 command>file 以覆盖的方式，把command的正确输出结果输出到file文件

[root@localhost ~]# cat file1 file2>file3 . 这种方式将命令的输出写到文件时，这个文件可以存在也可不存在，不存在就自动创建，存在则原文件内容将被覆盖。

2.1 command>>file 以追加的方式，把command的正确输出结果输出到file文件

3. 文件描述符

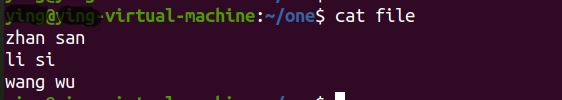
输出重定向的完整写法是fd>file, fd>>file，fd表示文件描述符，file descriptor。不写默认为1.

标准输入 stdin ， 0；

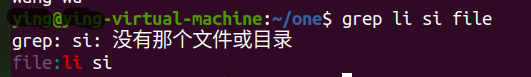
标准输出 stdout， 1；

错误输出 stderr， 2；

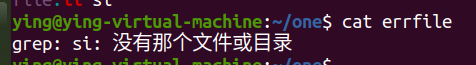
如：

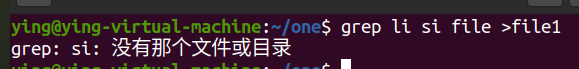


grep li si file: si 和file被视作两个文件名，查找li











若需要把正确和错误信息都保存在一个文件中，应这样写：

ying@ying-virtual-machine:~/one$ grep li si file >file1 2>& 1

2>& 之间不允许有空格，fd> 间不允许有空格

若既不想把结果保存到文件中，也不想显示到屏幕上，可以把结果重定向到/dev/null 文件中，



/dev/null视作Linux系统的垃圾箱，数据都会被丢弃，不能恢复。

4.2.4 命令执行操作符

1. 顺序执行

使用“；”连接多条命令，这些命令会依次执行。

2. 逻辑与

“&&”连接，这些命令之间就有逻辑关系。只有第一条命令正确执行了，&&连接的第二条命令才会执行。

3. 逻辑或

“||”连接多条命令，则只有前一条命令执行错误，后一条命令才能执行。

4.2.5 小括号和大括号

当用户启动shell时，这个shell是 \_\_ 交互\_\_ 的，当它执行完第一条命令后，接着显示另一个提示符，始终在前台等待用户。还有一种shell是 \_\_ 非交互\_\_ 的，当在交互shell中运行一个外部命令或者执行一个脚本时，交互shell会创建一个子shell，运行结束后子shell进程结束。

小括号（）和大括号{}都可以将若干命令括起来，组合在一起，逻辑上视为一条命令。

后台进程在运行过程中与用户无交互，无交互是指 \_\_不响应用户的输入和终端控制信号\_\_ ，在后台运行的命令一样会将结果输出到显示器上。

第5章 用户管理

5.1 用户与用户组管理

1. 用户名和ID的对应关系存储在/etc/passwd文件中。

2. 在默认情况下，管理员创建一个用户，就会在 /home 目录下创建一个用户主目录。

3. 存放用户密码信息的目录是 /etc

5.2 用户和用户组管理相关的文件

1. 用户基本信息：/etc/passwd文件的权限是 \_rw-r--r--\_ ,用户密码信息：/etc/shadow 文件的权限是 rw-------\_ 。

5.3 用户和用户组管理的命令

5.3.1 用户的添加

[root@localhost ~]# useradd [ 选项 ] 用户名

从创建的过程，系统读取 /etc/login.defs 和 /etc/default/useradd，根据这两个配置文件中定义的规则添加用户，在/etc/passwd、/etc/shadow、/etc/group、/etc/gshadow文件中添加用户数据；然后系统会自动在/etc/default/useradd文件设定的目录下建立用户主目录；最后复制/etc/skel目录中的所有文件到此主目录中。由此，一个新的用户创建完成了。

/etc/passwd 系统用户配置文件，存储了系统中所有用户的基本信息，所有用户都可对此文件执行读操作。

文件中每行用户信息划分为7个字段

用户名：密码：UID：GID：描述性信息：主目录：默认shell

用户名字符串，密码x表示设有密码，真正密码保存在/etc/shadow；

UID取值范围0-65535，某些情况下允许用户存在多个拥有不同用户名但UID相同的用户；GID初始组只能一个，附加组可以多个。UID在1-499供系统程序使用的用户，普通用户UID从500开始。

主目录，用户登录后有操作权限的访问目录，root超级管理员账户的主目录为/root，普通用户主目录为/home/UID，即在/home下建立和用户名相同的目录作为主目录。

默认shell：/bin/bash

5.3.2 密码配置命令

passwd [选项] 用户名

普通用户只能使用passwd命令修改 \_\_\_ 自己的 \_\_\_ 密码，而不能修改 \_\_\_ 其他用户的 \_\_ 的密码。超级用户root既可以使用passwd命令修改 \_\_ 自己的 \_\_ 密码，也可以修改 \_\_ 其他用户的 \_\_ 的密码

5.3.3 修改用户信息命令

[root@localhost ~]# usermod [ 选项 ] 用户名

-L 临时锁定用户Lock

-U 解锁用户 Unlock

5.3.4 修改用户密码状态命令

[root@localhost ~]# change [ 选项 ] 用户名

5.3.5 删除用户命令

[root@localhost ~]# userdel -r 用户名，只用root用户才能用

-r 表示在删除用户的同时删除用户的家目录。

用户个人文件：主目录默认位于 /home/用户名；邮箱位于 /var/spool/mail/用户名。

5.3.7 用户间切换命令

[root@localhost ~]# su [ 选项 ] 用户名

-：当前用户不仅切换为指定用户的身份同时所用的工作环境也切换（包括PATH变量、MAIL变量等），使用-选项可省略用户名，默认切换为root用户。环境变量是用来定义操作系统环境的，因此若系统环境没有随用户身份切换，很多命令无法正确执行。

没有-的话，则是临时切换。

使用su命令时,有-和没有-是完全不同的，- 选项表示 \_ 在切换用户身份的同时，将当前使用的环境变量也切换成指定用户的

5.3.8 用户组管理命令

添加用户组

[root@localhost ~]# groupadd [ 选项 ] 组名

修改用户组相关信息

[root@localhost ~]# groupmod [ 选项 ] 组名

删除用户组（群组）

[root@localhost ~]# groupdel 组名，仅适用于删除那些不是任何用户初始组的组

为了避免系统管理员root太忙，无法及时管理组，可以给组设置组管理员，代替root完成将用户加入或移出组的操作。

[root@localhost ~]# gpasswd [ 选项 ] 组名

选项为空时，表示给组设置密码，仅root用户可用

-a user : 将user用户加入组

-d user: 将user用户从组移除

第6章 文件管理

6.1 文件系统

1. 文件：文件系统中存储数据的一个命名对象，是用户中最小的逻辑存储单元。

2. Linux操作系统中，所有的文件和目录都被组织成从一个 根节点“/” 开始的倒置的树状结构。文件系统 \_\_ 按名\_ 访问文件。

6.1.2 文件结构和类型

1. 文件是Linux系统处理信息的基本单位，所有软件都组织成文件。所有文件都由两部分组成，数据项和文件控制块。文件控制块被称为索引节点。

2. 根据文件数据项是否具有结构，可将文件分为两大类：无结构文件（即字符流文件）和有结构文件（如目录文件）。但通常情况下，Linux文件类型被划分为普通文件、目录文件、链接文件及设备文件。

3. 说明目录或文件名位置的方法有两种，分别使用 \_ 绝对路径\_\_ 和 \_\_ 相对路径\_ 。前者指的是从根目录(/)开始写起的文件或目录名称，而后者则指的是相对于当前路径的文件或目录名称。

6.2 文件与目录操作

6.2.1 文件操作命令

1. concatenate 连接、连续

[root@localhost ~]# cat [ 选项 ] 文件名 ， 显示文件内容

-n 对输出的所有行进行编号

-b 只对非空行进行编号

或者

[root@localhost ~]# cat 文件1 文件2 > 文件3，连接合并文件，file3 原内容会被覆盖，若file3不存在会自动创建。

2. 分页显示文本文件内容

[root@localhost ~]# more [ 选项 ] 文件名 ，

3. 显式指定文件的前若干行内容

[root@localhost ~]# head [ 选项 ] 文件名

-n k : k表示行数，用来显示文件前k行的内容。

不设置显示的具体行数，则默认显示10行的文本数据。4

4. 查看文件末尾的数据

[root@localhost ~]# tail [ 选项 ] 文件名

-n k : k表示行数，用来显示文件后k行的内容。

-k: 同上

-f : 用来监听文件的新增内容，若想中止按Ctrl+C键

5. 创建文件，修改文件时间参数

[root@localhost ~]# touch [ 选项 ] 文件名

-a：只修改文件访问时间 P171

1. grep命令

global regular expressions print

在一个或多个文件中搜索某一特定的字符模式（即正则表达式），此模式可以是单一的字符、字符串、单词或句子。正则表达式是描述一组字符串的一个模式，可以是一些纯文本文字，也可以是用来产生模式的一些特殊字符。正则表达式中包含一些特殊字符，与通配符含义不同。

c\* 将匹配0个或多个字符c（c为任一字符）

.： 将匹配任何一个字符，且只能是一个字符。

[xyz]：匹配方括号中任意一个字符。

[^xyz]：匹配出方括号中字符外的所有字符。

^：行首定位符。

$：行尾定位符。

[root@localhost ~]# grep [ 选项 ] 模式 文件名

-n 在每一行的最前面列出行号

1. sed命令

主要用来自动编辑一个或多个文件，简化对文件的反复操作，编写转换程序等。

[root@localhost ~]# sed [ 选项 ] sed命令 文件名

选项：

-n：只显示匹配处理的行

-e：执行多个编辑命令时使用（一般用“；”代替）

-i：直接在文件中进行修改，而不是输出到显示器。

-f：从脚本文件中读取内容并执行（文件中的编辑命令每行一个，不用“；”隔开）

sed常用编辑命令：

p：打印匹配行print

d：删除指定行delete

a：在匹配行后面追加append

i：在匹配行前面插入insert

c：整行替换

r：读取文件的内容

w：将文本写入文件

s：字符串替换（匹配正则表达式）substitution

1. awk命令

更适合格式化文本，对文本进行较复杂的格式处理

[root@localhost ~]# awk [ 选项 ] ‘匹配规则|执行命令|’ 文件名

awk的主要特性之一是其较强的处理文本文件中数据的能力，它会自动给一行中的每个数据元素分配一个变量。默认情况下，将如下变量分配给它在文本行中发现的数据字段：

$0：代表整个文本行

$1：代表文本行中的第1个数据字段

$2：代表文本行中的第2个数据字段

$n：代表文本行中的第n个数据字段

在awk中默认字段分隔符是任意的空白字符，如空格或制表符。读取一行文本时，会用预定于的字段分隔符划分每个数据字段。若要采用其他字段分隔符的文本，可以用-F选项手动指定。

[root@localhost ~]# awk ‘{print $1}‘ file

[root@localhost ~]# awk ‘BEGIN {print “The File Contents：“}> {print $0} END {print”End of File”}‘ file

1. cp命令

复制文件和目录，借助某些选项，还可复制整个目录

[root@localhost ~]# cp [选项] 源文件 目标文件

-a：相当于-d、-p、-r选项的集合

-d：若源文件为软链接（对硬链接无效），则复制的目标文件也为软链接

-i：询问，若目标文件已经存在，则会询问是否覆盖

-l：把目标文件建立为源文件的硬链接文件，而不是复制源文件

-s：把目标文件建立为源文件的软链接文件，而不是复制源文件

-p：复制后目标文件保留原文件的属性

-r：递归复制，用于复制目录

目标表文件必须是目录。

把源文件不改名复制到/tmp/目录下：

[root@localhost ~]# cp file /tmp/

若需改名复制，如下：

[root@localhost ~]# cp file /tmp/text

提示是否覆盖：

[root@localhost ~]# cp -i file /tmp/text

1. rm命令

永久性删除文件系统

[root@localhost ~]# rm [选项] 文件或目录

-f：强制删除force

-i：删除前询问

-r：递归删除，主要用于删除目录，可删除指定目录及包含的所有内容。

1. mv命令

可在不同目录间移动文件或目录，也可对文件和目录进行重命名。

[root@localhost ~]# mv [选项] 源文件 目标文件

-f：强制覆盖

-i：交互移动，若目标文件已经存在，询问用户是否覆盖（默认选项）

-n：若目标文件已经存在，则不会覆盖移动，而且不询问用户。

-v：显示文件或目录的移动过程。

-u：若目标文件已经存在，两者相比，源文件更新，则会对目标文件进行升级。

1. sort命令

排序命令。ASCII码值进行比较，升序输出。

-r 反向排序。

1. wc命令 统计

-l：只统计行数

-w：只统计单词数

-m：只统计字符数

6.2.2 目录操作命令

1. mkdir命令 make directories

-m 用于手动配置所创建目录的权限，而不再使用默认权限。

-p 递归创建所有目录。

自定义目录权限

2. rmdir

-p 递归删除空目录

3. cd命令

change directory 切换工作目录。

特殊符号

~ ：代表当前登录用户的主目录

- ：代表上次所在目录

**.** ：代表当前目录

**..** ：代表上级目录

cd ~ 命令的含义是 回到用户的（主）家目录

4.pwd命令

print working directory 显示当前所在的目录。

5.ls 命令

显示当前目录下的内容。

不使用任何选项时，默认只会显示非隐藏文件的名称，并以文件名进行排序。

-a 显示全部的文件，包括隐藏文件。（开头为”.”的文件）

6.ln 命令

ln [选项] 源文件 目标文件

-s：创建软链接文件，不加则创建硬链接文件

硬链接：不是是修改源文件还是硬链接，另一个文件数据都会发生改变；硬链接不能跨文件系统（分区）建立，不能链接目录。

软链接：相当于快捷方式，可跨分区，可链接目录。

使用ln命令生成了一个指向文件old的符号链接new,如果将文件old删除，不可能再访问文件中的数据

6.3 访问权限管理

可以使用ls命令查看文件和目录的权限信息，ls -al

6.3.2 权限位

文件权限有3种，对文件的读，写和执行权限。每行第一列为权限，一共10位，第1位表示文件的具体类型。

6.3.2 修改权限位的命令

1. chmod 命令

权限与数字对应关系： r -4，w -2，x -1

使用数字修改文件权限的命令格式：

chmod 权限值 文件名

若加选项 -R 表示连同子目录中的所有文件，也都修改为设定的权限。

文件的基本权限就是3种用户身份，所有者u，所属组g和其他人o 。a代表全部的身份。

chmod go+/-/=rwx

2. umask 命令

使用umask默认权限来给所有新建的文件和目录赋予初始权限。

直接通过umask命令得知默认权限的值！

默认是0022，第1个数代表的是文件所具有的特殊权限，后3位才是要用到的umask值，转变为字母形式是----w--w-。

文件和目录的真正初始权限可通过以下计算得到：

文件（或目录）的初始权限=文件（或目录）的最大默认权限-umask权限

Linux系统中，文件的最大默认权限为666，即rw-rw-rw-。

目录的最大默认权限是777，即rwxrwxrwx。

注：用字母形式做减法！最大默认权限中 和umask权限公共的部分，通过减法运算被遮盖掉，剩下的最大默认权限，才是最终赋予文件目录的初始权限。如666-033=644

3. chown 命令

用于修改文件（或目录）的所有者。基本格式：

chown 所有者 文件或目录

-R 表示连同子目录中的所有文件都更改。

若要同时更改所有者和所属组

chown 所有者：所属组 文件或目录

4. chgrp 命令

用于修改文件（或目录）的所属组。

chgrp 所属组 文件名（目录名）

第7章 进程管理

有3个作用：

1.判断服务器的健康状态；2.查看系统中所有进程；3.杀死进程

7.2 进程监控

1. ps 命令

缺乏时效性最常用进程监控命令，可查看系统中所有运行进程的详细信息。

-a：显示一个终端的所有进程（除会话引线外）

-u：显示进程的归属用户及内存的使用情况

-x : 显示没有控制终端的进程

-l ：以长格式显示更加详细的信息

-e：显示所有进程

建议记忆几个固定选项：

ps-aux ：可以查看系统中所有的进程

ps-le ： 可以查看系统中所有的进程，而且还能看到进程的父进程PID和进程优先级。

ps-l ：只能看到当前shell产生的进程。

僵尸进程：一般是由于进程非正常停止或程序编写错误，因此子进程先于父进程结束，而父进程又没有正确地回收子进程，从而子进程一直存在内存中。

2. pstree 命令

以树形结构显示程序和进程之间的关系

3. top 命令

可以动态地持续监听进程的运行状态。还提供了一个交互界面，用户可根据需要，人性化地定制自己的输出，进而更清楚地了解进程的运行状态。

top的输出内容是动态的，默认每隔3秒刷新一次。

缓存是在读取硬盘中数据时，把最常用的数据保存在 内存 的缓存区中，再次读取该数据时，就不去 硬盘中 读取了，而在 缓存 中读取；缓冲是在硬盘写入数据时，先把数据放入 缓冲区，然后再一起向硬盘写入，把 分散的写操作 集中进行，减少磁盘碎片和硬盘的反复寻道，从而提高系统性能。

设置缓存是为了加快数据从硬盘读取的速度，而设置缓冲是为了加快数据写入硬盘的速度。

4. lsof 命令

ps命令可以查询到系统中所有的进程，可以进一步知道这个进程到底在调用哪些文件。list opened files

-c字符串：只列出以字符串开头的进程打开的文件。

-p PID：列出PID所指的进程打开的文件。

7.3 结束进程

进程的管理主要是指进程的关闭与重启。进程的关闭要依赖进程的信号。系统中可以识别的信号较多，可以使用命令“kill-l”来查询。

常见的进程信号

1：sighup 是进程立即关闭，重新读取配置文件之后重启

9：sigkill 立即结束程序的运行；本信号不能被阻塞、处理和忽略；一般用于强制终止进程

15：sigterm 正常结束进程的信号，kill命令的默认信号；如果进程已经发生了问题，那么这个信号是无法正常结束进程的，这是可以尝试sigkill信号9.

在使用kill 命令来终止一个很消耗资源的进程的时候,最合适的信号值是9。

7.4 进程优先级

在Linux系统中，有2个参数表示进程优先级：PRI 和 NI。priority，nice。数值越小代表该进程越优先被CPU处理。不过，PRI值是由内核动态调整的，用户不能直接修改。只能通过修改 NI 值来影响 PRI 值，间接地调整进程优先级。

PRI和NI的关系：PRI（最终值）=PRI（原始值）+NI。

NI值越小，进程的PRI就会越小，该进程就越优先被CPU处理。

修改NI值时有几个注意事项：

* 1. NI范围是-20~19.
  2. 普通用户调整NI值的范围是0~19，而且只能调整自己的进程。
  3. 普通用户只能调高NI值，而不能调低。
  4. 只有root用户才能设定进程NI为负值，而且可以调整任何用户的进程。

默认情况下，进程的NI值为0。进程的NI值可以通过nice命令和renice命令修改，进而调整进程的运行顺序。

nice命令可以给要启动的进程赋予NI值，但不能修改已运行进程的NI值。

[root@localhost ~]# nice [ -n NI值] 命令

renice命令可以在进程运行时修改其NI值，从而调整优先级。

[root@localhost ~]# renice [优先级] PID

第8章 存储管理

8.1 存储设备的查看

从存储数据的介质上来区分，硬盘可分为机械硬盘（hard disk drive，HDD）和固态硬盘（solid state disk，SSD），机械硬盘采用磁性碟片存储数据，固态硬盘通过闪存颗粒存储。

机械硬盘的逻辑结构主要分为磁道、扇区和柱面。以下几种，IDE（Integrated Drive Electronics，电子集成驱动器）硬盘接口；SATA（Serial ATA，串口）接口；SCSI（Small Computer System Interface，小型计算机系统接口）。

扇区是磁盘的最小存储单位。

固态硬盘和机械硬盘最大的区别就是不再采用盘片存储数据，而采用存储芯片存储数据。

固态硬盘低能耗、无噪声、抗震动、低散热、体积小和速度快的优势；价格更高，使用寿命有限。

* + 1. 硬盘识别
    2. 存储设备的挂载

挂载指的是将硬件设备的 文件系统 和Linux系统中的文件系统，通过 指定目录（作为挂载点） 进行关联。而将文件系统挂载到Linux系统上，需要使用mount挂载命令。

为了能够把新建立的文件系统装载到系统目录中,还需要指定该文件系统在整个目录结构中的位置,称为 挂载点。

所有的硬件设备必须 \_\_ 挂载\_\_ 之后才能使用,有些硬件设备(比如硬盘)在每次系统启动时会， \_ 自动挂载\_ 。而有些(比如U盘、光盘)则需要 \_ 手动挂载\_。

1. mount 命令

[root@localhost ~]# mount -l

单纯使用mount命令，会显示出系统中已挂载的设备新宁县。使用-l，会额外显示出卷标名称。

[root@localhost ~]# mount -a

自动检查/etc/fstab文件中有无因疏漏而未被挂载的设备文件，若有则进行自动挂载操作。/etc/fstab文件为自动挂载文件，系统开机时会主动读取文件内容，根据文件配置，系统会自动挂载指定设备。

[root@localhost ~]# mount [-t系统类型] [-L卷标名] [-o特殊选项] [-n]设备文件名 挂载点

-t系统类型：指定欲挂载的文件系统类型。常见的支持类型有EXT2、EXT3、EXT4、iso9660、vfat、reiserfs等。

-L卷标名：除了使用设备文件名（如/dev/hdc6）之外，还可以利用文件系统的卷标名称进行挂载。

-n：默认情况下，系统会将实际挂载的情况实时写入/etc/mtab文件中。但在某些场景下（如单人维护模式），为避免出现问题，会刻意不写入，此时就需要使用这个选项。

-o特殊选项：可以指定挂载的额外选项，如读/写权限，同步/异步等。

卸载使用 umount 命令

1. 挂载光盘

建立挂载点目录：

[root@localhost ~]#mkdir /mnt/cdrom/

挂载命令

[root@localhost ~]# mount /dev/cdrom /mnt/cdrom/S

1. 挂载U盘

光盘的设备文件名是固定的（/dev/sr0 或/dev/cdrom），U盘的设备文件名是在插入U盘后系统自动分配的。

先以root用户身份登录，通过使用fdisk命令查看到U盘的设备文件名：

[root@localhost ~]# fdisk -l

假若U盘是sda1，接下来要创建挂载点，在目录/mnt下建立文件夹/usb：

[root@localhost ~]# mkdir /mnt/usb

然后挂载：

[root@localhost ~]# mount -t vfat /dev/sda1 /mnt/usb/

1. 自动挂载

若想实现开机自动挂载某个硬件设备，使用root身份在/etc/fstab文件中添加此设备即可。

8.2 分区管理

Linux中分区命令fdisk和parted。fdisk较常用，但不支持大于2TB的分区。大于2TB使用parted命令，parted也能分配较小的分区。

8.2.1 fdisk 命令

查询本机可以识别的硬盘和分区：

[root@localhost ~]# fdisk -l

给硬盘分区：

[root@localhost ~]# fdisk 设备文件名

不要在当前的硬盘上尝试使用fdisk，这会完整删除整个系统。

建立主分区的过程：fdisk硬盘名—n（新建）—p（建立主分区）—l（指定分区号）—按Enter键（默认从柱面1开始建立分区）—+5G（指定分区大小）。当然，分区还没有格式化和挂载，还不能使用。

主分区和扩展分区加起来最多只能建立4个，扩展分区最多只能建立1个。

所有的分区在建立过程中如果不保存就退出，是不会生效的。使用q命令不保存退出，使用w命令，保存退出。若不想重新启动系统可以使用partprobe命令，作用是让系统内核重新读取分区表信息。

8.2.2 parted 命令

parted命令是可以在命令行直接分区和格式化的，parted交互模式才是更加常用的命令方式，进入交互模式命令：

[root@localhost ~]# parted 硬盘设备文件名

在parted等待输入交互命令的位置输入help，可看到在交互模式下支持的所有命令。

分区完成后，需要进行格式化。使用parted交互命令格式化，则只能格式化成ext2文件系统。需要使用mkfs命令（不是parted交互命令中的mkfs，而是系统命令mkfs）。parted命令还可以调整分区的大小。

parted所有的操作都是立即生效的！！

* 1. 磁盘管理工具
     1. df 命令

用于显示Linux系统中个文件系统的硬盘使用情况，包括文件系统所在硬盘分区的总容量、已使用的容量、剩余容量等。与整个文件系统有关的数据都保存在超级块（Super Block）.

[root@localhost ~]# df [选项] [目录或文件名]

-a：显示所有文件系统信息，包括Linux系统特有的/proc、/sysfs等文件系统。

-h：使用人们习惯的MB或GB等单位自行显示容量。

-i：不显示硬盘容量，而显示inode的数量。

无选项默认将系统内文件系统信息KB为单位显示出来。

df命令后添加目录名，会自动分析该目录所在的分区，并将所在分区的有关信息显示出来。

8.3.2 du 命令

统计目录或文件所占磁盘空间容量的命令。

[root@localhost ~]# du [选项] [目录或文件名]

-a：显示所有子目录和子文件的磁盘占用量。

-h：使用习惯单位显示磁盘占用量

-s：统计总磁盘占用量，而不列出子目录和子文件的从磁盘占用量。

不带选项为统计当前目录的总磁盘占用量，同时会统计当前目录下所有子目录的磁盘占用量，不统计子文件，默认单位KB。

当删除一个文件时，不是马上就在文件系统中消失，当所有程序都不用该文件时，才会根据操作系统的计算规则释放已经删除的文件空间。

df命令能看到已经删除的文件，计算大小的时候，把这一部分的空间也加上了，因此更精确。

8.3.3 fsck 命令

检查文件系统并尝试修复出现的错误。

[root@localhost ~]# fsck [选项] 分区设备文件名

-a：自动修复文件系统，没有任何提示信息。

-r：交互式修复模式，修改文件前会进行询问，让用户确认并决定处理方式。

fsck命令修复磁盘分区以后，若发现分区中有文件丢失，可以到对应的“lost+found”目录中去查找，但由于无法通过文件名称分辨各个文件，可利用file命令查看文件系统类型，进而判断哪个是用户需要的文件。

8.4 磁盘阵列

第9章 设备管理

9.1 设备文件

Linux中，用户是通过文件系统与设备连接的。每个设备都是在其驱动程序控制下运行的，驱动程序则与/dev目录下的特殊文件联系在一起，在选择设备或者操作设备时都以这个文件的名称来代表这一设备。

Linux系统采用\_\_\_设备文件\_\_\_ 统一管理硬件设备,从而将硬件设备的特性及管理细节对用户隐藏，实现用户程序与设备无关性。从用户的角度来看；对设备的使用和一般文件的使用一样。

设备的命名规则：在/dev目录下，每一个驱动器以一个单独的设备文件来表现；设备的文件名一般由两部分组成，第一部分是 主设备号 ，第二部分是 次设备号 。主设备号代表设备的类型，如hd表示IDE硬盘，sd表示SCSI硬盘，tty表示终端设备等；次设备号代表同类设备中的序号，如IDE硬盘一般可以有4个，即/dev/hda、/dev/hdb、/dev/hdc、/dev/hdd，a代表第一个设备，hda1中的”1”代表had的第一个硬盘主分区，以此类推。SCSI硬盘就是/dev/sda、/dev/sdb等。fd是软盘，设备名称是/dev/fd0。Linux系统中，磁带机的设备名是/dev/st\*和/dev/nst\*，其中nst\*是指操作完成后不自动回卷磁带，st\*自动回卷磁带，常用nst\*。主机SCSI总线上找到的第一个磁带机是nst0或st0，第二个是nst1或st1，依此类推。磁带机的SCSI号越小，其设备名也就越靠前。

一台主机上可以有多块硬盘，系统采用a~p来代表16块不同的硬盘，硬盘的分区编号规则：主分区或扩展分区的编号从1开始，到4结束；逻辑分区从编号5开始。

硬盘设备有大量的扇区组成的，第一个扇区最重要，里面保存着主引导记录与分区表信息。主引导记录（信息） 需要占用446字节，分区表为64字节，结束符占用2字节；跟去表中没记录一个分区信息就需要16字节，最多只有4个分区信息可以写到第一个扇区中，这4个分区就是4个主分区。

/dev/null文件是空设备，等价于一个只写文件，所有写入它的内容都会永远丢失。读取数据返回为空。

/dev/zero也是一个伪文件，是一个输入设备。主要用处是创建一个指定长度的初始化空文件。

/dev/full是一个特殊的设备文件，在向其写入时总是返回“设备无剩余空间”（错误码为ENPOSPC），读取时则与/dev/zero相似，放回无限的二进制流。常被用来测试程序在遇到磁盘无剩余空间错误时的行为

/dev/random[urandom]用作随机数发生器或伪随机数发生器。

Linux系统采用 设备文件 统一管理硬件设备。

按照是否对应物理实体，设备可以分为两种：物理设备，虚拟设备。

9.2 常用的设备命令

系统启动时会检测主机硬件并加载适当的驱动程序（模块，modules），让硬件正确地启动与运行。内核所检测到的各项硬件设备，会记录在/proc中

9.2.1 查看CPU信息的命令

lscpu 命令 查看CPU和处理单元的信息

9.2.2 查看内存信息的命令

最常用命令free，查看虚拟内存最常用命令vmstat

1. free 命令

显示系统内存状态，包括空闲的和已用的物流内存，swap内存，以及被内核使用的缓冲区。

-k：以KB为单位，显示内存的使用情况，默认选项

-b：以字节为单位，显示内存使用情况

-m：以MB为单位，显示

-g：以GB为单位

-t：在输出的最终结果中，输出内存和swap分区的总量

-o：不显示系统缓冲区这一列

-s：间隔秒数：根据指定的间隔时间，持续显示内存使用情况

Swap指的是交换分区，也就是虚拟内存。

1. vmstat 命令

Virtual Meomory Statistics 虚拟内存统计。Linux内存管理主要通过“调页（Paging）”和”交换（Swapping）”来完成内存调度。

页面写入磁盘的过程称作Page-Out，页面从磁盘重新回到内存的过程被称作Page-In。

当内核需要一个页面，但发现此页面不在物理内存中（因为被Page-Out）时，就发生页面错误（Page Fault）。若pageout频繁发生，内核管理分页的时间超过运行程序的时间时，系统性能会急剧下降。这时系统已经运行非常慢或进入暂停状态，这种状态亦被称作颠簸Thrashing

9.2.3 查看块设备的命令

1. dmesg 命令

--可以查看Linux系统启动信息的命令

常用于查看系统的硬件信息。除此之外，开机信息也可以通过/var/log/目录中的dmesg文件查看。

2. lsblk 命令

列出系统中的所有可用块设备信息，还能显示他们之间的依赖关系，但不会列出RAM盘的信息。块设备如，硬盘、闪存盘、CD-ROM等。

块设备指设备将数据按可寻址的块为单位进行输入/输出，一次I/O操作固定大小的数据块，通过缓冲区来读写块设备，允许随机访问。这种方式适用于发送大量的信息，常见的块设备有硬盘、磁盘、光盘驱动器等。

9.2.4 磁盘配额的命令

1. quota 命令

查询磁盘空间的限制，并得知已使用多少空间。

quota [选项] 用户名

quota [选项] 群组名

2. quotacheck 命令

扫描文件系统（必须含有挂载参数usrquota和grpquota）并建立磁盘配额记录文件。

quotacheck [选项] 文件系统

3. quotaon 命令

启动磁盘配额服务。

4. quotaoff 命令

关闭磁盘配额服务。

第10章 网络管理

10.1 管理网络接口

计算机的基本网络参数包括那些？

1. IP地址：一个IP地址标识一块网卡接口，主要有网络号和主机号组成。
2. 子网掩码：IP地址必须和子网掩码成对出现，子网掩码用来确定网络部分的位数，在IP地址中对应子网掩码为1的位为网络部分。
3. 网关：网关的功能是跨区域通信。
4. DNS：DNS将域名和IP进行映射。

1. 用Linux作为联网机器

1.1 为每个网络接口分配一个适当的IP地址和子网掩码（Netmask）；

1.2 配置默认网关（Gateway）

1.3 配置一个或一个以上的DNS服务器

2. 网络接口卡（Network Interface Card，NIC）

联网操作是通过机器上的PCI（Peripheral Component Interconnect）设备，即网络接口卡来实现。Linux内核可以检测出所有连接的PCI设备，同时使用 lspci命令可验证给出的PCI设备是否能被内核检测到。

3. 网络接口

Linux内核不允许用户将NIC作为文件进行访问。在/dev目录下没有直接关联NIC的设备节点，但有相应的硬盘和声卡设备节点

Linux系统通过网络接口访问 NIC，常用Linux接口名称和类型：

|  |  |
| --- | --- |
| 名称 | 类型 |
| eth0 | 以太网 |
| lo | （虚拟）回环设备 |
| ppp0 | 使用PPP协议的串口设备（调制解调器） |
| tr0 | 令牌环（token ring） |
| fddi0 | 光纤 |

4. 用ifconfig -a检测接口

ifconfig -a命令可检测所有目前已识别的网络接口信息

ifconfig 命令不加选项，就只能报告活跃的网络接口。

1. 用ifconfig命令配置网络接口

ifconfig 查看相关网卡信息

1. 接口配置文件

**linux接口激活----/etc/init.d/network**

**linux接口配置文件----/etc/sysconfig/network-scripts**

Linux系统采用更普遍的方法来配置网络接口，即根据目录/etc/sysconfig/network-scripts/中以ifcfg-<interface>形式命名的配置文件来配置网络接口，例如，/etc/sysconfig/network-scripts/ifcfg-eth0

习惯上分配给回环接口的IP地址和子网掩码是 127.0.0.1/255.0.0.0。

10.2 基本IP路由和网关

1. IP网络 每个IP地址包括两个部分：网络部分和主机部分。

子网掩码的网络定义为A类 B类和C类网络，网络地址分别对应为包含第1个IP字段，后面3个0：255.0.0.0；前两个网络地址字段：255.255.0.0；前3个：255.255.255.0。

2. 以太网硬件（物理）地址

每个以太网接口的地址有两个，IP地址，硬件（MAC）地址。IP地址用于两台主机间的通信，MAC地址用于同一IP网络中不同主机间的通信。

3. 在本地网络中与其他机器通信

过程：确认通向同一网络地址；确定目的机器的硬件地址，使用本地网络的底层ARP协议（Address Resolution Protocol，地址解析协议）验证；

源机器会将找到的目的及其保留IP地址/MAC地址对，即APR缓存。可用arp -a命令来检测缓存。arp -n : 查看ip地址与mac地址的映射表

用来将以太网MAC地址和一个IP地址关联起来的网络协议是 ARP。

10.3 配置DNS客户机

1. 域名系统DNS

1）将IP地址与主机名对应起来的表。只要输入一个主机的名字，计算机系统就会自动的将这个名字转换成计算机了解的IP地址，这也是Linux系统中/etc/hosts文件的由来

2）DNS是域名系统（Domain Name System）分布式数据库，作用是使复杂难记的IP地址转换成简明易记的域名，实现 主机名 与 IP地址 的转换，在TCP/IP网络中有非常重要的地位。

2. 静态查询：/etc/hosts

3. 动态查询：/etc/resolv.conf

Linux应用程序使用通用架构来解析主机名，即resolve库。当解析一个主机名时，resolve库首先试图执行静态查询；接下来resolve库将试着通过咨询/etc/resolv.conf配置文件中列出的域名服务器来进行动态查询。

10.4 DHCP 配置详解

DHCP（Dynamic Host Configuration Protocol，动态主机配置协议）是一种网络管理协议。通过DHCP获得信息的机器没有固定的IP地址，相反，每当增加一个接口时（用ifup命令）都动态地获得一个IP地址，每当减少一个接口时（用ifdown命令）就释放一个IP地址。

为将一台机器当作DHCP客户机使用，在接口配置文件中将BOOTPROTO变量设为dhcp。

当使用DHCP配置接口时，dhclient守护进程会自动启动来管理DHCP通信。所有这些由dhclient守护进程接收到的信息都会立即得到应用。默认网关会添加到路由表中，域名服务器会写入/etc/resolv.conf文件，同时NIS域（Network Information Service，网络信息服务）也会自动设定

10.5 配置Web服务器

HTTP协议的全称是Hyper Text Transfer Protocol（超文本传输协议），他是起什么作用的协议?

HTTP协议与TCP/IP协议族内的其他众多协议相同，用于客户端和服务器之间的通信。

Apache是实现WWW服务器功能的应用程序，通常所说的“浏览Web服务器”，在 服务器端 为用户提供浏览 Web服务 的就是Apache应用程序。

10.6 配置Telnet

常用的远程控制Web服务器的方法

10.7 其他网络设置

1. 分配主机名

hostname命令，可以检验机器的主机名，还可动态分配主机名。

2. /etc/sysconfig/network文件

记录通用联网配置信息，可用来定义整合到启动过程中的shell变量。

3.将网络配置视为service

网络配置可用service服务来管理，service network stop。。。

10.8 网络诊断工具

1.ping、host和traceroute命令

1）ping命令

可以测试网络中本机系统是否能到达 一台远程主机 ，用来测试两台机器间底层IP连接性（连通性）的。最简单的情况，ping命令接受一个主机名或者IP地址作为其单一参数。

2）host命令

用来直接执行DNS查询。最简单的情况，只使用一个主机名作为参数，这时host命令会使用在/etc/resolv.conf配置文件中定义的域名服务器进行DNS查询。

3）traceroute命令

为数据包在路由器间穿行并最终到达目的地路径提供了详尽的说明。

2.诊断网络配置问题

3.用tcpdump命令检测网络活性

用来直接监视网络，显示独立数据包的信息概要。会一直执行，直到Ctrl+C组合键取消这个命令。

1）可列出所有当前活跃的网络接口的命令是：ifconfig。

3）习惯上分配给回环接口的IP地址和子网掩码是：127.0.0.1/255.255.0.0

4）定义本地主机名和IP地址间转换的文件是：/etc/hosts

5）可将Linux内核的主机名设置为station=.example.com的命令是

hostname station.example.com

6） 可准确定义Linux内核的主机名，并使其在启动时被动设定的文件是 /etc/hosts

7）可启用安装了红帽Linux系统的机器，使其充当路由器角色的文件是： /etc/syscfl.conf

/etc/sysctl.conf  这个文件是用来在Linux中开启路由功能的配置文件（这个配置文件是个内核的配置文件，主要用来修改内核的参数，随系统重新启动而启动一些功能）

8）在适当的文件中，用来启用路由功能的参数名称是：net.ipv4.ip\_forward；

https://blog.csdn.net/han156/article/details/77924734

9）用来直接执行DNS主机名解析的程序是：system-config-network-tui

10）用来截取使用HTTP协议（端口80）的网页浏览器和网页服务器间数据传输的命令是，只在端口80将数据传送给网页服务器，并将传输记录用二进制格式保存在文件http.capture中：tcpdump-p 80>http.capture

11）Linux网络接口配置文件名都以 ifcfg 为前缀，文件中变量 DEVICE 用来定义该文件含有哪个接口的配置信息

12）ifconfig可以配置网卡的Ip地址等参数，这与修改配置文件有什么差别？  ifconfig 暂时改变网络参数,并不写入配置文件，因此当计算机重新启动后,还是从配置文件中设置网络参数。

net.ipv4.ip\_forward = 0 进入这个配置文件找到第七行 ，把这个0改为1 就等于开启了路由功能。

net.ipv4.ip\_forward = 1 说明Linux开启了路由转发功能

第11章 系统服务与日志

* 1. Linux系统服务

有时也称为 守护程序，是在Linux系统启动时 自动加载 并在Linux系统退出时 自动停止 的系统服务

* 1. 启动和关闭服务

1. 若依据守护程序的启动与管理方式来区分，基本上可将守护程序分为 可独立启动的服务 和 通过一个总管程序来统一管理的服务 这两大类。stand alone & super daemon.
2. 独立的服务要想启动，主要有两种方式：一是 使用/etc/init.d/目录中的启动脚本来启动独立的服务 ，二是 使用service服务来启动独立的服务
   1. 查看日志
3. 系统中绝大多数日志文件由 rsyslogd服务 来统一管理
4. Apache服务日志有Apache软件自己产生并记录，但其日志文件格式和系统默认日志的格式是一致的。
5. /var/ 用来保存系统动态数据的目录，/var/log/目录 就是系统日志文件的保存位置。
   1. 管理日志

rsyslogd服务依赖其配置文件 /etc/rsyslogd.conf 来确定各个服务不同等级的日志信息会被记录的位置

第三篇 应用篇

第12章 VI编辑器

12.2 工作模式

命令模式、文本编辑模式和末行模式3种。

12.3 进入和退出VI

12.3.1 进入VI

vi ：进入VI

vi 文件名 ：文件名不存在则创建文件，存在则打开文件

vi+ 文件名 ： 进入VI，光标停在文件的最后一行开始处

vi+n 文件名 ：进入VI，光标停在文件的第n行开始处

vi+/字符串 文件名 ： 进入VI，光标停在文件中指定字符串第一次出现的行首。

12.3.2 保存文件或退出VI

命令模式下按两次Z键，将保存文件并退出VI。

末行模式下

w ：write，保存当前正在编辑的文件，但不退出VI

q ：quit，不进行文件保存，直接退出VI，若文件有改动过而没有保存，将产生错误。

q！ ： 强行退出VI，若文件内容有改动则恢复到文件的原始内容。

wq ：保存并退出VI （ZZ）

12.5 文本文件编辑

12.5.1 文本的插入

插入文本可以在命令模式下输入i、a或o命令。文本框底部的“INSERT”表示编辑器正处于文本编辑模式。

12.5.2 光标的移动

命令模式下：

o ： 在下一行插入一个空行，光标移动到行首

$ ： 光标移动到当前行的行尾

nw ： 光标右移n个字（或英文单词）

w ： 光标右移1个字（或英文单词）

nb ： 左移n个字（或英文单词）

b ： 左移1个字（或英文单词）

（ ： 光标移动本句的句首，若已经处于本句句首，则移动到前一句的句首

） ： 光标移到下一句句首

{ ： 光标移到本段段首，若已经处于本段段首，则移动到前一段段首

} ： 光标移到下一段段首

1G ： 光标移到文件首行的行首

G ： 光标移到文件末行的行首

nG ： 光标移到文件第n行行首，n为整数

Ctrl+G ： 报告光标所处位置，位置信息显示在VI最后一行

12.5.3 删除

文本编辑模式下，Delete键可以删除光标坐在位置的后一个字符，Backspace键可以删除光标所在位置的前一个字符。

命令模式下：

x ： 删除光标坐在位置的一个字符

nx ： 删除从光标开始的n个字符

dw ： 删除从光标开始的一个字

ndw ： 删除从光标开始的向右n个字

db ： 删除光标左边的一个字符

ndb ： 删除从光标开始的向左n个字符

d0 ： 删除从光标前一个字符到行首所有字符

d $ ： 删除光标所在字符到行尾的所有字符

dd ： 删除光标所在的行

ndd ： 删除从当前行开始的n行

d( ： 删除从当前字符开始到句首的所有字符

d) ： 删除从当前字符开始到句尾的所有字符

d{ ： 删除从当前字符开始到段首的所有字符

d} ： 删除从当前字符开始到段尾的所有字符

命令模式下：

u ： undo，可以撤销前一次的操作

12.5.4 查找与替换

命令模式下，查找文本：

？字符串 ： 在文本内容中查找与字符串对应的内容 （区分大小写）

n ： 查找下一个匹配

N ： 查找上一个匹配

替换文本：

s/查找的字符串/新字符串 ： 在当前行用新字符串替换查找的字符串，只替换第一个查找到的字符串。

s/查找的字符串/新字符串/g ： 在当前行用新字符串替换所有的查找的字符串。

1，ms/查找的字符串/新字符串/g ： 在第1行到m行用新字符串替换所有查找的字符串。

1，$ s/查找的字符串/新字符串/g ： 在整个文件中用新字符串替换所有查找的字符串。

12.5.5 复制与粘贴

yw ： 将光标所在位置到字尾的字复制到缓冲区中。

nyw ： 将光标所在位置开始的n个字复制到缓冲区中。

yb ： 从光标开始向左复制一个字符

nyb ： 从光标开始向左复制n个字符

y0 ： 复制从光标前一个字符到行首的所有字符

y $ ： 复制从光标开始到行末的所有字符

yy ： 复制当前行

nyy ： 复制从当前行开始的n行

p ： 在光标所在位置的后面插入复制的文本

P ：大写，在光标所在位置的前面插入复制的文本

np ： 在光标所在位置的后面插入复制的文本，复制n次

nP ： 在光标所在位置的前面插入复制的文本，复制n次

要复制example.txt所有文本，首先将光标移动到文件的开头位置，然后在命令模式下输入3yy，输入完成后在最后一行显示“3 lines yanked”，表示文本已经被复制。

12.6 存储缓冲区

数字编号缓冲区和命名缓冲区

12.6.1 数字编号缓冲区

VI编辑器使用9个临时缓冲区，编号从1到9。

每次新删除或复制的文本替换缓冲区以前的内容，“dd”命令后，VI将被删除的行存到缓冲区1，当再次dd删除另一行，VI将以前的内容放到下一个缓冲区2，新内容存到缓冲区1。

缓冲区总是保留最新被修改的内容。

使用p命令并在它前面加上缓冲区号就可以找回数字编号缓冲区的内容。如，9p，表示要将缓冲区9的内容复制到光标位置。指定缓冲区的格式可以表示为 np

12.6.2 命名缓冲区

26个命名缓冲区，用小写字母a到z命名。

用户可将删除或复制的文本存到指定缓冲区，然后使用p命令将指定缓冲区的文本复制到文件的其他地方。命名缓冲区内容保持不变，知道用户在删除或复制操作中指定了该缓冲区。

缓冲区命名格式：缓冲区名+命令

如，wdd 表示删除当前行，将它的副本保存到缓冲区w。z3yy，表示复制3行到缓冲区z。

12.7 编辑多个文件

启动VI时，可以给出多个文件名，当结束一个文件的编辑时，就可以继续编辑下一个文件。

输入n命令启动下一个编辑文件，若当前文件已被修改，就会显示“No write since last change（add ！to override）”，表示自从上次修改后不能写入，可n！命令忽略这个提示信息。

ar命令查看文件名列表。

12.8 VI编辑器的设置

12.8.1 set命令选项

用来设置选项，选项分为3类：布尔触发器、数字式、串。

1）布尔触发器，用户可以打开或关闭他们，像开关。通过名字来设置，在名字前面加上no来禁止。如，set C表示打开，输入set no C表示关闭。

2）数字式选项 接受数字值，选项不同，数值范围也不同。如，某项可以使用set C=1，表示将1赋给C。

3）串选项 同数字式选项类似，但是它接受串值。如，输入set C=yes，表示将yes赋给C。

set命令基本格式如下：

1. “set all” 表示在屏幕上显示所有的选项；
2. “set ” 表示只显示修改过的选项；
3. “set C？”表示显示选项C的值。

12.8.2 设置VI环境

时效性，set命令设置想要的值

常用VI参数选项：

autoindent：缩写ai，将新行与前一行的行首对齐。

ignorecase：缩写ic，在搜索选项中忽略大小写。

magic：允许在搜索时使用特殊字符。

number：缩写nu，显示行号。

report：通知用户上一个命令影响的行号。

scroll：设定使用Ctrl+D键命令翻动的行数。

shiftwidth：缩写sw，设置缩进的空格数，与autoindent选项一起使用。

showmode：缩写smd，屏幕的右下角显示VI编辑器的模式。

terse：显示简短的错误信息。

1）若用户想在每次对文件进行编辑时都收到反馈信息，可以输入 set report=0

2）VI编辑器打开文档后，想把10-15行中的第一个“abc”替换为“efg”，应输入命令

10, 15s/abc/efg

3）下列命令的含义：? VI：在文本内容中查找与“VI”对应的内容

1, $ s/VI/Emacs/g：在整个文件中用“Emacs”替换所有“VI”

3yy G P：复制从当前行开始的3行，并粘贴到文件末行的行首

4）编辑example.txt文件，把3行到10行间的字符串is都替换为are，在末行模式下输入：3，10s/is/are/g

5）删除从16行到20行的所有内容，应输入：16G 5dd

6）输入命令2p的含义是：将缓冲区2的内容复制到光标所在位置

7）将3至7行的内容复制到第1行行首应输入命令：3G 5yy 1G P

第13章 Emacs编辑器

13.2 Emacs的操作

ubantu

sudo apt-get install emacs

13.2.2 Emacs的启动

启动：emacs 或 emacs加上文件名

界面：标题栏、菜单栏、窗口区域、模式行、回显行。

模式行：启动界面倒数第二行。模式行显示状态信息。包括以下几个字段：

-:--- buf pos L#（Pattern）

1）冒号后面的3个字符中前2个字符表示缓冲区中文本的状态。

--：表示缓冲区未被编辑。

\*\*： 表示缓冲区文本已经被修改。

%%： 表示只读缓冲区中文本未被修改。

%\*： 表示只读缓冲区中文本已经被修改。

2）buf显示窗口缓冲的名字，一般为文件名。

3）pos可指示以下选项：All表示显示了整个文件；Top表示显示了文件的开始部分；Bot表示显示了文件的末尾部分；NN%表示在窗口顶部以上的部分显示缓冲区文件的百分比。

4）L#表示字母L后跟当前光标所在行的行号。

5）（Pattern）表示缓冲区中有效的模式。可用主模式包括Fundamental模式、Text模式、Lisp模式、C模式和其他模式。

13.2.3 Emacs的退出

先按Ctrl+X组合键，再按Ctrl+S键 ： 保存文件并退出Emacs

先按Ctrl+X组合键，再按Ctrl+C键 ： 不保存文件退出Emacs

先按Ctrl+X组合键，再按Ctrl+W键 ： 另存为文件

13.2.4 文件与缓冲区操作

先按Ctrl+X组合键，再按Ctrl+F键 ： 搜索并打开文件

先按Ctrl+X组合键，再按Ctrl+V键 ： 打开另一个文件以替换当前打开的文件

先按Ctrl+X组合键，再按Ctrl+I键 ： 在光标所在处插入文件

按组合键Alt+X ： 恢复自动保存的文件

先按Ctrl+X组合键，再按Ctrl+B键 ： 切换缓冲区

先按Ctrl+X组合键，再按Ctrl+S键 ： 保存缓冲区

按Ctrl+G组合键 ： 撤销命令

13.2.5 Emacs的基本操作

1. **光标的移动**

Ctrl+F 将光标向前移动一个字符

Ctrl+B 将光标向后移动一个字符

Ctrl+P 将光标移动到上一行

Ctrl+N 将光标移动到下一行

Ctrl+A 将光标移到当前行的行首

Ctrl+E 将光标移到当前行的行尾

Ctrl+V 将光标向前移动一屏

Alt+V 将光标向后移动一屏

Alt+F 将光标向前移动一个单词

Alt+B 将光标向后移动一个单词

Alt+< 将光标移到文本开头

Alt+> 将光标移到文本末尾

2. 删除

Backspace 删除光标前的一个字符

Ctrl+D 删除光标处的字符

Ctrl+X，再按U 撤销上次修改

Alt+D 删除光标后的一个单词

Ctrl+K 删除从光标到该行行尾的所有字符

Ctrl+X，再按Backspace 删除前一句

Ctrl+W **删除两个位置间的所有文本**

3. 重排

Ctrl+空格键 在当前点位置设置标记

Ctrl+X，Ctrl+X 互换光标点和标记的位置，该命令可以用来显示标记的位置

如，对于example.txt文件，将光标移动到第2行首字母T上。Ctrl+空格键，在屏幕底部的回显行会显示信息“Mark set”。将光标移至文件末尾设置点，按组合键Ctrl+X，Ctrl+X，互换光标点和标记，显示出标记的位置，确保设置了标记。

按Ctrl+W，删除选中的文本。被删除的文本放置在缓冲区内，可以重新插入文本的其他位置。

将光标放于第2行，按Ctrl+Y组合键，在光标位置插入被删除的文本

4. 搜索和替换

Ctrl+S 向前递增搜索

Ctrl+R **向后递增搜索**

替换命令需要提供两个参数：搜索字符串和替换字符串。每一个参数都以Enter键结束。命令按键为：先按Alt+Shift+5组合键，然后输入要搜索的字符串，再按Enter键，Emacs会提示输入要替换的字符串，输完按Enter。

空格键或Y 替换当前匹配并进入下一匹配处

Delete键或N 忽略此次匹配并进入到下一匹配处

，（逗号）键 显示替换结果

Enter键或Q 直接退出

.（句号）键 替换当前匹配并退出

！键 直接替换剩下的所有匹配

^键 回到前一匹配处

5. 大小写字母转换

将光标放在单词前

Alt+U 将整个单词转为大写

Alt+L 将整个单词转为小写

Alt+C **将首字母改为大写**

Ctrl+X，Ctrl+U 将指定区域改为大写

Ctrl+X，Ctrl+L 将指定区域改为小写

6. 帮助操作

Ctrl+H 调用Emacs的帮助

7. 窗口操作

Ctrl+X，再按2 将当前窗口水平分割为两个窗口

Ctrl+X，再按3 将当前窗口垂直分割为两个窗口

Ctrl+X，再按> 向右滚动当前窗口

Ctrl+X，再按< 向左滚动当前窗口

Ctrl+X，再按0 删除当前窗口

Ctrl+X，再按O 将光标放在其他窗口上

Ctrl+X，再按1 删除当前窗口外的所有窗口

13.2.6 Emacs环境下编译并运行C/C++程序

启动Emacs，输入 “emacs -nw” 命令

先按Ctrl+X，再Ctrl+F，打开文件 test.c

Ctrl+X，Ctrl+S，保存文件

编译文件，Alt+X，输入“eshell”命令进入缓冲区

输入“gcc test.c -o test” 命令以编译源文件，并通过“./test”命令运行程序

1）下列不属于光标移动命令的组合是

A. Ctrl+F B. Ctrl+A C. Alt+D D. Alt+B

2）重排是对文本中的内容进行操作，选中的一段文字表示区域，区域的起点叫做 标记，光标移动到的位置和标记之间就是区域，光标在文本中的位置叫做 光标点。

3）若回显行显示“Find file:/tmp/”，说明用户执行了搜索并打开文件的命令：先按Ctrl+X组合键，再按Ctrl+F组合键

4）用户如果想查看在Emacs中保存的文件，可通过 Enter键 命令查看文件列表。

第14章 OpenSSH

OpenSSH是安全shell协议（secure shell，简称SSH）的免费开源实现。可以用来进行远程控制、在计算机之间传送文件等。

SSH采用非对称加密技术（RSA）加密所有传输的数据。

OpenSSH提供了服务端后台程序和客户端工具，对远程控制和文件传输过程的数据进行加密处理。

SSH是 安全外壳 协议。

14.1 安装OpenSSH

OpenSSH是Linux最常用的SSH服务器/客户端软件。

CentOS 下，查看是否默认安装了OpenSSH，rpm -qa openssh\*

启动服务：service sshd start

开启防火墙的22号端口：

sudo firewall -cmd –zone =public –add-port=22/tcp --permanent

14.2 配置OpenSSH

OpenSSH的主配置文件 /etc/ssh/sshd\_config。

常见的配置选项：

1）设置SSH的端口号是22（默认端口号22）：Port=22。

14.3 OpenSSH的使用

14.3.2 文件传输

**SFTP** 是安全文件传送协议，可以为传输文件提供一种安全的加密方法。**SCP** 是远程文件复制的协议，复制过程是加密的。

1）以下协议中，为远程登录会话和其他网络服务提供安全性的协议是

A. FTP B. HTTP C. SSH D. ICMP

2）SSH采用的加密算法是： RSA

3）SSH是由IETF（the Internet Engineering Task Force）指定的建立在应用层基础上的 安全网络 协议。

4）SSH是专为 远程登陆 会话和其它网络服务提供安全性的协议。

5）SSH采用了 非对称加密技术（RSA）加密所有传输的数据

第15章 Linux系统的应用程序开发技术

15.5 GCC编辑器

15.5.1 C语言程序的编译过程

gcc -E test.c -o test.i

gcc -S test.i -o test.s

gcc -c test.s -o test.o

gcc test.o -o test

简化的GCC命令：

gcc test.c -o test

15.5.2 GCC基本用法

gcc [选项] [文件名]

默认情况下，GCC会在系统默认的路径中（如usr/lib）寻找所需的库文件，当使用 -L 选项，会首先到-L指定的目录下去寻找相关库文件。

15.6 GDB调试器

15.6.1 GDB的启动和退出

输入命令 “gdb 程序名“ 或者先输入 “gdb” ， 然后输入 “file 程序名”

15.6.2 调试前的准备

使一个可执行文件被GDB调试，需要在编译程序时加入“-g”，这个选项可以在GCC编译程序时产生调试信息。

gcc -g test.c -o test

./test

若想要详细了解某个gdb命令的使用方法，可使用help命令“help list”和help all“”

15.6.3 显示和查找源代码

调试时查看源代码可使用“list”命令：

list ： 每次调用该命令显示10行代码，若再次运行该命令，则显示接下来的10行

list m,n 显示从第m行到n行的代码

list 源文件:m,n 显示源文件中第m行到n行的代码

list 函数名 显示函数周围的代码

list 源文件:函数名 显示源文件中函数名指定的周围代码

可以在源文件中从当前行向后查找第一个匹配的字符串，命令格式：“search 字符串””forward 字符串”。也可反向查找字符串，“reverse-search 字符串”。

15.6.4 执行程序

要使程序开始运行，在（gdb）的提示符下输入“run”或“r”

15.6.5 设置断点

1. break 行号 ： 将断点设置在某行

如，（gdb）break 4

Breakpoint 1 at 0\*400525：file test.c line 4.

1表示第一个断点，0\*400525是断点所在的内存地址

设置好断点后，可以输入“run”命令运行程序

2. break 函数名 ： 断点设置在函数名上

3. break 行号或函数名 if 条件 ： 通过条件设置断点

4. info breakpoints 命令 查看当前所有的断点

Num 表示断点的编号；

Type 表示断点的类型；

Disp 表示断点在生效一次后是否失去作用，如果是则为dis，否则为keep；

Enb 表示当前断点是否有效，如果是，y，否则为n；

Address 表示终端所处的内存地址；

What 表示发生在哪个函数的第几行，“stop only if i==4”标明是一个条件断点。

5. disable 断点编号 ： 使断点失效

还可通过“enable”继续使其生效

6. 删除断点

clear 删除程序中的所有断点

clear 行号 删除此行的断点

clear 函数名 删除该函数的断点

clear 断点编号 删除指定编号的断点

15.6.6 查看和设置变量的值

执行到断点的时候，往往需要通过查看变量的值来了解当前程序的执行情况

1. 查看变量的值

print 变量名或表达式

2. 查看变量的类型

whatis 变量名或表达式，用于显示某个变量或表达式结果的数据类型

3. 设置变量的值

set variable 变量=值

print 变量=值

15.6.7 调试的控制

查看变量的值后，可以让程序继续运行。是要一步一步运行，还是让程序一直运行下去

1. kill命令

用于结束当前程序的调试，输入该命令后，会询问是否退出当前程序的调试，输入y结束，输入n继续调试。

2. continue 命令

让程序继续运行到下一个断点或运行完整个程序

如，使用run让程序运行到第1个断点处，查看变量n的值，接下来使用continue让程序继续运行至第2个断点处，再查看n的值。

=

3. next和step命令

单步调试功能，每次执行一条语句

next命令会把该函数当作一条语句执行，不会进入函数的内部；

step命令则会进入函数内部，继续单步调试程序。

使用next命令每次执行一条语句，执行一条语句后可以使用print命令查看变量的值。

4. nexti和stepi命令

用于单步执行一条指令，而不是一条语句，对一条语句而言，往往对应多条指令。

如，对一条循环语句“for（i=0；i<6；i++）”而言，使用“nexti”或“stepi”命令，需要输入多次才能执行完毕。

GDB命令

list 缩写l 显示多行源代码

break 缩写b 设置断点

info 缩写i 描述程序的状态

run 缩写r 开始运行程序

display 缩写disp 跟踪查看某个变量

step 缩写s 执行下一条语句，若该语句为函数调用，则进入函数，执行其中的第一条语句

next 缩写n 执行下一条语句，若该语句为函数调用，则不会进入函数内部执行

print 缩写p 打印内部变量值

continue 缩写c 继续程序的运行，直到遇到下一个断点

set variable name=v 设置变量的值

start 缩写st 开始执行程序，在main函数的第一条语句前面停下来

file 装入需要调试的程序

kill 缩写k 终止正在调试的程序

watch 监视变量值的变化

backtrace 缩写bt 查看函数调用信息

frame 缩写f 查看栈帧

quit 缩写q 退出GDB环境

15.7 make

make命令是GNU工程化编译工具，用于编译大量互相关联的源代码，使用它可以实现项目的工程化管理，提高开发效率。

make工具可以用来维护程序模块关系和生成可执行程序。执行该命令时，将make工具的编译指令放入一个名为makefile（或Makefile）的文件中，该文件定义了整个项目的编译规则、模块之间的依赖关系和文件的编译顺序。

make命令可以使整个项目的源文件进行自动编译，用户只需要运行一条命令，这种方式极大的提高了软件的开发效率。

make命令从makefile文件中获取模块之间的依赖关系。

15.7.1 makefile文件的构成

一个makefile文件的基本单元是规则，一条规则指定一个或多个目标文件，目标文件后面跟的是编译生成目标文件所依赖的文件或模块，最后是生成目标文件的命令。

规则格式如下：

目标文件列表 分隔符 依赖文件列表

[命令]

[命令]

“[]”代表该项可选。

15.7.2 执行makefile

当用户输入make命令后，首先在当前目录下寻找名为makefile或Makefile的文件。

1）头文件#include<my.h>和#include”my.h”中的<>和“”有什么区别？

答：<>表示编译器将在系统预设的头文件目录下搜索头文件；“”表示先在当前目录下搜索，找不到时再去系统头文件目录下搜索头文件。

第16章 Web服务器搭建、应用开发与部署

Linux下，采用Eclipse开发工具，基于MVC的Java Web开发，使用当下较为流行的SSM框架进行演示。

16.1 Web服务器搭建的准备

16.1.1 配置Java

要先将Linux自带的openJDK卸载，后安装Oracle的JDK

java -version 查看已安装的Java版本信息

rpm -qa | grep java 查找已安装的openJDK位置

之后逐一卸载，下载JDK压缩包

解压后，**配置环境变量** gedit ~/.bashrc

bashrc文件中配置4个环境变量：JAVA\_HOME，JRE\_HOME，PATH，CLASSPATH

配置完毕关闭文件，在终端执行 source ~/.bashrc 使环境变量立即生效

16.1.2 配置MySQL

MySQL的配置文件my.cnf p350

#设置MySQL的安装目录 basedir

#设置MySQL数据库的数据存放目录 datadir

#

16.2 Web服务器的搭建

16.2.1 安装配置Tomcat

使用 startup.sh 命令启动服务，调用方法 /usr/local/tomcat8/bin/startup.sh

使用 shutdown.sh 命令关闭服务

16.2.2 配置DNS、DHCP、FTP服务

1. 配置域名系统DNS

实现域名解析的软件模块是 BIND。

DNS服务使用的端口是 53。

设置正向解析的数据库文件（named.bob.com)

设置反向解析的数据库文件（named.172.16.5)

2. 配置DHCP

动态主机配置协议DHCP是一个局域网的网络协议。

3. 配置FTP

FTP是文件传输协议。

16.3 开发环境与配置

16.4 网站设计开发与部署

16.4.1 MVC 简介

MVC 是模型 Model 视图 View，控制器 Controller， 一种软件设计典范。

16.4.2 常见框架

1. Struts 2

Struts 2 是一个基于模型、视图、控制器的MVC设计模式应用框架。

2. Spring

Spring是一个轻量级控制反转（IoC）和面向切面（AOP）的容器框架。

3. MyBatis

M有Batis是一款优秀的持久层框架，支持定制化SQL、存储过程以及高级映射。