

Linux 的基本配置与使用

学时计划：4 学时 理论，4 学时 实验

（实验二：Linux 下的驱动程序与应用软件管理）

教学大纲：

- 1、和 Windows 在一起的 Linux 无法启动了！
- 2、超级用户 root
- 3、桌面系统（X 服务器）
- 4、个性化配置 Linux 桌面系统
- 5、驱动程序与应用软件的安装
- 6、操作系统升级
- 7、讨论与思考

Linux 操作系统安装完成之后，就是如何使用 Linux 操作系统，本讲主要在 Ubuntu 桌面操作系统下讲解 Linux 操作系统的基本配置、个性化设置、驱动程序和应用软件安装，以及操作系统升级。

最后，重点介绍超级用户 root 和桌面系统的基本原理。

一、和 Windows 在一起的 Linux 无法启动了！

1. 1Linux 无法启动了！

场景描述：

张三的计算机安装了 Windows 7 和 Ubuntu11.04 两个操作系统，安装的方式是：

第一步：

安装 Windows 7 操作系统。将 500GB 的硬盘上划分了 3 个分区，C 盘 100GB、D 盘 120GB、E 盘 180GB，另有 100GB 的硬盘没有划分分区。Windows 7 操作系统安装到 C 盘，D 盘用来存放工作和学习资料，E 盘存放娱乐资料。

第二步：

安装 Ubuntu 11.04 操作系统。操作系统安装在尚未划分硬盘分

区的空间内，分区按照 Linux 推荐的分区方式。

第三步：

安装 GRUB，并设置为默认使用 Ubuntu 11.04。

在硬盘上安装 Windows 和 Linux 双操作系统后，计算机一直运行良好。直到一天，Windows 操作系统感染了病毒，张三计划重新安装 Windows 7 操作系统。

具体的安装方式是：

第一步：设置计算机从光盘启动。

第二步：将计算机 C 盘全新格式化，全新安装 Windows 7 到 C 盘。Windows 7 安装顺利完成。

故障发生了！

张三的计算机在 Windows 7 安装完成后，启动计算机就直接进入 Windows 7。GRUB 的启动界面不出现，无法使用还正常的 Ubuntu 11.04 操作系统。

1.1.1 故障发生的原因

故障发生的原因很简单，Windows 7 操作系统重新安装时，自动用 Windows 7 的启动程序覆盖了 Grub 多重引导程序。但是 Windows 7 的引导程序却不具备发现 Linux 操作系统和引导 Linux 操作系统的功能。

1.1.2 解决故障的办法

解决故障的办法很简单：在 Windows 7 安装成功后，重新安装 Grub 即可。

具体的操作方法是：

第一步：使用 Linux 光盘启动计算机，如果是 OpenSUSE 操作系统，进入“Rescue System”模式，或者在 Ubuntu 下使用“Try Ubuntu without change to your computer”。进入 Linux 操作系统。

第二步：从光盘运行 Linux 操作系统后，通过下属命令重新安装 Grub 并完成设置。

```
Grub
Find /boot/grub/stage1
Root (hdx,y)
Setup (hd0)
Quit
```

注意，命令中的 X, Y 要根据具体情况进行调整。

1.2 Linux 和 Grub 对分区的表示方法

1.2.1 Linux 下的分区设备的命名方式是：

sda(/dev/sda)就是指第一块硬盘本身(SATA、SCSI 硬盘是用 sda 表示的,而 IDE 硬盘则是用 hda 表示)。

sda1 表示第一块硬盘第 1 个主分区 (扩展分区在这里当作主分区看)；

sda2 表示第一块硬盘第 2 个主分区；

sda3 表示第一块硬盘第 3 个主分区；

sda4 表示第一块硬盘第 4 个主分区；

sda5 表示第一块硬盘第 1 个逻辑分区；

sda6 表示第一块硬盘第 2 个逻辑分区；依次类推...

第二块硬盘则表示为 sdb 或者 hdb。

由于 sda、hda 等名称是 Linux 下表示法，所以想要查看本机的硬盘信息的话，Linux 下可用命令查看。

```
sudo fdisk -l
```

1.2.2 Grub 对硬盘分区设备的命名方式是：

系统的第一个硬盘驱动器表示成(hd0)，其上的第一个分区表示为(hd0,0)，也就是说对于硬盘，采用(hdx,y)的形式来表示，x、y 都是从 0 开始计数的，x 表示硬盘号，y 表示分区号。

由于主分区只能有四个，所以第一硬盘的四个主分区分别用(hd0,0)、(hd0,1)、(hd0,2)、(hd0,3)来表示；逻辑分区则从(hd0,4)开始算，即第一逻辑分区用(hd0,4)，第二逻辑分区用(hd0,5)来表示，依次类推。

一般机子的硬盘都是一个主分区，其余是逻辑分区，因此 C 盘用(hd0,0)，D 盘用(hd0,4)来表示。

光盘用(cd)表示，第一软驱用(fd0)表示。

Grub 下用 ls 命令查看分区下的文件，从而确定硬盘的分区信息。

```
ls 列出当前的所有设备
```

```
ls -l 详细列出当前的所有设备，对于分区，会显示其 label 及 uuid。
```

ls (hd0,1)/ 列出(hd0,1)分区根目录下的所有文件。

1.3 linux 的启动流程

Linux 操作系统的启动流程是了解 Linux 的基础，也是对于操作系统无法启动时解决问题的基本理论依据。如果不了解如何启动 Linux，那么就无法判断系统无法启动的原因。也就是大部分的计算机管理员或者用户在操作系统无法启动时，选择重新安装操作系统而不是进行维护的原因。

1.3.1 Linux 的启动流程

Linux 的启动可以基本划分为 10 个基本步骤。

Step1:加载 BIOS 的硬件信息，并获取第一个启动设备的代号。

Step2:读取第一个启动设备的 MBR 的引导加载程序（即 lilo、grub、spfdisk 等）的启动信息。

Step3:加载操作系统的核心信息，核心开始解压，并尝试驱动所有的硬件设备。

Step4:核心执行 init 程序并获得运行信息。

Step5:init 执行/etc/rc.d/rc.sysinit 文件。

Step6:启动核心的外挂模块（/etc/modprobe.conf）。

Step7:init 执行运行的各个批处理文件（Script）。

Step8:init 执行/etc/rc.d/rc.local 文件。

Step9:执行/bin/login 程序，等待用户登录。

Step10:登录之后开始以 shell 控制主机。

1.3.2 Linux 启动过程的关键问题

(1) Linux 核心的引导

Linux 操作系统的核心是放在文件系统里的，要想正确加载核心就必须提前识别文件系统。系统刚启动的时候，就需要通过引导加载程序（即 LIL0、Grub、spfdisk 等）来识别文件系统，引导核心。要想加载 Linux 的核心就必须能识别 Linux 的文件系统，要加载 Windows 核心就必须能识别 Windows 文件系统。

Grub 是同时支持 Linux 和 Windows 的。Windows 的加载程序并不支持 Linux 文件系统，所以在多重启动设置的时候，总是要先装 Windows 而后装 Linux 的原因也就是如此。

Linux 操作系统核心文件在哪里呢？一般来说，它会放在/boot 里，并且取名为/boot/vmlinuz。例如，Ubuntu 11.04 Desktop 的核心文件存放在/vmlinuz。

在加载核心的过程中，系统只会“载入根目录”，并且是以只读方式载入的。有时为了让某些功能可以用文件的方法来读取，有的系统在启动时，会建立虚拟盘（ramdisk），这就需要使用 initrd 以及 linuxrc 了。在这种情况下，加载核心的时会一并加载 initrd 的映像文件（boot/initrd-xxxx.img），并利用 linuxrc（在 initrd 的映像文件内）程序来加载模块。在核心驱动外部硬件的工作完成之后，initrd 所建立的虚拟盘就会被删除。

(2) 第一个程序 init 的加载和 runlevel 设置

在核心加载完成之后，系统就准备好了，等待程序的执行。整个 Linux 系统中，第一个执行的程序就是“/sbin/init”。init 程序利用设置文件“/etc/inittab”来获取运行等级，通过运行等级的设置值启动不同的服务项目。运行等级是指 Linux 通过设置不同等级来规定系统用不同的服务来启动，让 linux 的使用环境不同。

“/etc/inittab”中有这么一句

“si::sysinit:/etc/rc.d/rc.sysinit”，表明系统需要主动使用“rc.sysinit”这个 shell 脚本来设置系统环境。但这个文件的文件名在不同的 Linux 发行版中是不一样的，需要自行查看确认。

说明：了解到 ubuntu 在 6.10 开始用 upstart 替代 init，主要脚本都在/etc/event.d 下面，默认情况下/etc 下没有 inittab 文件。

举例：inittab 文件的内容：

```
[root@linux ~]#vi /etc/inittab
# 设置系统启动默认的运行等级设置项目
id:3:initdefault:

# 开始启动运行等级的服务前，使用检测与初始化系统环境的设置文件：
si::sysinit:/etc/rc.d/rc.sysinit

# 7 个不同运行等级需要启动的服务的脚本放置位置路径：
10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3
14:4:wait:/etc/rc.d/rc 4
15:5:wait:/etc/rc.d/rc 5
16:6:wait:/etc/rc.d/rc 6

# 是否运行按下[ctrl]+[alt]+[del]就重新启动的设置项目：
```

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# 本机终端启动的个数：
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# 在 X Window（运行级别 5）环境下启动脚本设置项目
x:5:once:/etc/X11/prefdm -nodaemon
```

(3) init 处理系统的初始化流程 (/etc/rc.d/rc.sysinit)

获取网络环境与主机类型。读取网络环境设置文件 "/etc/sysconfig/network"，获取主机名称与默认网关等网络环境。

测试与载入内存设备/proc 及 usb 设备/sys。除了/proc 外，系统会主动检测是否有 usb 设备，并主动加载 usb 驱动，尝试载入 usb 文件系统。

决定是否启动 SELinux。SELinux (Security-Enhanced Linux) 是美国国家安全局 (NSA) 对于强制访问控制的实现，是 Linux 系统上最杰出的新安全子系统。

接口设备的检测与即插即用 (pnp) 参数的测试。

用户自定义模块的加载。用户可以在

"/etc/sysconfig/modules/*.modules" 加入自定义的模块，此时会加载到系统中。

加载核心的相关设置。

设置系统时间 (clock)。

设置终端的控制台的字形。

设置 raid 及 LVM 等硬盘功能。

以只读方式查看检验磁盘文件系统。

进行磁盘配额 quota 的转换。

重新以读取模式载入系统磁盘。

启动 quota 功能。

启动系统随机数设备 (产生随机数功能)。

清楚启动过程中的临时文件。

将启动信息加载到"/var/log/dmesg"文件中。

如果想知道启动过程中发生了什么事可以查看 dmesg 文件。

(4) 启动系统服务 "/etc/rc.d/rc*.d" 与设置文件 "/etc/sysconfig"。

在结束了 inittab 中的 rc.sysinit 之后,系统可以顺利工作了,只是还需要启动系统所需要的各种服务,这样主机才可以提供相关的网络和主机功能。因此根据之前设置的运行等级,会启动不同的服务项目。如果当时在 inittab 中选择了等级 3,系统则会在 "/etc/rc.d/rc3.d" 目录中运行相应的服务内容,选择等级 5,就在 "/etc/rc.d/rc5.d" 目录内。

该目录下的内容全部都是以 S 或 K 开头的链接文件,都链接到 "/etc/rc.d/init.d" 目录下的各种 shell 脚本。S 表示的是启动时需要 start 的服务内容,K 表示关机时需要关闭的服务内容。如果需要自己增加启动的内容,可以再 init.d 目录中增加相关的 shell 脚本,然后在 rc*.d 目录中建立链接文件指向该 shell 脚本。这些 shell 脚本的启动或结束顺序是 S 或 K 字母后面的数字决定,例如 S01sysstat 表示第一个执行 sysstat 脚本,S99local 表示排在第 99 位执行 rc.local 脚本。

(5) 用户自定义引导程序 (/etc/rc.d/rc.local)

一般来说,自定义的程序不需要执行上面所说的繁琐的建立 shell 增加链接文件的步骤,只需要将命令放在 rc.local 里面就可以了,这个 shell 脚本就是保留给用户自定义启动内容的。

(6) 启动终端和 X-Window 界面

完成了系统所有的启动任务后,Linux 会启动终端或 X-Window 来等待用户登录。

tty1,tty2,tty3... 这表示在运行等级 1,2,3,4 的时候,都会执行"/sbin/mingetty",而且执行了 6 个,所以 Linux 会有 6 个纯文本终端,mingetty 就是启动终端的命令。

除了这 6 个之外还会执行"/etc/X11/prefdm -nodaemon"。这个主要启动 X-Window。

(7) 有关变换运行等级

当 Linux 登录之后, 有时候需要更换运行等级, 一种方法是改变 "/etc/inittab" 内的设置内容, 将 "id:3:initdefault:" 中的数字改成相应等级, 然后重启即可。

如果只是暂时地改变运行等级, 下次启动还是按原等级登录, 可以直接使用 `init [0-6]` 命令来改变运行等级。一般来说, 运行等级的不同只是相关的启动服务内容的不同而已, 因此使用命令改变等级会比较两个改变等级之间的服务内容, 关闭一些新等级中不需要的服务项目, 启动新等级需要的服务, 而保留新等级和原等级中共有的服务内容。查询目前等级的命令很简单, 只需要输入 `runlevel` 即可。如: `[root@linux ~]# runlevel` 结果显示: `N 3`

二、超级用户 root

2.1 什么是 root

root 权限跟 SYSTEM 权限可以理解成一个概念, 高于 administration 权限。

root 是 Linux 和 unix 系统中的超级管理员用户帐户, 该帐户拥有整个系统至高无上的权利, 所有对象他都可以操作, 所以很多黑客在入侵系统的时候, 都要把权限提升到 root 权限, 用 windows 的方法理解也就是将自己的非法帐户添加到 root 用户组。

administrator 是 windows nt 内核系统中的超级管理员用户帐户, 也拥有极高的权限, 而 SYSTEM 用户组可以理解为计算机系统自身的权限, 可以操作一切对象, 对一切进程进行操作。

2.2 Debian 和 Ubuntu 的 root 用户

对于绝大多数的 Linux 发行版而言, 安装中都需要设置两个权限: 一个是 root 用户, 另一个是用于登陆系统的普通用户。但是在 Debian 和 Ubuntu 系统中, 系统却没有 root 用户, 仅有一个普通用户。

在安装中设置的用户在某种情况下充当了 root。日常使用中, 该账号按照自己的权限完成操作, 没有任何特殊权限。在需要使用 root 权限进行操作时, 使用 `sudo` 命令来运行相关程序。Sudo 命令运

行时会要求输入口令，这个口令就是该普通账号的口令。

在 Ubuntu 中，可以通过特殊的设置，启用 root 口令。但是这样的操作是不安全的。

2.3sudo

sudo 以系统管理者的身份执行指令，也就是说，经由 sudo 所执行的指令就好像是 root 亲自执行一样。

命令格式：

```
sudo [-h|V|ksh]
```

```
sudo [ -b ] [ -p prompt ] [ -u username/#uid ] -s
```

```
sudo command
```

参数说明：

-V 显示版本编号

-h 会显示版本编号及指令的使用方式说明

-l 显示出自己(执行 sudo 的使用者)的权限

-v 因为 sudo 在第一次执行时或是在 N 分钟内没有执行 (N 预设为 5) 会问密码，这个参数是重新做一次确认，如果超过 N 分钟，也会问密码。

-k 将会强迫使用者在下一次执行 sudo 时问密码 (不论有没有超过 N 分钟)

-b 将要执行的指令放在背景执行

-p prompt 可以更改问密码的提示语，其中 %u 会代换为使用者的帐号名称，%h 会显示主机名称

-u username/#uid 不加此参数，代表要以 root 的身份执行指令，而加了此参数，可以以 username 的身份执行指令 (#uid 为该 username 的使用者号码)

-s 执行环境变量中的 SHELL 所指定的 shell，或是 /etc/passwd 里所指定的 shell

-H 将环境变量中的 HOME (家目录) 指定为要变更身份的使用者家目录 (如不加 -u 参数就是系统管理者 root)

三、桌面系统（X 服务器）

Linux/UNIX 操作系统的图形桌面环境经历了由无到有，由 X-Windows 到 GNOME、KDE 的发展历程，本节介绍 Linux/UNIX 操作系统的图形桌面环的发展历程，并解释 X-Windows 的工作原理，分析 GNOME 和 KDE 各自的技术特点。

3.1 什么是桌面系统

传统的 UNIX 操作系统都是只有命令行终端的用户界面，用户要完成某项操作，就必须在命令行中输入各种命令。这种操作方式对用户要求比较高，用户必须要牢记并熟练使用操作系统的各种命令。直到 20 世纪 80 年代中期，UNIX 业界出现了第一个图形化用户界面标准——X-Windows。

X-Windows 又简称为 X，是在 1984 年由麻省理工学院（MIT）和当时的 DEC 公司合作开发的一个图形视窗环境。准确地说，X-windows 并不是一个像微软 Windows 操作系统一样的完整的图形环境，而是图形环境与 UNIX 操作系统内核间的中间层。其并不负责控制视窗界面的控制，而是把它交给了第三方的图形环境程序进行处理。由于 X-Windows 为开发人员提供了开发的应用程序接口（Application Programmers Interface, API），所以任何厂家都可以在它的基础上开发出自己的 GUI 图形环境。

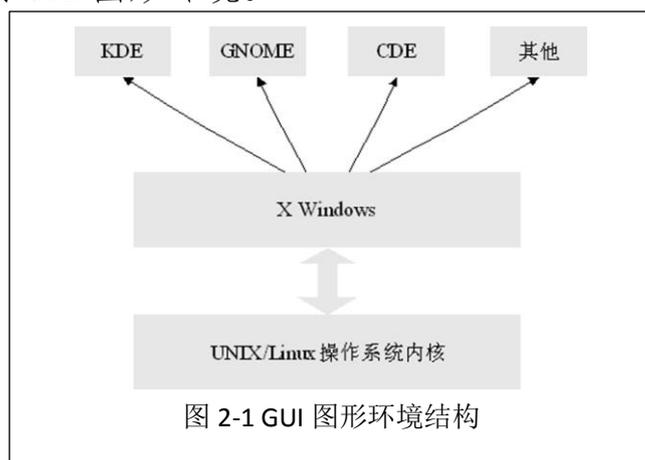


图 2-1 GUI 图形环境结构

在 Linux 上，桌面系统和 Windows 的桌面系统是不一样的，Linux 的桌面系统是一个应用软件的概念。用户如果选择安装桌面系统这个应用软件，那么 Linux 操作系统就可以使用图形话管理，例如 Ubuntu 11.04 Desktop 版本。如果用户没有选择安装图形界面，那么 Linux

操作系统就没有图形化管理，例如 Ubuntu 11.04 Server 版本。

在桌面系统的基础上，还有另一个概念，那就是主题/风格。例如在 Linux 桌面系统上可以进行桌面窗体的风格化变换，可以调整色彩、桌面墙纸、字体大小、甚至安装桌面皮肤等。这个概念和 Windows 的桌面系统是一致的。

演示：

在 Windows 操作系统中更换桌面主题。

在 Linux 操作系统中更换桌面主题。

在 Fedora 15、Ubuntu 11.04 Desktop 中查看两种 Linux 桌面系统。

3.2 Linux 中常见的桌面系统有哪些？

GNOME 和 KDE 是目前 Linux/UNIX 系统下最流行的两大图形桌面环境，经过多年的发展，无论是 GNOME 还是 KDE，在稳定性以及易用性方面都已相当优秀。但与微软的 Windows 相比，GNOME 和 KDE 在整体设计和操作习惯方面都有很大的不同。

KDE 自推出后吸引了众多的自由软件开发者参与，也得到了像 SuSE、Caldera、IBM、RedHat、西门子等公司的资金和技术支持，因而迅速发展。界面华丽、应用程序丰富是 KDE 的最大特点，其应用程序从浏览器、邮件收发工具、办公软件、下载软件、视频/音频播放器、即时通信工具到刻录工具等都一应俱全。

GNOME 项目的规模不及 KDE，其项目最初的时候只有图形环境，几乎所有的应用软件都是由其他的开源项目提供，包括著名的 FireFox 浏览器、OpenOffice.org 办公套件以及图形图像处理软件 Gimp 等。后来由于 KDE 的 Qt 版权问题，众多厂家把注意力转向了 GNOME，GNOME 也因此得以迅速发展，并最终成为了当今 Linux 两大 GUI 图形环境之一。

3.2.1 GNOME

GNOME 是于 1997 年 8 月由墨西哥的 Miguel De Icaza 为首的 200 多名程序员所开发的一个完全开源免费的图形用户环境。在发展过程中得到了占 Linux 市场份额最大的发行商 Red Hat 公司的支持，拥有了大量的应用软件。同时它也是 Red Hat Linux 发行版本所使用的默

认图形用户环境，Red Hat Enterprise Linux 5.2 中使用的 GNOME 版本是 2.16.0。

3.2.2 KDE

KDE, K 桌面环境 (Kool Desktop Environment) 的缩写。一种著名的运行于 Linux、Unix 以及 FreeBSD 等操作系统上面自由图形工作环境，整个系统采用的都是 TrollTech 公司所开发的 Qt 程序库 (现在属于诺基亚公司)。

3.3 切换使用不同的桌面系统

目前 Ubuntu、Fedora 使用的桌面系统是 GNOME，Kubuntu、openSuSE 使用的桌面系统是 KDE。

3.3.1 在 Ubuntu 11.04 Desktop 上安装 KDE

通过网络直接安装的方法是：

```
sudo add-apt-repository ppa:kubuntu-ppa/backports
sudo apt-get updatesudo
apt-get install kubuntu-desktop

sudo startkde
```

3.3.2 选择不同桌面系统的发行版

现在大部分的 Linux 发行版，都会结合不同的桌面系统制作对应的发行版。用户可以根据自己的喜欢，选择不同的桌面系统和对应的发行版本。

Fedora 15 集合四个不同的桌面系统，制作了 4 个不同的 Linux 发行版，也就是说 Fedora 15 有 4 个不同的版本。

Ubuntu 11.04 Desktop 结合六个不同的桌面系统，发行了 1 个 Linux 发行版，同时发行了 5 个 Ubuntu 派生版本。

四、个性化配置 Linux 桌面系统

本部分的介绍依据 Ubuntu 11.04 Desktop 32bit 进行。

4.1 个性化配置

在本部分中，主要介绍四个方面的操作：

配置显示模式和桌面墙纸

使用控制台模式

查看硬盘的数据

关机与注销

演示：

在 Ubuntu 11.04 Desktop 32bit (GNOME) 下进行演示：

配置显示模式和桌面墙纸

使用控制台模式

查看硬盘的数据

关机与注销

4.2 网络配置

配置 Ubuntu 的网络

创建 VPN 与使用

使用控制台配置网络

使用控制台配置网络的命令：

查看网络配置参数：

```
Sudo ifconfig
```

配置网络参数：

```
Sudo vi /etc/network/interfaces
```

演示：

在 Ubuntu 11.04 Desktop 32bit (GNOME) 下进行演示：

配置 Ubuntu 的网络

创建 VPN 与使用

使用控制台配置网络

4.3 系统运行性能

查看系统正在运行的进程

查看系统的 CPU、Memory and Swap、Network 的运行历史

查看系统的硬盘情况

演示：

在 Ubuntu 11.04 Desktop 32bit (GNOME) 下进行演示：

查看系统正在运行的进程

查看系统的 CPU、Memory and Swap、Network 的运行历史
查看系统的硬盘情况

五、驱动程序与应用软件的安装

5.1 安装驱动程序

在 Linux 中，驱动程序是针对 Linux 内核的，因此选择相应的驱动程序是，只需要针对 Linux 内核就可以了。

具体的安装方法，不同厂商的驱动程序各不相同，只需按照相应的用户手册安装即可。

5.2 安装与卸载应用软件

在 Linux 上安装和卸载应用软件越来越成为一件简单的事情。

Ubuntu 11.04 Desktop 上集成了一个软件 Ubuntu Software Center，通过此软件可以方便的进行软件的安装、卸载和管理。

Fedora 15 上集成了一个软件 Add/Remove Software，通过此软件可以方便的管理软件，并在线安装。

演示：

演示 Ubuntu 11.04 Desktop 的 Ubuntu Software Center。
演示 Fedora 15 的 Add/Remove Software。

六、操作系统升级

6.1 Linux 有没有漏洞？

毋庸置疑，Linux 是肯定有漏洞的，并且肯定也不会少。

<http://sebug.net/>记录了大部分有影响力的软件的漏洞情况。通过此网站可以查看到 Linux 的漏洞发生情况。

Solaris 的漏洞情况：<http://sebug.net/appdir/Solaris/>

Linux Kernel 的漏洞情况：<http://sebug.net/appdir/Linux/>

Apeche 的漏洞情况：<http://sebug.net/appdir/Apache/>

6.2 系统升级

不同的 Linux 发行版都提供了系统升级的方式，其系统升级的方

式和 Windows Update 的工作方式是相同的。

Ubuntu 11.04 Desktop 可以通过 Update Manager 进行系统升级。

Ubuntu 11.04 Desktop 和 Server 都可以使用命令控制台进行系统升级。

具体的升级命令：

更新系统的更新列表：

```
Sudo apt-get update
```

开始进行软件升级：

```
Sudo apt-get upgrade
```

Fedora 15 可以通过 Software Update 进行系统升级。

演示：

演示 Ubuntu 11.04 Desktop 的 Update Manager。

演示 Fedora 15 的 Software Update。

演示 Ubuntu 下使用 apt-get 命令进行系统升级。

七、讨论与思考

7.1 桌面系统

为什么 Windows 的桌面系统不可更改？

Windows 的桌面系统也是一个应用程序么？

7.2 Linux 启动

安装 Windows 和 Linux 操作系统时，是否可以不安装 Grub？

安装 Ubuntu 和 Fedora 双操作系统时，应该遵循什么顺序？

安装 Windows 7、Ubuntu 11.04 Desktop、Fedora 15 三个操作系统时，应该遵循什么顺序？