

习题和答案

第 1 章 Linux 概述	1
第 2 章 shell 及常用命令	4
第 3 章 vi 编辑器的使用	7
第 4 章 X Window 系统的使用	9
第 5 章 Linux 系统的常用软件	11
第 6 章 硬件管理	11
第 7 章 网络基本配置	12
第 8 章 常用网络服务的配置和使用	15
第 9 章 系统管理与监控	19
第 10 章 Linux 系统的安全管理	21
第 11 章 shell 程序设计	24
第 12 章 gcc 的使用与开发	26
第 13 章 gtk+图形界面程序设计	27
第 14 章 Qt 图形界面程序设计	28
第 15 章 集成开发环境 KDevelop 的使用	31

第 1 章 Linux 概述

1. 什么是 Linux?

Linux 是一套免费使用和自由传播的类 UNIX 操作系统，源代码开放，能运行于各类硬件平台，包括 Intel x86 系列和 RISC 处理器。这个系统是由世界各地成千上万的程序员设计和实现的。其目的是建立不受任何商品化软件的版权制约的、全世界都能自由使用的 UNIX 兼容产品。

2. Linux 有哪些特性?

- (1) 开放性
- (2) 多用户
- (3) 多任务
- (4) 良好的用户界面
- (5) 设备独立性
- (6) 丰富的网络功能
- (7) 可靠的系统安全
- (8) 良好的可移植性

3. Linux 与 Windows 操作系统的主要区别是什么?

(1) 从发展的背景看，Linux 是从一个比较成熟的操作系统发展而来的，而其他操作系统，如 Windows 等，都是自成体系，无对应的相依托的操作系统

(2) 从使用费用上看，Linux 是一种开放、免费的操作系统，Windows 是封闭的系统，需要有偿使用。

(3) Linux 上丰富的应用软件也是自由的，而在 Windows 下，几乎所有的软件都有独立的版权，需要购买使用，即使某些软件可以免费使用，也一般不提供其源代码，更不用说由用户修改扩充其功能了。

(4) Windows 对硬件配置要求高，而 Linux 在低端 PC 系统上仍然可以流畅运行

4. Linux 与 Unix 的共同点与不同点是什么?

共同点：由于 Linux 是从 Unix 发展来到，它遵循 Unix 开放标准，基本支持同样的软件、程序设计环境和网络特性，可以说 Linux 是 UNIX 的 PC 版本，Linux 在 PC 机上提供了

相当于 UNIX 工作站的性能。

与商用 Unix 的不同点有：1) Linux 是免费软件，用户可以从网上下载，而商用的 UNIX 除了软件本身的价格外，用户还需支付文档、售后服务费用；2) Linux 拥有 GNU 软件支持，Linux 能够运行 GNU 计划的大量免费软件，这些软件包括应用程序开发、文字处理、游戏等方面的内容；3) Linux 的开发是开放的，任何志愿者都可以对开发过程做出贡献；而商用 UNIX 则是由专门的软件公司进行开发的。

与自由 Unix 的不同点：

1) 在组织方式上，FreeBSD 由它的核心团队（core team）的领导，他们负责原始程序的开发与维护。有 core team 的优点是原始程序会有一致性，会有组织的被更新，但是整个系统的活力操纵在 core team 手中，缺乏活力。Linux 没有核心团队，在 Linus 的主导下来自世界各地的爱好者都可以发布自己的 patch，缺点是源代码杂乱无章且可能会相互冲突。

2) 在发展方向上，FreeBSD 的核心团队将主要精力投入在 UNIX 自身的风格和特点上。Linux 通常会首先加入商品化系统上的各种东西，比如新的硬件驱动、samba 等。因此，从易用和可用上讲，Linux 更容易上手和使用。

3) 在系统核心功能上二者区别不大，但在 Linux 系统上，几乎可是找到任何需要的功能

4) 在系统的性能上，据专家分析，FreeBSD 在网络性能、软件移植性和系统规范化上略胜一畴，而在硬件支持、磁盘 IO 操作等方面 Linux 略强一些。

5. 什么是 GNU 软件，什么是 GPL 和 LGPL？

GUN 项目主要由自由软件基金资助的一个项目，目标是开发一个自由的、UNIX 类型的操作系统，称为 GNU 系统。GNU 是“GNU’s Not UNIX”的首字母的递归缩写，目前使用 Linux 内核的各种 GNU 操作系统应用非常广泛。所有的 GNU 软件和派生工作均遵循 GNU 通用公共许可证，即 GPL。

GPL 和软件是否免费无关，它主要目标是保证软件对所有的用户来说是自由的。GPL 通过如下途径实现这一目标：

- 它要求软件以源代码的形式发布，并规定任何用户能够以源代码的形式将软件复制或发布给别的用户。
- 它提醒每个用户，对于该软件不提供任何形式的担保。
- 如果用户的软件使用了受 GPL 保护的软件的一部分，那么该软件就继承了 GPL 软件，并因此而成为 GPL 软件，也就是说必须随应用程序一起发布源代码。
- GPL 并不排斥对自由软件进行商业性质的包装和发行，也不限制在自由软件的基础上打包发行其他非自由软件。
- 遵照 GPL 的软件并不是可以任意传播的，这些软件通常都有正式的版权，GPL 在发布软件或者复制软件时声明限制条件。但是，从用户的角度考虑，这些根本不能算是限制条件，相反用户只会从中受益，因为用户可以确保获得源代码

LGPL 允许用户在自己的应用程序中使用程序库，即使不公开自己的源代码。但是，要求用户必须能够获得在应用程序中使用的程序库的源代码，并且允许用户对这些程序库进行修改。

遵循 LGPL 的一种方法是，随应用程序一起发布目标代码以及可以将这些目标程序和受 LGPL 保护的程序库链接起来的 makefile 文件。

遵循 LGPL 的另一种比较好的方法是使用动态链接

6. 什么是 Linux 的内核版本和发行版本？

内核版本指的是在 Linux 的创始人 Linus 领导下的开发小组开发出的系统内核版本号，Linux 内核的版本号主要由三部分构成：主版本号、次版本号、次次版本号。主版本号表示

系统内核有大的改动，次版本号表示系统内核有小的改动，开始支持一些新的特性，一般表示系统内核对新的硬件支持进行了改进。如果更改之后还处于测试阶段，那么次版本号为奇数，如果已经过了测试阶段，改动是成功的，那么次版本号是偶数。次次版本号表示系统有一点儿改动，对内核影响不大。

一些组织或公司将 Linux 内核与应用软件和文档包装起来，并提供安装界面、系统配置与管理工具，这样就构成了一种发行版本。发行版本的版本号随发行者的不同而不同，与系统内核的版本号是相互独立的。

7. 你知道哪些主要的 Linux 发行版本？

Red Hat Linux, Red Flag Linux, Blue Point, TurboLinux, Suse Linux, Debian GNU/Linux, Mandrake Linux 等。

8. 什么是内核，它的作用是什么？

内核是系统的核心，是运行程序和管理磁盘、打印机等硬件设备的核心程序。它的总共有管理 CPU、内存和外设。

9. 什么是 shell，shell 的作用是什么？

shell 是系统的用户界面，它实际上是一个命令解释器，它解释由用户输入的命令并把它们送到内核，把执行的结果显示给用户。不仅如此，shell 有自己的编程语言，允许用户编写由 shell 命令组成的程序。

10. 在安装 Linux 之前应该如何获取系统的硬件资料？

在安装 Linux 之前，先确定你的计算机的硬件是否能被 Linux 所支持。需要关注的硬件包括显卡、声卡、网卡、打印机、鼠标等。要获知相应硬件的型号，可以在 Windows 下查看，也可以通过设备本身的标志查看。检查有没有相应的设备驱动程序。关注硬盘分区、大小等信息。如果声卡、显卡的型号不能识别，一般可以模拟成标准硬件来使用。

11. Linux 有几种安装方法？

- 1) 从光盘安装
- 2) 从硬盘安装

12. Fedora10 在安装时提供哪几种方式建立 Linux 的分区？

Fedora10 在安装时提供了五个关于分区的选项：

在选定磁盘上删除所有分区并创建默认分区结构：如果想在磁盘上仅安装一个 Fedora 系统，可以选择这个选项。特别注意，如果选择了这个选项，那么该磁盘中已经安装的所有其他系统将被破坏。

在选定驱动器上删除 Linux 分区并创建默认的分区结构：如果系统中存在任何 Linux 分区，选择这个选项可以删除所有的 Linux 分区，然后在腾出的空闲分区中安装 Fedora 系统。选择这个选项不会影响或者修改已经分配给其他非 Linux 操作系统占用的分区空间。

重新定义现有分区大小并在剩余空间中创建默认布局：该选项允许调整现有分区的大小，然后自动创建默认的分区结构。

使用选定驱动器中的空余空间并创建默认的分区结构：保留现有分区中的数据，仅在空闲空间上创建新分区。

建立自定义的分区结构：在前几个选项中，系统根据驱动器的大小以及内存大小自动创建分区布局。本选项则允许用户自己定义分区结构。选择该选项后，将打开分区管理窗口，在这里可以方便地创建、编辑、删除分区。

13. 安装 Linux 至少需要哪两个分区？还有哪些常用分区？

至少有 / 和 swap 分区。其它常用分区有：/bin /home /usr /var /usr/local /tmp 等

14. 安全专家建议，安装 Linux 时，最好为主要的目录建立单独的分区，试分析这样做的优点。

由于每个分区指定了自己的大小，当某个分区的数据超过自己的大小限制时，不会挤占其它分区的空间。

15. swap 交换分区的作用是什么？

它实际上是用硬盘实现虚拟内存，即当系统内存使用率比较高的时候，内核会自动使用 swap 分区来模拟内存。

16. 安装 Linux 都需要收集哪些信息？

安装过程所使用的语言、键盘鼠标类型、安装类型、磁盘分区、系统语言、安装软件包。

17. 如何在安装时选择安装某个软件？

在安装过程中，安装程序将显示软件选择界面。默认情况下，Fedora 安装程序会自动安装一组适合桌面图形环境的软件。在该界面上，还可以选择软件安装所需的程序库。选择“现在定制”后点击【下一步】按钮，打开软件包定制界面。在这里，可以对系统安装的软件进行更详细的定制。

18. 什么是运行级别？Linux 有几个运行级别？

运行级别是指 Linux 的运行状态，Linux 系统有六种不同的运行级，这六种运行级分别为：

- 0: 停机
- 1: 单用户模式，就像 Win9X 下的安全模式。
- 2: 多用户，但是没有 NFS
- 3: 完全多用户模式，标准的运行级
- 4: 保留，一般不用
- 5: X Window 图形界面模式
- 6: 重新启动

19. 如何更改默认启动级别？

在配置文件/etc/inittab 文中，有一行文本“id: 5: initdefault: ”标志了 Linux 的默认启动级别，5 表示默认进入 X Window 图形界面，如果把这里的数字改为 3，则默认进入控制台界面。注意，不要把 initdefault 设置为 0 和 6，这两个级别是系统在关机和重启过程中自动使用的。

20. 如何登录和注销？

登录分为图形界面登录和命令行登录，在提示用户名和口令时分别输入用户名和口令即可登录。

要注销图形化环境，选择【系统/注销】命令，弹出注销确认对话框，或者按<Ctrl+Alt+Backspace>键，直接执行注销命令。

输入命令 logout，则又回到登录界面，可以重新登录。

21. 如何安全关闭 Linux 系统？

在图形界面下，选择【系统/关闭】命令，选择【关闭系统】按钮执行关闭命令，系统关闭自身所有进程。也可以在注销后出现的重新登录界面上选择【关闭】按钮来关闭系统。某些计算机会在关闭系统后自动切断电源，如果计算机不这样做，那么，当看到“Power down.”消息后，便可以手工切断计算机的电源。

可以在控制台执行 halt 命令、init 命令或者 shutdown 命令来关闭系统。

第 2 章 shell 及常用命令

1. 判断下列命题是否正确

- 1) Linux 的文件名与命令不区分大小写。(×)。
- 2) \$ls **和\$ls *"*"的显示结果完全一致。(×)
- 3) rm 和 rmdir 的作用一样，都是删除整个目录。(×)
- 4) Linux 文件名的命名规则必须遵守 8.3 的格式。(×)

2. 什么是 Linux 终端？Linux 终端又称为什么？

Linux 终端也称为虚拟控制台，是 Linux 从 UNIX 继承来的标准特性。显示器和键盘合称终端，因为它们可以对系统进行控制，所以又称为控制台，一台计算机的输入输出设备就是一个物理的控制台。如果在一台计算机上用软件的方法实现了多个互不干扰独立工作的控制台界面，就是实现了多个虚拟控制台。Linux 终端的工作方式是字符命令行方式，用户通过键盘输入命令进行操作，可以通过 Linux 终端对系统进行控制。

3. 默认情况下，Linux 有几个虚拟终端？如何在不同终端之间切换？如何在 x Window 与终端之间进行切换？

通常情况下，Linux 默认启动六个虚拟终端。虚拟控制台的选择可以通过按下<Alt>键和功能键<Fn>(n=1~6)来实现。如果启动方式选择了直接启动 X Window，那么 X Window 在第七个虚拟终端上。在 X Window 图形操作界面中按<Alt+Ctrl+Fn>(n=1~6)就可以进入控制台字符操作界面。在控制台操作界面里按<Alt+Ctrl+F7>即可回到刚才的 X Window。

4. 什么是 shell？shell 在用户与操作系统之间的作用是什么？

shell 是一个命令解释器，它可以用来启动、挂起、停止程序，还允许用户编写由 shell 命令组成的程序。shell 是系统的用户界面，它提供了用户与内核进行交互的一种接口。它接收用户输入的命令并把它送入内核执行，把执行结果显示给用户。

5. 什么是通配符？常用的操作通配符有哪些？

Linux 系统中的通配符除了“*”和“?”以外，还可以使用“[”、“]”和“-”组成的字符组模式，以便精确地扩充需要匹配的文件范围。

通配符*：通配符*可以代表文件名中的任意字符或字符串。通配符*不能与单句点开头的文件名相匹配。以单句点开头的文件在 Linux 中是隐藏文件。

通配符?：通配符?可以匹配任意一个字符。

字符组模式：通配符“[”、“]”、“-”用于构成字符组模式。“[”和“]”将字符组括起来，表示可以匹配字符组中的任意一个。“-”用于表示字符范围。例如[abc]表示匹配 a 或者 b 或者 c，[a-f]表示匹配从 a 到 f 这个范围之中的任意一个字符。

6. 什么是文件系统？什么是文件？

在多数操作系统中都有文件的概念。文件是 Linux 用来存储信息的基本结构，它是由命名（称为文件名）的存储在某种介质（如磁盘、光盘和磁带等）上的一组信息的集合。Linux 系统中有三种基本的文件类型：普通文件、目录文件和设备文件。

在计算机系统中存有大量的文件，如何有效的组织与管理它们，并为用户提供一个使用方便的接口是文件系统的一大任务。Linux 系统通过目录将系统中所有的文件分级、分层组织在一起，形成了 Linux 文件系统的树型层次结构。以根目录“/”为起点，所有其他的目录都由根目录派生而来。

7. Linux 系统文件命名有什么规定？

文件名是文件的标识，它由字母、数字、下划线和句点组成的字符串来构成。Linux 要求文件名的长度不超过 255 个字符。为了便于管理和识别，用户可以把扩展名作为文件名的一部分，通常句点之后的部分为扩展名。扩展名对于将文件分类十分有用。

8. Linux 系统的目录/usr、/home、/bin、/dev、/var、/etc/中主要存放什么文件？

/usr 存放用户安装的软件，用户安装的软件一般位于此目录；/home 存放系统用户自身的数据；/bin 存放 shell 命令；/dev 存放系统设备的信息；/var 主要存放系统可变信息的内容，如日志、邮件、打印队列等；/etc 存放系统可执行文件。

9. 如何使用 cat 命令将多个文件连接起来显示？

使用命令：cat file1 file2 ... 就可以将多个文件首尾相连显示在屏幕上。

10. more 命令和 less 命令有什么区别？

more 可将文件显示在屏幕上，但它每次只显示一页。显示文件时，按空格键可显示下一页，按<q>键退出显示，<h>键给出帮助信息；

less 的作用与 more 十分相似，都可以用来浏览文本文件的内容，不同的是 less 允许使用者来回卷动以浏览已经看过的部份，同时 less 并不在一开始就读入整个文件，因此在遇上大型文件的时，会比较快。与 more 相比，less 的功能更加强大，在 less 命令执行时，可以使用的指令与 vi 编辑器的命令以及 more 的指令类似。

11. 如何用复制、删除命令实现文件的移动？

步骤如下：1) 首先使用 cp 目录将文件从原目录拷贝到目标目录；2) 在原目录中删除原文件

12. 如何使用 ls 命令查看隐藏文件信息？

使用-a 选项将显示所有文件包括隐藏文件的信息，ls 默认将名称以 “.” 开头的文件或目录视为隐藏，不会列出；

13. 如何递归地将当前目录下所有的 C 语言程序拷贝至 /home 目录中，要求如果有同名文件，则覆盖同名文件？

执行命令 cp，并使用选项-rf，-r 表示若 source 中含有目录，则递归地将目录下的文件也依序拷贝至目的地。-f 表示若目的地已经有同名文件存在，则在复制前先予以删除再进行复制。

14. 如何强制删除当前目录及其子目录下的所有 c 程序。

执行命令：rm -rf *.c

15. 给出命令，在当前工作目录下的 A 目录中，删除名为 B 的子目录。若 B 删除后，A 目录成为空目录，则 A 也予删除。

rmdir -p A/B

16. 给出查找/usr 目录下大小超过 50k 并且最近 2 小时内被更新过的文件的命令。

查找/usr 下大小超过 50k 的文件：

find /usr/ -size 50k -cmin -120

17. 如何使用命令完成把当前工作目录切换到/root 并显示是否切换成功？

首先使用 pwd 命令查看切换之前的目录，然后执行命令 cd /root；

最后使用 pwd 命令查看当前目录，看是否是/root，如果是表示切换成功。

18. 给出命令将当前目录下的文件 file 的时间记录改成 2005 年 5 月 10 日 18 点 30 分。

执行命令：touch -t 0505181830 file 或者命令 touch -d "6:30pm 05/10/2005" file

19. 如何执行 tar 命令对文件执行备份和恢复？

备份：tar cvf xx.tar * 或者 tar czvf xx.tar.gz *

恢复 tar cvf xx.tar 或者 tar czvf xx.tar.gz

20. 给出命令将当前目录下的所有 c 程序备份并为一个文件。

Tar cvf xx.tar *.c 或者 tar czvf xx.tar.gz *.c

21. 假设有一个 rpm 包的软件：software-1.2.3-1.i386.rpm，简述软件的命名含义，并说明如何安装及如何查看是否已经安装？

software-1.2.3-1 表示软件名称和版本，i386 表示该软件是为 386 体系结构机器编写编译的，rpm 表示该软件使用 rpm 软件包管理工具打包。

安装命令为：rpm -i software-1.2.3-1.i386.rpm

查看命令为：rpm -q software-1.2.3-1.i386.rpm

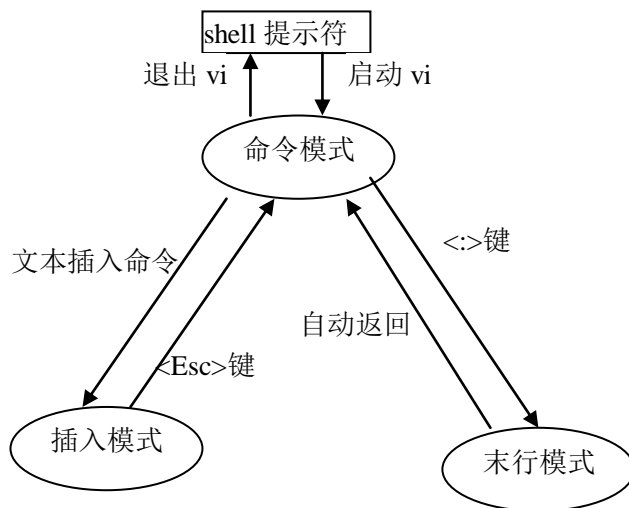
22. 下面是执行 “ls -l software” 命令得到的信息，通过联机帮助，解释这些信息：

第 1 个域表示文件的访问权限；
 第 2 个域表示硬链接的个数
 第 3 个域表示文件的所有者
 第 4 个域表示文件所属的群组
 第 5 个域表示文件大小
 第 6 个域表示文件最近一次修改的时间
 最后一个域表示软件名称。

第 3 章 vi 编辑器的使用

1. vi 有几种工作模式，各工作模式之间如何进行转化？

vi 有三种基本工作模式，分别是：命令模式（command mode）、插入模式（insert mode）和底行模式（last line mode）三种模式之间的转换如下图。



2. 在 vi 中如何保存正在编辑的文件？

在命令模式下，连按两次大写字母<Z>，若当前编辑的文件曾被修改过，则 vi 保存该文件后退出，返回到 shell；若当前编辑的文件没被修改过，则 vi 直接退出，返回到 shell。

在末行模式下，有几种命令可以保存文件：

1) :w vi 保存当前编辑的文件，但并不退出 vi，而是继续等待用户输入命令。

2) :w <newfile> 把当前文件的内容保存到指定的文件 newfile 中，而原有文件保持不变。若 newfile 是一个已存在的文件，则 vi 在窗口的状态行给出提示信息：

File exists (use ! to override)

表示指定的文件已经存在，如果要替换原有内容，需要使用“!”。

3) :w! <newfile> 把当前文件的内容保存到指定的文件 newfile 中，如果 newfile 已经存在，则覆盖原有内容。

3. 有几种方法可以退出 vi 编辑器？

在末行模式下，有四种方法可以退出 vi 返回到 shell：

1) :q 系统退出 vi 返回到 shell。在用此命令时，若编辑的文件没有被保存，则 vi 在窗口的最末行显示如下信息：

No write since last change (use ! to overrides)

提示用户该文件被修改后没有保存，需要使用！放弃保存。出现此提示后，vi 并不退

出，继续等待用户命令。

2) :q! vi 放弃所作修改而直接退到 shell 下。

3) :wq 先保存文件，然后再退出 vi 返回到 shell。

4) :x 该命令的功能与命令模式下的 ZZ 命令功能相同。

4. 如何让屏幕上显示正在编辑的文件的行号？

在末行方式下输入如下命令即可给文本加行号：

:set number 或者:set nu

5. 说明 vi 中的字、句子、段落的含义。

在 vi 中“字”有两种含义。一种是广义的字，它可以是两个空格之间的任何内容。狭义上的字是指：英文单词、标点符号和非字母字符，如“!”、“@”、“#”、“\$”、“%”、“^”、“&”、“*”、“(”、“)”、“-”、“+”、“{”、“}”、“[”、“]”、“~”、“|”、“\”、“<”、“>”、“/”等，均被当成是一个字。

句子被定义为是以句号“.”、问号“?”和感叹号“!”结尾，且其后跟着至少一个空格或一个换行符的字符序列。

段被定义为以一个空白行开始和结束的片段。

6. 练习使用 vi 中在当前行左右移动光标的命令

l 或者→、h 或者←、w、<n>w、b、<n>b、e、0、\$等命令的使用，这些命令使光标在同一行上移动。

7. 练习使用 vi 中光标上下移动的命令

练习使用命令：j 或者↓、+、k 或者↑、-

8. 练习使用 vi 中在文本块间移动光标的命令

练习使用命令：)、(、}、{

9. 在 vi 中如何将光标直接定位到某行？

在命令模式下，执行<行号> G，则将光标移至行号所指定的行的行首。

10. 练习屏幕上下移动光标和滚动屏幕的命令。

三个关于光标在全屏幕上移动并且文件本身不发生滚动的命令。它们分别是 H、M 和 L 命令；

滚屏命令：< Ctrl+u > < Ctrl+d >

分页命令：< Ctrl+f > 或者<PgDn>、< Ctrl+b > 或者<PgUp>

屏幕调零命令：

[行号] z [行数]<回车>

[行号] z [行数].

[行号] z [行数]-

11. 在命令模式下，如何删除文本中的字符、字、行和文本块？

字符删除命令：x、dh

字和行删除命令：dw、db、dd、d\$、d0、<n>dd

文本块删除命令：d)、d(、d}、d{

12. 在命令模式下，如何查找匹配某个模式的行？

/<pattern> 回车

?<pattern> 回车

13. 在命令模式下，如何修改一段文本？

vi 提供了三种修改命令，分别是 c、C 和 cc。它们修改文本的范围是由光标的当前位置和光标移动命令二者限定的。

14. 在命令模式下，练习文本的几种替换命令。

vi 提供的替换命令有取代命令：r 和 R、替换命令：s 和 S 和字替换命令：cw

15. 如何复制 n 行文本？

[n]yw: 复制从光标所在位置开始的 n 个字到缓冲区；

16. 如何移动一段文本？

在文件中移动文本通过如下三步完成：

- 1) 使用文本删除命令将要移动的文本删除；
- 2) 使用光标移动命令将光标移动到目标位置；
- 3) 使用命令“p”将刚删除的文本粘贴到目标位置。

17. 如何将连续的 n 行文本合并成一行？

命令：[n]J 表示把光标所在行与下面的 n-1 行合并为一行，如果省略 n，则表示把光标所在行与下面一行合并。

18. 说明三个命令：插入命令、附加命令、打开命令的区别。

在命令模式下，执行 i 命令，表示从光标所在位置开始插入文本，大写的 I 命令表示将光标移到当前行的行首，然后在其前插入文本；

在命令模式下，执行 a 命令，表示从光标所在位置之后追加新文本。新输入的文本放在光标之后，光标后的原文本将相应地向后移动。大写的 A 命令表示首先将把光标挪到所在行的行尾，从那里开始插入新文本

在命令模式下，执行 o 命令，vi 将在光标所在行的下面新开一行，并将光标置于该行的行首，等待输入文本。大写的 O 命令是在光标所在行的上面插入一行，并将光标置于该行的行首，等待输入文本。

第 4 章 X Window 系统的使用

1. GNOME 桌面环境包括 菜单面板、桌面区 和 窗口面板 三个区域。
2. GNOME 菜单面板包括 应用程序、位置 和 系统 三个菜单，其中 应用程序 用于启动系统安装的软件，位置 用于快速访问计算机资源，系统 用于管理和配置计算机。
3. GNOME 文件浏览器由 菜单栏、工具栏、位置栏、状态栏、侧边栏 和 浏览窗格 等组成，其中 菜单栏 和 浏览窗格 是必须有的。

4. 简述工作区的含义，Fedora 10 中默认包含多少个工作区，怎样调整工作区的个数？

Linux 中的工作区切换器为用户提供了使用多个工作区的能力，用户可以根据自己的喜好将程序在不同的工作区打开，查看所需内容时，选择相应的工作区即可，因此不必把所有运行着的应用程序都堆积在一个可视桌面区域，这样可避免所有窗口置于一个桌面时显得杂乱无章。

默认包括 4 个工作区。

可在“工作区切换器首选项”中调整工作区。

5. Linux 中如何永久删除指定文件或文件夹？

清空回收站可以将文件或目录永久删除；在删除文件或目录时按住 Shift 键，然后再删除。

6. 简述如何创建链接。

方法 1: 在文件管理器中选中该项目，单击鼠标右键，选择【创建链接】命令，则在文件管理器的窗口中出现一个图标，名字为“到...的链接”，并且带一个小箭头符号。只要将该图标拖放到桌面，就可以建立到该项目的链接。

方法 2: 选中该项目，按住鼠标中键（三键鼠标）的滚轮键，拖放到桌面，当放开鼠标中键时，会弹出【移动到此处】、【复制到此处】、【在此处创建链接】等命令，选择【在此处创建链接】命令，就可以在桌面创建该项目的链接。

7. 比较 X Window 系统与 Microsoft Windows 系统之间的异同。

X Window 与微软公司的 Windows 图形界面不同, X Window 不是系统内核的必备部分, 而是内核之上的一个应用, 是一个用户与系统交互的界面。用户可以根据个人爱好, 像选择中意的媒体播放软件一样选择习惯的界面系统。而且, X Window 具有如下特点:

- 良好的网络支持: X Window 采用了 C/S 网络结构, X Client 和 X Server 可以通过网络来通信, 而且有良好的网络透明性。这样, 复杂的图形桌面可以显示在维护良好、功能强大、易于管理的服务器上, 用起来非常方便。
- 个性化的窗口界面: X Window 并未对窗口界面作统一的规范, 程序员可以根据需求自行设计, 其中最有名的就是后面将要介绍的 GNOME 与 KDE。
- 不内嵌于操作系统: X Window 只定义了一个标准, 而不属于某个操作系统, 因此可在不同的操作系统上运行相同的 X Window 软件。

8. X Window 系统由哪些部分组成, 每部分的作用是什么?

整个 X Window 由三个部分组成:

- X Server: 控制输入、输出设备并维护相关资源的程序, 它接收输入设备的信息, 并将其传给 X Client, 而将 X Client 传来的信息输出到屏幕上。不同的显卡需要选择不同的 X Server, 在配置 X Window 时最主要的工作就是配置 X Server。
- X Client: 应用程序的核心部分, 它与硬件无关, 每个应用程序就是一个 X Client。X Client 可以是终端仿真器 (Xterm) 或图形界面程序, 它不直接对显示器绘制或者操作图形, 而是与 X Server 通信, 由 X Server 控制显示。
- X protocol: X Client 与 X Server 之间的通信协议。X 协议支持网络, 因此 X Client 和 X Server 既可以在同一台计算机上, 也可以运行在不同的机器上。X 支持的网络协议有 TCP/IP、DECnet 等。

9. 什么是虚拟桌面?

通常用户在屏幕上见到的只是整个桌面的一部分。把桌面分成不同的工作区, 一个桌面可以有最多 36 个工作区, 默认是 4 个。虚拟桌面包括所有的工作区以及显示在它们上面的内容, 例如, 图标、菜单和窗口等。使用工作区的好处是: 用户可以根据自己的喜好将程序在不同的工作区打开, 查看所需内容时, 选择相应的工作区即可, 这样可避免所有窗口置于一个桌面时显得杂乱无章。

10. X Window 下的终端窗口与控制台有什么区别?

就像在 windows 界面下可以启动 DOS 窗口一样, 在 X Window 图形操作界面上也可以启动控制台界面, 从而可以同时拥有这两种操作界面。终端窗口向用户提供了一个标准的命令行操作界面。用户可以在此窗口中的命令行提示符处输入命令及其参数, 命令行执行的结果显示在该终端窗口上, 命令执行完后会又出现 shell 提示符, 可以继续输入命令。可以从终端窗口启动任何 X 程序, 这是终端窗口与控制台之间的重要区别。

11. 什么是风格?

风格影响着用户桌面元素的外观, 提供不同的背景图象、动画和动作音效。

12. GNOME 提供了几种面板, 它们的特点是什么, 如何设置它们的属性?

GNOME 提供菜单面板、桌面区和窗口面板。其设置和使用参见教材 4.2.3

13. 练习设置屏幕保护程序。

略。

14. 练习 GNOME 的文件浏览器的使用。

略。

15. 如何更改 GNOME 的文件图标?

方法 1: 在属性中修改。

方法 2: 通过侧边栏将徽标拖动到文件上。

方法 3: 执行菜单【编辑/背景和徽标】命令, 将该窗口中的徽标拖放到文件上, 即可在该文件上方出现该徽标。

第 5 章 Linux 系统的常用软件

1. 套件及功能如下:

Writer: 文档处理;

Calc: 电子表格编辑;

Impress: 演示文稿、幻灯片编辑;

Base: 数据库功能;

Draw: 图形、图片编辑。

2. Writer。

3. Impress。

4. Calc。

5. 【文件/打开】

6. 【文件/另存为】, 然后设置【保存类型】。

7. 需要: 用户名、电子邮件地址、电子邮件接收服务器地址、电子邮件接收服务器身份验证信息、电子邮件发送服务器、电子邮件发送服务器身份验证信息和其他选项等。

8. (1) 浏览、下载历史; (2) 表单和搜索历史; (3) Cookies; (4) 高速缓存; (5) 活动的已登录的会话; (6) 站点设置;

9. 略。

10. 略。

11. 略。

12. 略。

13. 略。

第 6 章 硬件管理

1. 如果要使用 USB 存储设备、光盘或软盘等存储设备, 必须将这些设备中的“小”目录树像嫁接一样挂载 (Mount) 到 Linux 系统的“大”目录树中。当存储设备挂载成功之后, 就可以将其作为“大”目录树中的一个目录来进行访问了。使用完毕, 还有将这些设备从目录树中卸载下来。

2. 挂载磁盘设备主要使用 mount 命令。在挂载设备前, 首先要确定下列三种信息: (1) 挂载对象的文件系统类型; (2) 挂载对象的设备名称; (3) 设备挂载到哪个目录, 即挂载点。

3. Linux 支持的常见的文件系统有: (1) vfat, 对应于 Windows 的 Fat16 和 Fat32 文件系统; (2) ext3: Linux 下的文件系统; (3) iso9660: 光盘文件系统。

4. 在 Linux 系统中, 设备名称通常都在/dev 目录下, 设备名称的命名要遵循一定的规则。例如, /dev/hda1 是一个设备名称, 其中“hd”即硬盘 (Hard Disk), 代表 IDE 设备, “sd”代表 SCSI 设备 (SCSI Drive)。

5. 如果挂载的设备使用的是 Windows 文件系统, 则使用参数 cp936, 这样能保证正常显示中文字符, 命令如下:

```
mount -o iocharset=cp936 /dev/sda1 /mnt/usb
```

6. 可以使用 `fdisk -l` 命令或 `cat /proc/partitions` 命令查看系统中挂载的设备。
7. 用户可以执行卸载命令卸载设备，卸载命令格式为：`umount [挂载点或设备名]`。
8. 挂载：
 软驱：`mount -t vfat /dev/floppy /mnt/floppy`
 光驱：`mount -t iso9660 /dev/cdrom /mnt/cdrom`
 卸载：
 软驱：`umount /dev/floppy`
 光驱：`umount /dev/cdrom`
9. 当一台机器中同时安装了 Linux 和 Windows 两个操作系统时，用户可以修改 `/etc/fstab` 文件，这样在启动 Linux 系统时可以自动挂载 Windows 系统的硬盘分区，从而实现系统间的数据共享。
10. 略。
11. 可以使用格式化命令创建文件系统，格式化命令 `mkfs` 的一般格式为：
 `mkfs [选项] [-t <文件系统类型>] [设备名称] [区块数]`
12. 略。
13. 执行【系统/硬件/屏幕分辨率】菜单项，或者执行终端命令“`gnome-display-properties`”，可以启动监视器分辨率设置对话框，通过该对话框可以实现调整功能。
14. 可以通过到特定显卡（芯片）厂商的网站下载 Linux 平台下的特定驱动程序。
15. （1）下载驱动程序；（2）安装驱动程序；（3）设置声卡参数；
16. 执行菜单【系统/首选项/硬件/音效】或者执行终端命令“`gnome-sound-properties`”，可以打开声卡的配置与检测工具。
17. 首先检查系统是否可以自动识别打印机设备，如果不识别可以到打印机厂商的网站根据设备型号下载 Linux 平台下的驱动程序，安装并设置相应参数。
18. 可以使用打印机驱动附带的打印测试功能。
19. 首先看该设备系统是否能自动识别，如果不能寻找设备附带的 Linux 平台驱动程序或者到相关网站下载安装。
20. 略。

第 7 章 网络基本配置

1. 什么是网络接口设备？请举例说明。
把计算机连接到网络的设备称为网络接口设备，例如：以太网卡、无线网卡和调制解调器等。
2. 什么是设备别名？请举例说明。
设备别名就是给设备取的另一个名字，当说明某个设备时，别名与原名都指同一个物理设备。一般使用别名是为了系统管理方便。例如第 1 个以太网卡名字为 `eth0`，可以为它取多个别名，`eth0:0`、`eth0:1`……，它们都是指的第 1 个以太网卡，但可以为不同的别名设置不同的网络属性，例如，设置不同的 IP 地址。
3. 有哪几种方法配置网络接口设备？
可以使用三种不同的工具来配置网络接口设备：使用网络接口配置程序 `netconfig`、使用图形配置工具、使用终端命令 `ifconfig`。
4. 假设在启动系统时，系统报错提示你本机的 IP 地址已被占用，说明错误原因，并给出解决方法。
发生这种情况，一定是由于在同一个网络上有某个系统与你的机器使用了相同的 IP 地址，并且该机器已经启动，你就不能再使用这个 IP 了。

这时，或者更改另一个系统的 IP，然后再重新启动本机的网络；或者将自己的 IP 改为其它的与别人不冲突的地址，然后再重新启动网络。否则网络不可用。

5. 在不启动 X Window 的情况下，如何配置网络接口设备的 IP 地址？

有两种方法：

- 1) 在终端中输入命令 `netconfig`，打开 TCP/IP 配置界面。用户可以选择是否使用动态 IP，可以设置网络接口的 IP 地址、子网掩码、默认网关、主域名服务器。
- 2) 使用 `ifconfig` 网络接口配置命令。

命令使用格式为：`ifconfig <设备名> <IP 地址> netmask <掩码>`

6. 有几种方法控制网络接口的启动与禁止？请举例说明。

- 1) 在网络配置的图形界面中，通过“激活”或者“解除”按钮可以启动或者禁止网络接口；
- 2) 使用网络控制程序 `network: /etc/rc.d/init.d/network start|stop|restart`
- 3) 使用命令 `ifconfig: ifconfig <设备名> up|down`
- 4) 使用命令 `ifup/ifdown: ifup/ifdown <设备名>`
- 5) 在 x Window 下，执行【系统/管理/网络】，打开“网络设备控制”窗口，可以通过【激活】或者【取消激活】按钮来启动或者禁止相应接口。

7. 怎样快捷地查看本机的网络接口配置信息？

可以使用终端命令 `ifconfig` 方便地查看系统目前所有活跃的网络接口的详细信息，
`ifconfig [设备名]`

8. 如何判断本机网络的连通性能？

可以使用 `Ping` 命令或者 `traceroute` 命令检测网络的连通性。

步骤 1：测试本机网络操作系统是否工作正常

命令：`ping 127.0.0.1` 或 `ping localhost`

步骤 2：测试本机网络接口，即本机的 IP 地址

`ping 192.168.14.11`

如果返回正常信息，则表示本机网卡正常。

步骤 3：测试本地局域网中网关或其它主机

`ping 192.168.14.254` 网关

`ping 192.168.14.15` 同一个网络的其它主机

步骤 4：测试远程网络中的网关或其它主机

`ping 192.168.214.254` 测试与远程局域网网关的连通性

`ping 192.168.214.45` 与远程主机的连通性

步骤 5：测试 DNS 服务器

`ping abc.def.com` 使用域名访问其它主机

需要注意的是：如果主机安装了防火墙等安全工具，可能会阻止外部主机对系统使用 `ping` 命令。所以，以上测试过程中 `ping` 其它主机时，可能返回错误信息“Destination host unreachable”。这可能是目标主机安装了防火墙的原因。

9. 如何判断同网中的某台主机是否在网络中？

使用 `ping` 命令：`ping 主机地址`

注意目标主机不能安装防火墙。

10. 有什么方法了解本机与另一台主机之间有哪些路由器？

使用 `traceroute` 命令：`traceroute 目的主机 IP 或者域名`

11. 怎样在本机中远程登录另一台主机？

使用 `telnet` 命令：`telnet 目标主机地址或域名`

在给出提示信息时分别输入用户名和口令

12. 如果一台主机 A 中安装了一个网卡，但它需要使用三个 IP 地址，并分别对应使用三个不同的主机名。如果可以使主机 B 能够按不同的主机名访问不同的 IP 地址，如何配置两台主机？

在主机 A 中：使用设备 eth0 和两个别名 eth0:0 eth0:1，分别为这三个设备配置 IP 地址；

在主机 B 中：配置/etc/hosts 文件，使得上述三个 IP 地址与相应的主机名对应；

例如：192.168.14.15 qq.yys.com qq

定义/etc/host.conf 文件，使得系统首先访问/etc/hosts 文件

13. 哪个 TCP/IP 配置文件中包含了 TCP/IP 各种协议的名称？

文件/etc/protocols 定义了使用的网络互联协议及协议号

14. 如何知道 Linux 系统能够提供哪些网络服务？

通过查看/etc/service 文件可以得知 Linux 系统能提供的网络服务，但仍需安装配置相应的服务软件。

15. 要根据主机名获得对应的 IP 地址，可能涉及哪些 TCP/IP 配置文件？

1) /etc/hosts 文件中包含了主机名与 IP 地址以及别名的对应；

3) 文件/etc/resolv.conf 保存了 DNS 服务器的 IP 地址，以及查询这些服务器的顺序。

2) /etc/host.conf 文件中指出是先访问/etc/hosts 文件，还是先访问域名服务器；

16. Linux 可以设置静态路由表，如何查看路由表信息？

不加任何参数的 route 命令显示本机路由表的内容

17. 邻近的同事设置其主机名称为“wang.yys.com”，你如何在网络命令中使用这个名称？

- 1) 首先在/etc/hosts 中添加一条记录，将该主机名与相应的 IP 地址对应：

例如：192.168.14.15 wang.yys.com

- 2) 设置/etc/host.conf 文件使其先访问/etc/hosts 文件，后访问域名服务器：

order hosts,bind

18. 如何决定查询一个域名时，使用本机主机名列表文件与 DNS 服务器的顺序？

/etc/host.conf 文件决定了在查询域名时使用本机主机名列表文件与 DNS 服务器的顺序，order hosts,bind 表示首先使用本机主机名列表文件，order bind,hosts 表示首先使用域名服务器。

19. 用户怎样才能从本机信任地访问另一台远程主机？

Linux 系统可以设置信任主机关系，如果 B 信任 A 主机，则来自 A 主机的远程登录请求可以不用提供口令。

例如：在 B 主机的/etc/hosts.equiv 文件中设置一行：

a.yys.com testuser

表示信任来自 a.yys.com 主机中登录名为 testuser 的连接请求，从而不需要检查它的口令。

/etc/hosts.equiv 文件是设置等价主机的配置文件，其格式：

<信任的主机> <信任的用户>

其中后者可以省略，表示信任该主机上的所有用户。这样只要提供 B 主机中存在的登录用户名，就不必检查其口令。

20. 安装并配置一台 Linux 系统的 telnet 服务器，并在其它主机中使用终端登录连接该主机。

步骤 1：检查/etc/services 文件是否启用了关于 telnet 服务的设置

telnet 23/tcp

telnet 23/udp

步骤 2: 在/etc/rc.d/xinetd.d/目录下建立 telnet 服务配置文件并配置相应内容:

```
service telnet
{
    disable = no
    flags      = REUSE
    socket_type = stream
    wait       = no
    user       = root
    server     = /usr/sbin/in.telnetd
    log_on_failure += USERID
}
```

步骤 3: 重新启动 xinetd 服务

21. 当用户使用一个主机名时, 系统怎样才能最快地查找到其对应的 IP 地址?
 - 1) 首先在/etc/hosts 中添加一条记录, 将该主机名与相应的 IP 地址对应;
 - 2) 设置/etc/host.conf 文件使其先访问/etc/hosts 文件, 后访问域名服务器:
22. 为了方便用户记忆, 主机域名可以表示主机提供的网络服务, 例如: www.yys.com 与 ftp.yys.com 可以分别表示提供 WWW 服务与 FTP 服务的主机, 它们可以是不同主机, 也可以是同一台主机。如果在 IP 地址为 192.168.1.1 的 Linux 主机 A 中提供这两种网络服务, 如何配置才能使客户机 B 能够按主机域名获取 A 主机提供的服务?
 - 1) 首先在/etc/hosts 中添加两条记录, 分别将两个主机名与 IP 地址对应;
 - 2) 设置/etc/host.conf 文件使其先访问/etc/hosts 文件, 后访问域名服务器:
order hosts, bind
 - 3) 允许一个主机名对应多个 IP 地址, 在/etc/host.conf 文件中增加一行:
multi on

第 8 章 常用网络服务的配置和使用

1. Linux 系统提供哪些常用的网络服务?

Linux 作为一个网络操作系统可以提供各种网络服务举例: Apache web 服务器; FTP 服务器; Bind DNS 服务器; Sendmail 邮件服务器; Samba 服务器等。
2. 用户如何设置在 Linux 系统开机自动启动某个网络服务?
 - 1) 在 X Window 下执行【主菜单/系统配置/服务器设置/服务】, 进入服务配置窗口。选中对应服务, 用鼠标选择其选择框, 然后执行菜单【文件/保存改变】。这样 Linux 系统将在下一次启动时自动启动该服务。
 - 2) 在终端中输入命令 ntsysv 即可进入文本界面的服务管理工具。该工具的功能是设置在 Linux 系统启动时自动运行哪些服务。
 - 3) chkconfig [--level 运行级] 服务名 [状态]。其中服务通常只能运行在 3、4、5 级中。状态可以是 on (启动)、off (停止) 或者 reset (重置) 三种状态之一。
 - 4) 使用 xinetd 控制
3. 8.1.1 与 8.1.2 介绍的管理工具在功能上有什么不同?

X Window 下的服务设置程序有三个功能:

 - 1) 即时启动/停止/重新启动某个服务
 - 2) 在系统启动时自动运行某个服务, 这样 Linux 系统将在下一次启动时自动启动该服务。

3) 编辑系统运行级别。

ntsysv: 功能是设置在 Linux 系统启动时自动运行哪些网络服务。

4. 如何查看当前系统运行了哪些服务进程?

在 X Window 下执行【系统/管理/服务】，进入服务配置窗口，该窗口左边窗格中列出了系统中安装的网络服务。凡是复选框中被选了的，就是当前运行的服务。

在终端中输入命令 ntsysv 即可进入文本界面的服务管理工具，那些前面有*的就是启动了的服務。

管理员可以通过查看当前的进程树命令 pstree 来获知系统正在运行哪些服务，命令格式为: pstree

5. 什么是守护程序?

Linux 系统的服务程序一般由运行在后台的守护程序 (daemon) 来执行的。服务的守护程序在开机时自动运行，时刻监听客户的请求，随时为客户提供服务。这类程序除非主动退出或人为终止，否则它们会运行至关机为止。

6. 什么是 xinetd?

xinetd 是网络守护进程服务程序 (又称超级服务器)，它可以为所有的服务提供守护程序的功能。当收到系统的服务请求时，xinetd 会自动将请求转给对应的服务程序。

7. xinetd 与一般的守护程序有什么不同?

一般的守护程序只监听一个自己的端口，对到达该端口的服务请求进行服务。而 xinetd 是超级服务器，它可以替多个程序监听多个端口，当有请求到达某个端口时，它负责将请求转交给相应的服务程序。

凡是由 xinetd 控制的服务，xinetd 都控制了所有到达的访问请求，因而可以让 xinetd 对访问进行控制。比如，控制访问的流量、访问来源等。

使用 xinetd 固然可以节省资源，但对于服务量很大的守护程序 (如 HTTP 服务、FTP 服务) 将影响服务性能。所以，对于这类服务程序通常直接运行自己的守护程序。

8. Apache 服务器有哪些功能?

支持最新的 HTTP/1.1 协议。支持通用网关接口 (CGI)，支持虚拟主机，支持 HTTP 认证，集成的代理服务器，支持服务器端包含命令 (SSI)，支持安全 SOCKET 层 (SSL)，简单而且强有力的基于文件的配置 (HTTPD.CONF)。

9. 如何安装最新的 Apache 服务器?

1) 获取最新 Apache 软件: 到 Apache HTTP Server Project 的网站 (<http://httpd.apache.org/>) 或它的镜像站点上直接下载压缩的源代码文件。

2) 将源代码解压缩

3) 执行自动配置: `./configure --prefix=/home/myapache`

4) 编译: `make`

5) 执行安装命令: `make install`

10. 如何使用 Apache 创建一个个人 web 网站。

1) 在作为网站的主机上安装 Linux 系统，并安装 Apache 服务器。

2) 部署主页文件: 使用各种网页设计工具，如 Windows 系统下的 FrontPage、Dreamweaver 等，以及各种脚本语言，如 perl、php 等制作好个人网页，建议首页的文件名为 index.htm 或 index.html。然后，将网页文件复制到 Linux 系统中的 Apache 服务器指定的根文档目录中。

3) 修改 Apache 的配置

修改配置文件 `/etc/httpd/conf/httpd.conf` 来修改 Apache 的配置。

4) 重新启动 Apache 服务器，使新的配置生效。

在客户机中，打开浏览器，输入地址 `http://<Apache 服务器的 IP 或主机名>`

11. 有几种方法控制 Apache 的启动与关闭？

在 Linux 系统中，Apache 服务的守护程序名称是 `httpd`。所以，启动系统中的 `httpd` 程序就是启动 Apache 服务器。

- 1) 在 X Window 下执行【系统/管理/服务】，进入服务配置窗口，可以在这里即时启动 `httpd`，也可以指定在某个运行级别运行 `httpd`，从而启动到该级别时就可以启动 `apache`。
- 2) 在终端中输入命令 `ntsysv` 即可进入文本界面的服务管理工具，可以设置在 Linux 系统启动时自动运行哪些服务
- 3) `service 服务名 [start|stop|restart]`
- 4) 对于 Apache 服务器的启动/关闭/重启操作，可以使用 `xinetd` 管理，需要设置 `/etc/xinetd.d/httpd` 配置文件。由于 Web 服务的访问量很大，会使 `xinetd` 程序占用很多系统资源，从而无法满足其它服务的要求，而且会影响 Apache 服务器的性能。所以，建议使用前者服务管理方法。
- 5) 可以直接操作系统中安装的 Apache 程序。在终端中使用命令：
`/etc/rc.d/init.d/httpd [start|stop|restart]` 可以启动、停止或者重新启动 Apache 服务。

12. 如何改变 Apache 服务器的监听端口？

HTTP 服务默认端口为 80。修改 Apache 配置文件 `httpd.conf`：

`listen: <端口号>` 指定 Apache 在哪个 TCP 端口进行侦听

13. 怎样在 Apache 服务器中使用 SSL 功能？

在 Linux 系统中附带 OpenSSL 工具箱软件来实现 SSL 通信，用户也可以到网站上下载 OpenSSL 工具箱软件的升级包进行安装。Apache 服务器需要通过加载 SSL 模块 `mod_ssl` 来调用 OpenSSL 进行 SSL 通信。

首先验证 Linux 系统是否已安装 `mod_ssl`，如果没有安装，用户可以从网络中下载相应的安装包进行安装。

基本的 SSL Apache，就是 Apache 服务器需要向用户证明自己就是用户访问的那个服务的提供者，这种情况下需要一个私钥和一个由此私钥签署的证书，在 SSL 配置文件中指定私钥和证书文件的位置之后就可以启动 SSL Apache 了；双向的 SSL 认证，在要求服务器端认证的同时还会要求客户端认证，要求用户出示自己的客户端 SSL 证书。

14. 什么是 FTP？

FTP (File Transfer Protocol) 是一种文件传输协议，FTP 服务是网络中最为常用的服务之一。它实现了服务器与客户机之间的文件传输和资源再分配，是资源共享的方式之一。

15. vsFtpd 有哪些功能？

vsFTPd (very secure FTP daemon) 是一个功能强大的 FTP 服务器，能运行在大部分 UNIX 类操作系统上，具有简洁性、安全性、快速性的特点，支持很多其他的 FTP 服务器不支持的特征，例如：

- 支持虚拟 IP
- 支持虚拟用户
- 可以独立操作或者由 `xinetd` 管理
- 可以对每个用户进行配置
- 带宽限制
- 支持 IPv6
- 支持通过 SSL 的加密

● 高速

16. 在一台 Windows 主机中如何连接一台 FTP 服务器？

可以使用网络浏览器和终端命令等方式：

打开系统的网络浏览器，在地址栏中输入 `ftp://<FTP 服务器的 IP 地址或主机名>`，该命令表示连接使用 FTP 协议。一旦连接成功，就可以直接使用图形界面操作，非常方便。

如果是以本地用户身份访问，则输入命令：`ftp://<本地用户名>@<FTP 服务器的 IP 地址或主机名>`，系统若连接成功，则会弹出一个对话框，要求用户输入口令。

在系统终端中，执行下述命令连接 FTP 服务器：

```
ftp <FTP 服务器的 IP 地址或主机名>
```

一旦连接成功，系统会要求用户输入用户名和口令。如果用户是以匿名方式登录，则默认的用户名是 `anonymous` 或 `ftp`，口令则是空的或是任意字符串，有些系统要求匿名口令为一个电子邮件地址。

进入系统后，会出现 FTP 系统的命令提示符“`ftp>`”，在该提示符后可以输入各种 FTP 命令，进行相关操作。

17. 如何在 Windows 与 Linux 系统主机之间共享文件夹？

主机之间共享文件可以使用 `ftp`、`telnet` 等，在 Linux 上构架 SMB 服务也可实现 Windows 与 Linux 系统主机之间共享文件，这是一种简单快捷的方式。SMB (Server Messages Block, 信息服务块) 是一种在局域网上共享文件和打印机的一种通信协议，它为局域网内的不同计算机之间提供文件及打印机等资源的共享服务。SMB 协议是客户机/服务器型协议，客户机通过该协议可以访问服务器上的共享文件系统、打印机及其他资源。通过设置“NetBIOS over TCP/IP”使得 Samba 不但能与局域网络主机分享资源，还能与全世界的电脑分享资源。

18. 什么是 SMB？为什么使用 samba 软件？

SMB (Server Messages Block, 信息服务块) 是一种在局域网上共享文件和打印机的一种通信协议，它为局域网内的不同计算机之间提供文件及打印机等资源的共享服务。SMB 协议是客户机/服务器型协议，客户机通过该协议可以访问服务器上的共享文件系统、打印机及其他资源。通过设置“NetBIOS over TCP/IP”使得 Samba 不但能与局域网络主机分享资源，还能与全世界的电脑分享资源。

在 Linux 上运行 SMB 的软件很多，最常用的是 Samba。Samba 属于 GPL 软件，能运行在 Linux(unix) 环境下。

19. 结合有关编程语言，开发一个简单的 Apache 服务器。

略。

20. 构建一台服务器，提供多种服务，包括：WWW 服务、FTP 服务、网络共享服务。

略。

21. 简单说明 DNS 的工作原理。

DNS 是一个树状的分布式数据库，本地负责控制整个分布式数据库的部分段，每一段中的数据通过客户/服务器模式在整个网络上均可存取。在 DNS 中，每个域分别由不同的组织进行管理。每个组织都可以将它的域再分成一定数量的子域并将这些子域委托给其他组织进行管理，域既包括主机又能包括它的子域。域名被用做 DNS 数据库中的索引，域包含所有域名在该域的主机。

域名解析的工作原理主要由以下几步实现：

1. 客户机将域名查询请求发送到本地 DNS 服务器，服务器在本地数据库中查找客户机要求的映射。

2. 如果不能在本地找到客户机查询的信息，将客户机请求发送到根域名服务器。根域名服务器负责解析客户机请求的根域部分，它将包含下一级域名信息的服务器的地址返回给

客户机的 DNS 服务器。

3. 客户机的 DNS 服务器利用根域名服务器解析的地址访问下一级 DNS 服务器，得到维护再下一级域名的 DNS 服务器的地址。

4. 按照上述方法递归地逐级接近查找目标，最后在维护目标域名的 DNS 服务器上找到相应的 IP 地址信息。

5. 客户机的本地 DNS 服务器将查询结果返回客户机。

6. 客户机利用从本地 DNS 服务器查询得到的 IP 地址访问目标主机。

22. 设计一个网络服务器群，配置一台 DNS 服务器。

略。

第9章 系统管理与监控

1. 什么是用户账号？Linux 系统的账号是如何分类？

Linux 系统是多用户的网络操作系统，用户在使用系统时，必须标明自己的身份，这就是用户账号，为了证明自己的身份，每个账号还需要一个口令。

Linux 系统的账号有用户账号和组账号两类：

用户账号：通常一个操作者拥有一个用户账号，每个用户账号有唯一的识别号 UID (User ID) 和自己所属组的识别号 GID (Group ID)。

组账号：是一组用户账号的集合。通过使用组账号，可以设置使一组用户对文件具有相同的权限。管理员在以组为单位对资源的访问权限进行配置时，可以节省日常维护时间。

2. 什么是 UID 与 GID？Linux 系统管理员创建用户的 UID 有什么特点？

每个用户账号有唯一的识别号即 UID (User ID)，它是一个整数。系统自动建立的用户 ID 小于 500，新建的第一个用户 ID 为 500，以后依次递增。其中 root 账号的 ID 为 0，具有超级用户权限。

每个组有唯一的识别号 GID (Group ID)。系统生成的组 ID 小于 500，管理员新建的第一个组 ID 为 500，以后依次递增。

3. Linux 系统中如何保存用户口令信息？

以前版本的 Linux 把用户口令以加密的形式存储在文件 /etc/passwd 中。由于该文件对任何人都可读，因而存在安全隐患。后来发布的 Linux 系统都使用了隐藏 (Shadow) 套件来加强口令文件的安全性。口令隐藏技术可以使 /etc/passwd 仍保持可读性，但不再包含口令，用户的口令被加密后保存在 /etc/shadow 中。默认安装的 Fedora 10 使用 Shadow 机制，并且用 SHA-512 散列算法对口令加密

4. 在创建用户时，如何指定其主目录？

在 X Window 中，执行【系统/管理/用户和组群】，或在终端中使用“redhat-config-users”命令，打开用户管理器，单击添加用户按钮，弹出创建新用户对话框。选择“创建主目录”，在“主目录”后面的输入框中可以指定该用户的主目录。

5. 如何查看当前系统登录的所有用户？

使用命令 who 或者命令 w

查看当前在线的用户情况。

6. 如何由超级用户改变为普通用户身份？

使用命令 su 可以临时改变用户的身份：

例如，执行命令 su lhy

就可以将身份切换为 lhy 了。

7. Linux 系统如何保存用户的家庭电话等个人信息？

在新建一个用户时，通常没有输入用户信息。可以使用 chfn 命令，系统会出现相关提

示信息要求操作者录入用户的信息，录入完毕后，系统会将这些信息保存到文件 `/etc/passwd` 中与用户对应的行中，并以逗号分隔。

命令格式：`chfn [用户名]`

如果不指定用户名则修改自己的用户信息。

8. 如何删除一个用户及其所建立的文件？

删除用户的命令为 `userdel`，该命令的格式为：

`userdel <用户名>`

但这个命令仅删除账号，而该账号创建的文件还会保留在系统中。如果系统不要保存这些文件，可以使用带选项的命令：`userdel -r <用户名>`

9. 结合一个单位的实际情况，在一台 Linux 主机中为每个成员开设一个账号，并建立一些公共资源和私有资源，设计一个方案对账号和资源进行管理。

- 1) 分析不同资源的性质，是公共的还是私有的，不同人员对它的访问权限如何；
- 2) 根据访问权限将用户分组；
- 3) 创建用户组；
- 4) 创建用户，并将其放入适当的组；
- 5) 使用 `chmod` 设置资源的访问权限。

10. 什么是进程，进程与程序有何区别？

Linux 系统上所有运行的东西都可以称之为一个进程。每个用户任务、每个系统管理任务，都可以称之为进程。进程是一个程序的运行。进程与程序是有区别的，进程不是程序，虽然它由程序产生。程序只是一个静态的指令集合，不占系统的运行资源；而进程是一个随时都可能发生变化的、动态的、使用系统运行资源的程序。一个程序可以启动多个进程。

11. 分析交互式进程与守护进程的区别。

- 交互进程——由 shell 启动的进程，用户发布一个命令后系统执行，执行完后显示结果，等待用户再次输入新的命令；
- 守护进程——在后台持续运行的进程，不接受用户从控制台发布的命令，按照固定的方式运行

12. 分析前台进程与后台进程的区别。

所谓前台，是指一个进程控制着标准输入和输出。在程序执行时，shell 暂时挂起，程序执行完毕后回到 shell。前台进程运行时，在同一个控制台上用户不能再执行其他的程序。所谓后台进程，是指一个程序不从标准输入接受输入，一般也不将结果输出到标准输出上。一些运行时间较长、运行之后不需要用户干预的程序适合运行在后台。

13. 如何手工启动一个进程？如何中止一个后台进程？

手工启动进程又可以分为前台进程和后台进程。

1) 前台启动：这是手工启动一个进程的最常用的方式。一般地，用户键入一个命令，例如 `ls -l`，就已经启动了一个进程，而且是一个前台的进程。

2) 后台启动：如果再命令后面加一个 `&` 符号，就从后台启动了一个进程

对于一个后台进程而言，使用组合键 `Ctrl+c` 无效，必须使用 `kill` 命令。首先应该使用 `ps` 命令来查看该进程对应的 `pid`，假如该进程的 `pid` 为 345，然后使用 `kill` 命令来终止该进程。输入命令：`kill 345`

有时候可能会遇到这样的情况，某个进程已经挂起或闲置，但是使用 `kill` 命令却杀不掉。这时候就必须发送信号 9，强行关闭此进程：`kill -s 9 345`。但这种“强制”方法很可能会导致打开的文件出现错误或者数据丢失之类的错误，所以不到万不得已不要使用强制结束的办法。如果连信号 9 都不响应，就只能重新启动计算机了

14. 如何使用 at 命令调度进程？

at 命令用于在指定时刻执行指定的命令序列。输入 at 执行的命令有两种方法：

- 在 shell 提示符下输入“at 时间”，然后按回车键。这时在下一行 shell 会等待用户继续输入要执行的命令。每一行输入一个命令，所有命令都输入完毕后按 Ctrl+d 键结束。
- 将各个命令写入 shell 脚本中，然后使用下面格式设置在指定时间执行 shell 脚本中的命令：

at 时间 -f 脚本文件。

15. 如何调度系统在指定时间执行指定命令。

使用 at 命令让系统再指定时间执行指定命令

使用 cron 程序调度系统在指定时间执行指定命令

16. 说明 cron 程序的工作原理。

cron 命令在系统启动时由一个 shell 脚本自动启动，进入后台。

cron 启动后搜索/var/spool/cron 目录，寻找以/etc/passwd 文件中的用户名命名的 crontab 文件，被找到的这种文件将载入内存。如果没有 crontab 文件，就转入“休眠”状态，释放系统资源。cron 每分钟“醒”过来一次，查看当前是否有需要运行的命令。如果发现某个用户设置了 crontab 文件，它将以该用户的身份去运行文件中指定的命令。命令执行结束后，任何输出都将作为邮件发送给 crontab 的所有者，或者/etc/crontab 文件中 MAILTO 环境变量中指定的用户。

17. 如何查看系统当前运行的进程。

查看目前进程状况：ps

选项组合 aux 可以显示最详细的进程情况：ps aux

18. 有哪些方法可以停止一个进程。

如果是前台进程，使用 Ctrl+C 即可中止进程，如果是后台进程，则必须使用 kill 命令。

19. 如何显示当前内存用量？

1) 使用 top 命令可以显示内存利用率；

2) free 命令可以显示系统的物理内存、内核缓冲区、cache 和交换区的总量以及已经使用的、空闲的、共享的、在内核缓冲区的和被缓存的内存数量。

3) 使用图形化的系统监视器

20. 使用什么命令可以得到系统磁盘空间用量。

df 命令可以报告系统的磁盘空间用量，包括各个分区的磁盘总量、已用的空间、未用的空间、已用的百分比、挂载点。如果要以 MB 和 GB 为单位，则需要使用“df -h”命令

21. 试分析不同日志文件的内容和功能。

在 X Window 下，执行【应用程序/系统工具/系统日志】命令，可以打开日志查看器。选中某个日志文件，可以再右侧显示该日志的内容和功能。

可以使用文本编辑器打开日志文件，查看各个域的内容和格式。在图形化的日志查看器上，可以查看日志的日志的内容。

日志内容包括：日期时间，主机，事件。

第 10 章 Linux 系统的安全管理

1. 标准 Linux 系统对文件权限是如何划分的？用户访问时都有哪些限制？

在 Linux 中，将文件访问权限分为 3 类用户来进行设置：文件所有者 (u)、同组用户 (g) 和其他用户 (o)，可分别为这 3 类用户设置不同的操作权限。Linux 系统对每一个文件

设定四种访问权限：可读取（r, readable）、可写入（w, writable）、可执行（x, eXecute）和无权限（-）。系统根据文件的权限属性来判断是否允许用户进行相关的操作。

2. 系统怎样分区才能更安全？

为了防止缓冲区溢出攻击，在安装系统时就应该注意，如果用“/”分区记录数据，如日志（log）文件和邮件（email），就可能因为大量的日志或垃圾邮件而导致系统崩溃。所以建议为“/var”目录开辟单独的分区，用来存放日志和邮件，以避免“/”分区被溢出。最好为特殊的应用程序也单独开一个分区，特别是可以产生大量日志的程序，如 web 服务其等。建议为“/home”目录单独分一个区，这样攻击者就不能通过溢出某个用户的存储空间而填满整个“/”分区，从而就避免了部分针对 Linux 分区溢出的恶意攻击。

3. 设置一个安全的用户口令有哪些要求？

口令是系统中认证用户的主要手段，系统安装时默认的口令最小长度通常为 5，为保证口令不易被猜测攻击，可增加口令的最小长度，至少等于 8。同时应限制口令使用时间，保证定期更换口令。此外，在选择口令时尽量不要选择单词、用户名、生日等常用词，最好同时包含字母、数字、大小写、非字母数字符号，以防备攻击者通过猜测口令攻破系统。

4. 如何防止别人使用 ping 命令探测本机？

ping 命令通常用于测试主机和网络，黑客可以利用这一点进行网络扫描，伺机攻击网络主机。为了增强主机的安全性，建议禁止主机系统对该命令作出响应。可以编辑文件 /etc/rc.d/rc.local，增加下述命令：

```
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

其含义是不处理所有接收到的 ICMP ECHO 数据包。这样，每次系统启动后都执行该指令，可以阻止系统响应任何从外部或内部来的 ping 请求。

5. 怎样禁止远程 Telnet 登录服务？

关闭本机的 telnet 服务，逐一检查是否使用某种方法启动了 telnet 服务。

6. 怎样减小在网络中被攻击的风险？

系统管理：BIOS 安全设置、安全分区、系统文件实行最小权限原则、限制用户占用的资源、即使升级系统

用户管理方面：关闭不用帐号、注意 suid 程序的权限、对口令进行安全管理

网络服务管理方面：关闭不必要的服务、禁止响应 ping 命令、向其它用户屏蔽系统信息

7. 如何临时禁止一个用户的使用？

编辑/etc/passwd 文件，将该用户对应的行的第 2 个域更改为*，则该用户被停用，当要启用该用户时，可以将第 2 个域更改为 x 即可。

8. 如何设置用户自动注销？

Linux 可以控制系统在空闲时自动从 shell 中注销。编辑/etc/profile 文件，设置变量“tmout”为超时时间：

```
tmout=600
```

那么所有用户将在 600 秒无操作后自动注销。注意：修改了该参数后，必须重新登录，更改才能生效。

9. DAC 与 MAC 有哪些异同？

DAC 和 MAC 都是访问控制模型，按用户身份及其所归属的某预定义组来限制用户对某些信息项的访问，或限制对某些控制功能的使用。

DAC，由用户有权对自身所创建的访问对象(文件、数据表等)进行访问，并可将对这些对象的访问权授予其他用户和从授予权限的用户收回其访问权限。在 DAC 中，对某个

客体具有所有权（或控制权）的主体能够将该客体的一种访问权或多种访问权自主地授予其它主体，并可在随后的任何时刻将这些权限回收。DAC 是一种方便但不安全的文件系统访问控制方法，由于系统存在特权用户、对文件访问权限的划分不够细致以及程序对资源的完全控制等问题。

MAC 由系统对用户所创建的对象进行统一的强制性控制，按照规定的规则决定哪些用户可以对哪些对象进行什么样操作系统类型的访问，即使是创建者用户，在创建一个对象后，也可能无权访问该对象。强制访问控制模型 MAC 的最初目的是实现比 DAC 更为严格的访问控制策略。在 MAC 中，系统根据主体被信任的程度和客体所包含的信息的机密性或敏感程度来决定主体对客体的访问权。系统事先给访问主体和受控对象分配不同的安全级别，用户不能改变自身和客体的安全级别。在实施访问控制时，系统先对访问主体和受控对象的安全级别进行比较，只有在主体和客体的安全级别满足一定规则时，才允许访问。

10. 仅仅依靠 DAC 的 Linux 系统存在哪些不足？

- 1) 存在特权用户 root: 任何人只要得到了 root 权限，就可以为所欲为；
- 2) 对于文件的访问权的划分不细: 在 linux 系统中，对文件的操作只有所有者、所有组和其他这三类，没有办法进行再细的划分，所以不能更好的对访问进行控制；
- 3) suit 程序的权限升级: 如果设置了 suid 权限的程序有漏洞的话，很容易被攻击者利用；
- 4) DAC 问题: 文件目录的所有者可以对文件进行所有的操作，这给系统整体的管理带来了不便。

11. 什么是 SELinux？它的主要作用是什么？

SELinux 的全称是 Security-Enhanced Linux，即安全增强的 Linux。它在 Linux 内核中包含必要的访问控制，在传统的强制访问控制的基础上加入了灵活性支持，同时引入了基于角色的访问控制中的角色等概念，来克服传统强制访问控制的局限。受 SELinux 保护的程序只允许访问它们正确工作所需的文件系统部分，也就是说如果程序有意或无意地访问或修改它的功能所不需要的文件或者不在程序所控制的目录中的文件，则访问会被拒绝，动作会被记录到日志中。

12. SELinux 都有哪些特性？

没有 SELinux 保护的 Linux 的安全级别和 Windows 一样，是 C2 级，但经过 SELinux 保护的 Linux 安全级别可以达到 B1 级，下面是 SELinux 的一些特点：

- 1) 强制访问控制 MAC；2) 类型加强 TE (Type Enforcement)——对进程只赋予最小权限；3) 进行类型的迁移——防止权限升级；4) 基于角色的访问控制 RBAC——对用户只赋予最小权限；5) SELinux 策略——决定保护类别和方式。

13. 如何启动和禁止 SELinux，都有哪些办法可以知道 SELinux 当前的运行状态？

在 Fedora 系统中，我们可以使用 `sestatus` 命令来查看当前的 SELinux 模式，可以使用 `getenforce` 命令来显示当前的 SELinux 运行状态。可以使用 `setenforce` 命令来修改当前运行的 SELinux 模式，设置为 `enforcing`: 启用并强制执行系统上的 SELinux 安全机制，记录它拒绝的任何动作；设置为 `permissive`: 启用 SELinux，但是不强制执行安全策略。在该模式下，SELinux 不阻止任何操作，即使该操作违反了安全策略，但它会把违反的内容记录下来；设置为 `disabled`: 关闭 SELinux。

14. 什么是类型、什么是角色、什么是安全上下文？

类型 (type): 有时也称域 (domain)，类型把主体或者客体分成相关的组，以类型符号来标识，类型是基本的安全属性，SELinux 根据类型做授权决策，控制进程访问并阻止越权企图。所以，可以认为类型就是一组允许执行的操作的列表；角色 (role): SELinux

中每个用户被授予进入一种或者多种角色，在任何时刻一个用户只能处于一种角色。每个角色与不同的域关联，角色决定了用户可以访问哪些域，即通过角色管理用户的权限；安全上下文（context）：SELinux 的三个安全属性：用户身份、角色、类型，结合在一起构成了安全上下文。系统中的任何主体和对象都有自己的安全上下文，SELinux 根据安全上下文做安全决策。

15. 安全增强的内涵都有哪些？

第一、受保护的程序只允许访问它们正确地工作所需的文件系统能够部分，意味着程序的执行被严格监控，不能再访问或者修改它不需要的文件；第二、进行严格的访问控制审查，利用 SELinux 策略定义来实施具体的保护；第三、更为严格的权限升级核定，避免不恰当的类型迁移；第四、引入角色概念，赋予用户不同的角色，系统根据角色设置访问权限。

16. SELinux 与 Linux 系统原有的访问控制之间有什么关系？

安装并启用 SELinux 之后，Linux 系统的原有安全机制，即基于“所有者——组——其他”的访问控制机制仍然有效，二者可以共存。当在系统中执行某一操作时，原有的访问控制机制首先起作用。如果原有的访问控制规则允许了该操作，那么 SELinux 将会检查并且根据自己的安全策略允许或拒绝访问；但是如果原有的访问控制机制拒绝了该操作，那么 Linux 系统直接拒绝该操作，而与 SELinux 检查无关。

17. 如何查看安全上下文？

执行命令“ps -Z”则列出了当前运行进程的安全上下文，执行命令“ls -Z”可以列出文件对应的安全上下文。

18. 如何修改一个目录中默认文件的安全上下文？

chcon 命令设置或者更改一个或多个文件的安全上下文。若文件原来没有安全上下文则设置新的安全上下文，否则修改为指定值。

19. SELinux 的策略如何选择？用户如何根据自己的需要去定制 SELinux 策略？

SELinux 访问控制策略的制定、更改非常复杂。一般是下载安装缺省策略，然后在此基础上进行细微调整。制定或更改策略可以通过控制台命令实现。Fedora 系统默认安装了一个图形化的 SELinux 管理工具，可以非常方便地设置、修改 SELinux 的安全策略。用户可以使用此工具来依据自己的需要定制 SELinux 策略。

20. SELinux AVC 消息有什么作用？如何查看 AVC 消息？

AVC 消息是由 SELinux 因访问拒绝或者因制定的审计规则而产生的审核消息，Fedora10 默认安装并且打开了 SELinux 诊断工具 setroubleshoot，这个工具为桌面用户提供了 SELinux 访问受限的通知和详细信息，以及如何处理它们的建议。当系统中的某个操作违反了 SELinux 的安全策略时，该工具在桌面上弹出 AVC 警报。当看到 AVC 警报时，可以双击该通知图标，以打开 SeTroubleshoot 浏览器，也可以通过菜单项【应用程序/系统工具/SELinux 故障排除工具】来打开 SeTroubleshoot 浏览器，在 SeTroubleshoot 浏览器上面列出了所有检查到的 AVC 消息。

第 11 章 shell 程序设计

1. shell 程序就是把用户键入的 shell 命令按照控制结构组织到一个文本文件中，批量的交给 shell 去执行。

2. 安装有 bash、dash、sh、nologin 等多种 shell 解释器，可以通过命令：cat /etc/shells 命令查看系统中安装有哪些 shell，及这些 shell 的安装路径。

3. 与 C 语言等高级语言程序需要最终形成二进制可执行文件不同，shell 程序是通过 shell 命令解释器解释执行的，不生成二进制的可执行代码，这点和 Dos 下批处理程序（.BAT 文

件)的特性类似。

4. 包括: (1) 命令解释器路径; (2) 函数声明; (3) 程序主体;

5. Bash 脚本语言是一种弱类型的脚本语言。所谓弱类型脚本语言是指这种语言对类型的要求不严格, 同一个变量随着使用场合的不同, 可以存储不同类型的数据。

给一个变量赋值采用如下的格式:

变量=值 (注意: 等号两侧不能有空格)

6. 如果在函数内部需要使用传递给函数的参数, 一般用\$0、\$1、.....、\$n, 以及\$#、\$*、\$@这些特殊变量。\$0 为执行脚本的函数名, \$1 是传递给函数的第 1 个参数, 依次类推, \$n 为传递给函数的第 n 个参数; \$#为传递给函数的参数个数; \$*和\$@为传递给函数的所有参数, 两者的区别在于\$*把所有参数作为一个整体, 而\$@把所有参数看作拥有多个参数的集合, 可以单独访问每个参数。

7. 常见的运算符:

整数的算术运算符: +、-、*、/、%;

赋值运算符: +=、-=、*=、/=、%=;

位运算符: <<、>>、&、|、~、^;

位运算赋值运算符: <<=、>>=、&=、|=、~=、^=;

逻辑运算符: &&, ||, !, >, >=, <, <=, !=, ==

8. 有 if、case、for、while、until 等。

9. 通过\$n (n=0、1、2.....), \$#, \$*, \$@特殊变量和 shift 等命令实现。

10. /etc/bashrc 脚本负责进行一些全局变量的设置工作。

11. 代码清单及程序注释如下:

```
#!/bin/bash
#可执行文件的路径
COMMAND_PATH=/usr/local/bin/myserver
#得到文件的文件名, 本例中为: myserver
COMMAND_FILE=${COMMAND_PATH##*/}
#通过 ps 命令获取运行程序的经常标识符即 pid
function grep_proc_id()
{
    ps -C ${COMMAND_FILE} -o pid --no-headers
}
#启动可执行文件
function start_svc()
{
    echo "service starting"
    ${COMMAND_PATH}
}
#停止可执行文件
function stop_svc()
{
    pid=`grep_proc_id`
    if [ -n "${pid}" ] && [ 0 -ne ${pid} ];then
        kill -9 ${pid}
        echo "kill ${pid} "
```

```

        fi
        echo "servic stoped"
    }
#重新启动可执行文件
function restart_svc()
{
    stop_svc
    start_svc
    echo "service restart"
}
#输出程序使用方法及简单实例
function USAGE()
{
    echo "Usage: svc [start] [restart] [stop]"
    echo "to start,restart or stop service "
    echo "${COMMAND_PATH}"
    echo "Example: svc restart <ENTER>"
}
#程序主要路径
if [ $# -lt 1 ]; then
    USAGE
else
#根据传入参数完成启动、重启、终止程序的功能
    case $1 in
        start) start_svc ;;
        restart) restart_svc ;;
        stop) stop_svc ;;
        *) USAGE ;;
    esac
fi

```

第 12 章 gcc 的使用与开发

1. 一个 C/C++ 程序从开始编码到生成可执行的二进制文件至少要经过 4 个步骤：（1）预处理（Preprocessing）；（2）编译（Compilation）；（3）汇编（Assembly）；（4）链接（Linking）
2. 略。
3. 略，可参见正文表 12-1。
4. 默认路径放在/usr 目录下，其中库文件在/usr/lib 下，头文件在/usr/include 目录下。
5. 为了简化生成代码的步骤，GNU 提供了 make 工具。

make 工具读入一个文本文件，文本文件中主要记录了一些规则。规则说明了生成最终的二进制代码依赖哪些模块以及这些模块如何生成的信息。为了简化，这个文件通常命名为 makefile。

makefile 文件的一般格式是：

目标：依赖项列表

（Tab 缩进）命令...

其中，“依赖项”一般为生成目标所需的其他目标或者文件名。

6. 略，可参见正文表 12-3。

第 13 章 gtk+图形界面程序设计

1. 在 Xlib 之上出现了很多更高层的开发工具包。它们将底层的 Xlib 的 API 进行封装，提供更高级的接口，达到降低开发难度，提高开发效率的目标。此外，这些工具包大多并非完全专注于界面的开发，在常用数据结构（树、链表等）甚至数据库操作等方面都提供了接口。下面将介绍的 GTK+就是这样一个开发工具包。

2. GTK+由 3 个库组成：（1）GLib：是 GTK+和 Gnome 的底层核心库，提供了可以供 C 语言处理的数据结构、可移植的封装接口和用来处理事件循环、线程、动态加载的接口，构成了一个类似对象的体系结构。（2）Pango：是处理界面布局和国际化的库，构成了 GTK+2.0 处理字符和字体的核心。（3）Atk：提供了一些其他功能的接口。一个实现对 Atk 接口支持的应用软件，可以实现屏幕阅读、放大、使用其他输入设备等功能。

3. 除了 GIMP 外，大量使用 GTK+的著名软件还有图形桌面系统 GNOME、字处理软件 Abiword、图表软件 Dia 以及辅助开发 GTK+程序的 Glade 等。

4.

（1）头文件源代码：

```
#include <gtk/gtk.h>
//btnOne 的点击事件处理函数
void on_btn_One_clicked(GtkWidget *widget,gpointer data){
    g_print("Hello from btn_One!\n");
}
//btnTwo 的点击事件处理函数
void on_btn_Two_clicked(GtkWidget *widget,gpointer data){
    g_print("Hello from btn_Two!\n");
}
//btnThree 的点击事件处理函数
void on_btn_Three_clicked(GtkWidget *widget,gpointer data){
    g_print("Hello from btn_Three!\n");
}
//窗口关闭事件处理函数
void on_delete(GtkWidget *widget, GdkEvent *event,gpointer data){
    g_print("Window Closed.\n");
    gtk_main_quit();
}
```

（2）主程序源代码：

```
int main(int argc,char * argv[]){
    GtkWidget *window;//窗口
    GtkWidget *btnOne;//按钮 1
    GtkWidget *btnTwo;//按钮 2
    GtkWidget *btnThree;//按钮 3
    GtkWidget *table;//table 布局

    gtk_set_locale();
    gtk_init(&argc,&argv);
    //创建窗口
```

```

window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
gtk_window_set_title(GTK_WINDOW(window), "Hello Gtk!");
gtk_container_set_border_width (GTK_CONTAINER (window), 10);
//创建 3*3 的表格布局并和窗口关联
table = gtk_table_new(3,3,TRUE);
gtk_container_add(GTK_CONTAINER(window),table);
//关联窗口关闭事件
g_signal_connect (G_OBJECT (window), "delete_event",
                  G_CALLBACK (on_delete), NULL);
//创建按钮并关联相对应的事件处理程序
btnOne = gtk_button_new_with_label ("btn_One");
g_signal_connect (G_OBJECT (btnOne), "clicked",
                  G_CALLBACK (on_btn_One_clicked), NULL);
btnTwo = gtk_button_new_with_label ("btn_Two");
g_signal_connect (G_OBJECT (btnTwo), "clicked",
                  G_CALLBACK (on_btn_Two_clicked), NULL);
btnThree = gtk_button_new_with_label ("btn_Three");
g_signal_connect (G_OBJECT (btnThree), "clicked",
                  G_CALLBACK (on_btn_Three_clicked), NULL);
//在窗口布局上放置三个按钮
gtk_table_attach(GTK_TABLE(table),btnOne,0,2,0,1,GTK_FILL,GTK_FILL,0,0);
gtk_table_attach_defaults(GTK_TABLE(table),btnTwo,0,2,1,3);
gtk_table_attach_defaults(GTK_TABLE(table),btnThree,2,3,0,3);
//显示窗口
gtk_widget_show_all(window);
gtk_main();
return 0;
}

```

第 14 章 Qt 图形界面程序设计

1. Qt 是和 GTK+一样，是在 XLib 之上的开发包。
2. Qt 是采用信号/槽的方式实现消息管理，而 GTK+是采用类似回调函数的方式实现的。
3. 著名项目有：KDE、Nokia 相关软件、FEKO 等。
4. Qt 软件虽然由商业化公司维护，但是 Qt 也为软件的开发提供了多种选择方式。Qt 的版本总的可以分为自由软件和商业两类版本，无论那种版本都可以获得软件的源代码。自由版本可以免费使用，可以采用 GPL 或 LGPL 软件许可证的方式。商业版的版本是需要付费的，使用商业版的版本开发出的软件受到更少的限制，还可以提供及时的客户支持服务。在自由软件版本中如果想获得质量高的客户支持服务可以通过付费的方式按需要购买特定的支持服务项目。

5、

```

/*****
*
*myWidget.h
*
*****/

```

```

#include <qlayout.h>
#include <qpushbutton.h>
class MyWidget: public QWidget{
    Q_OBJECT
private:
//三个按钮一个布局
    QPushButton *btnOne;
    QPushButton *btnTwo;
    QPushButton *btnThree;
    QGridLayout *grid;
public:
    MyWidget(    QWidget* parent = 0, const char* name = 0);
    ~MyWidget();
protected:
//关闭事件处理程序
    void closeEvent(QCloseEvent *);
signals:
//关闭事件
    void closeSignal(QCloseEvent *);
public slots:
//按钮点击处理程序
    void onBtnOne();
    void onBtnTwo();
    void onBtnThree();
};
/*****
*
*myWidget.cpp
*
*****/
#include "myWidget.h"
#include <qmessagebox.h>
#include <qapplication.h>
//构造函数
MyWidget::MyWidget( QWidget* parent, const char* name): QWidget( parent, name){
    //创建按钮并关联事件处理函数
    btnOne=new QPushButton( "One", this, "one" );
    connect( btnOne, SIGNAL(clicked()), SLOT(onBtnOne()) );
    btnTwo=new QPushButton( "Two", this, "two" );
    connect( btnTwo, SIGNAL(clicked()), SLOT(onBtnTwo()) );
    btnThree=new QPushButton( "Three", this, "three" );
    connect( btnThree, SIGNAL(clicked()), SLOT(onBtnThree()) );
    //创建表格布局，并放置按钮
    grid = new QGridLayout( this, 3, 3, 10 );

```

```

        grid->addMultiCellWidget(btnOne,0,2,0,0);
        grid->addMultiCellWidget(btnTwo,0,0,1,2);
        grid->addMultiCellWidget(btnThree,1,2,1,2);
        //关联推出事件处理程序
        connect( this, SIGNAL(closeSignal(QCloseEvent *)), qApp,SLOT(quit()) );
    }
    //析构函数
    MyWidget::~MyWidget()
    {}
    //关闭窗口时发出关闭信号
    void MyWidget::closeEvent(QCloseEvent *ce){
        emit closeSignal(ce);
    }
    //按钮事件处理函数
    void MyWidget::onBtnOne(){
        QMessageBox::information(this,"BtnOne","Hello from btnOne!");
    }
    void MyWidget::onBtnTwo(){
        QMessageBox::information(this,"BtnTwo","Hello from btnTwo!");
    }
    void MyWidget::onBtnThree(){
        QMessageBox::information(this,"BtnThree","Hello from btnThree!");
    }
    /*****
    *
    *main.cpp
    *主程序
    *
    *****/
    #include "myWidget.h"
    #include <qapplication.h>
    int main( int argc, char **argv ){
        QApplication a( argc, argv );
        MyWidget w;
        a.setMainWidget( &w );
        w.show();
        return a.exec();
    }
    #####
    #qmake 生成的的工程文件
    #####
    TEMPLATE = app
    INCLUDEPATH += .

```

```
# Input
HEADERS += myWidget.h
SOURCES += main.cpp myWidget.cpp
```

第 15 章 集成开发环境 KDevelop 的使用

1. KDevelop 为用户在 Linux 下进行开发提供了统一的视图，集成了软件开发中用到的大多数工具。软件从设计、开发到最后的编译、发行所有的工作都可以在 KDevelop 下完成。另外，KDevelop 提供了动态加载插件的机制，可以通过编写插件来扩充 KDevelop 的功能，满足用户的特殊要求。

2. “类”、“文件树”、“文档”和“消息”视图是使用最频繁的视图。

3. “消息”视图显示系统的即时状态，主要是编译时产生的错误、警告等提示信息。

4. 开发新程序从创建新的工程开始。KDevelop 根据开发的软件类型，提供了多种工程模板，用户可以选用适合的工程模板，先建立一个框架，然后再对这个框架进行改进，完成自己特定的功能。要创建新工程，先选择菜单【工程/新建工程】，打开创建新工程向导。