

第2章 linux基础





本章内容

- 了解**LINUX**操作系统的结构
- 掌握登录和退出过程
- 简要介绍一些常用的**shell**
- 介绍一些初学者会用到的命令
- 简要介绍**shell**中的一些元字符
- 掌握几个常用文本编辑器
- 命令**echo, exit, hostname, ls, man, passwd, set, setenv, uname, whatis, whereis, who, whoami, alias, cal, cat, cd, mkdir, pwd, rmdir, uptime**





Linux软件体系结构

应用程序
用户接口
(AUI)

应用程序：编译器、字处理器、电子制表软件、ftp、telnet、web浏览器、基于X的GUI

Linux shell: Bash、Tcsh、Zsh等

应用程序
界面
(API)

语言函数库：C、C++、Java等

系统调用接口（内核的入口）

操作系统

Linux 内核：

文件管理

进程间通信IPC

主存及二级存储管理

进程管理

CPU调度

设备驱动程序：鼠标驱动、打印机驱动、CD-ROM驱动、DVD驱动、硬盘驱动等

硬件





登录和退出

- **LINUX**系统是**多进程**、**多用户**和**交互式**的计算环境。
- 两种登录和退出**Linux**系统的方式
 - 基于文本的界面
 - ▶ 局域网连接
 - ▶ 因特网连接
 - ▶ 独立连接
 - telnet软件（命令）用于远程登录
 - 基于图形用户界面



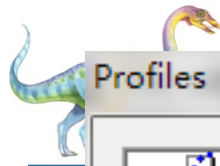


基于文本的界面的连接到Linux系统的方式

- 局域网连接：一般使用远程登录软件通过连接到局域网的一台Linux主机或服务器上。在Windows下使用的远程登录软件有SSH、telnet等，在<http://www.ssh.com/>下载SSH软件。
- 因特网连接：因特网连接与局域网连接相似。因特网中的Linux服务器很多。ssh的设置，如：
 - host: 192.168.227.128 , user:user1, port:22, password:
- 独立连接：用户使用一台装有Linux系统的计算机，使用文本的界面。
- 当使用以上三种方式连接Linux系统时，需要向系统输入正确的用户名和口令。在成功登录到Linux计算机后，屏幕会出现一个诸如“\$” (有可能是其他的符号)的shell提示符。接着用户可以使用各种各样的linux命令了。

Login : **your username**
Password: **your password**





Profiles



- Quick Connect
- Profiles
 - Profile Name
 - defaultsftp

- Colors
- Tunneling
- File Transfer
- Favorite Folders
- Connection
- Cipher List
- Authentication
- Keyboard

Configure protocol settings for the connection. New settings will take effect upon next login.

Specify * as the host name or the user name to be prompted for the information when the profile is chosen for

Host
User
Port

Encryption

MAC

Compression

Terminal

☐ Connect through fire

☐ Request tunnels only (disable ter

OK

Cancel





```
192.168.227.128 - Profile Name - SSH Secure Shell

File Edit View Window Help

[Icons]

Quick Connect Profiles

Last login: Tue Mar 19 12:34:58 2013 from 192.168.227.1
[user1@localhost ~]$ ls
Desktop dir1 filebat
[user1@localhost ~]$

Connected to 192.168.227.128  SSH2 - aes128-cbc - hmac-md5 86x24
```





文本界面登录

- 如果安装过程没有选择图形桌面环境，或在启动过程中没有启动图形桌面环境，则系统引导后进入文本界面：

login : **your username**

password: **your password**

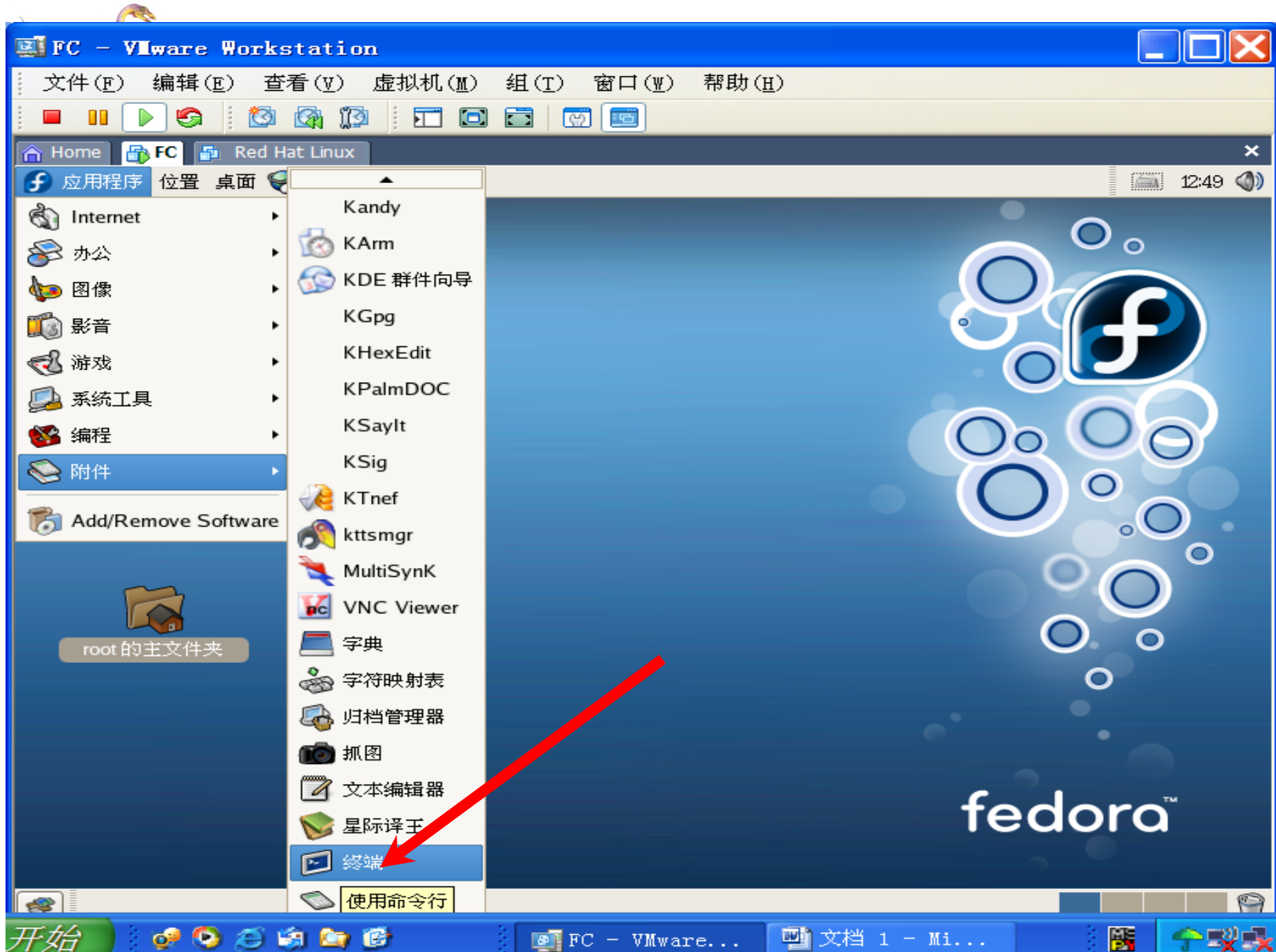


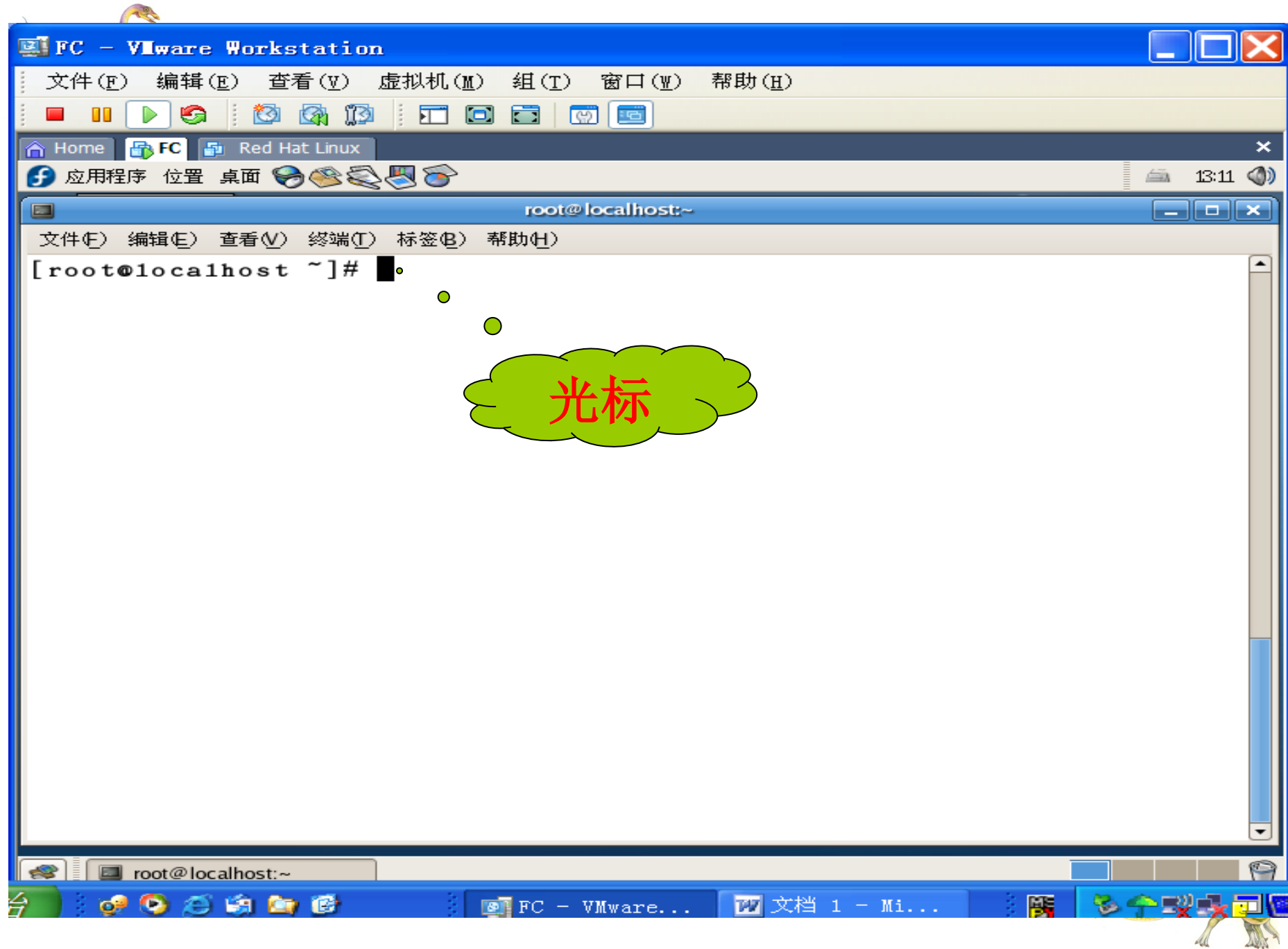


基于图形用户界面

- 用户使用一台装了**GNOME**或**KDE**图形桌面的**Linux**系统，可以使用图形界面的登录窗口登录到系统中。
- 使用**Xmanager**或**X-Win32**等客户端软件，在**windows**下登录到**Linux**，可以使用**linux**的图形界面。
- 在图形界面中通过鼠标点击“**应用程序→附件→终端**”操作进入文本界面。**如下图**
- **Linux**的很多命令需要在文本界面下完成的，**Linux**用户使用的命令要比**Windows**操作系统多，且功能强大。









启动方式

- **Linux**系统的运行级别（启动级）从 **0-6** 共7个。
 - **0** 为**停机**，关闭系统。
 - **1** 为**单用户模式**，就像**Windows**下的安全模式类似。
 - **2** 为**多用户模式**，但是没有**NFS** 支持。
 - **3** 为**完整的多用户模式，是标准的运行级**。
 - **4** **保留**，在一些特殊情况下可以用它来做一些事情。例如在笔记本电脑的电池用尽时，可以切换到这个模式来做一些设置。
 - **5** **X Window** 系统了。
 - **6** 为**重新启动**，运行 **init 6** 机器就会重启。 **startx**命令
- 运行级配置文件放在**/etc/inittab**中（**ubuntu**没有改文件），有一行“**id:5:initdefault**”
- **root**身份在终端上执行**telinit n**，进入运行级**n**。





登录和退出（续）

- 文本界面启动，用户退出系统
按<Ctrl-D>键或logout命令
- 图形界面，用户退出系统，Fedora Core:
 - 鼠标点击“桌面→注销”
- 关机命令：
 - 命令行方式：shutdown, halt, init 0, poweroff等，需要root权限
 - 图形桌面：鼠标点击“桌面→关机”





关机命令

- 使用 **shutdown** 或 **halt** 命令关闭Linux系统。当然使用这些命令需要有管理员用户权限。
- 例：下面是指定在早上 **8:00** 关机。
\$ shutdown -h 8:00
- 例：下面的命令是指定计算机在三分钟后关机。
\$ shutdown -h +3
- 例：下面的命令是指定计算机立刻关机。
\$ halt
- 例：下面的命令使计算机重新开机。
\$ reboot
\$ init 6

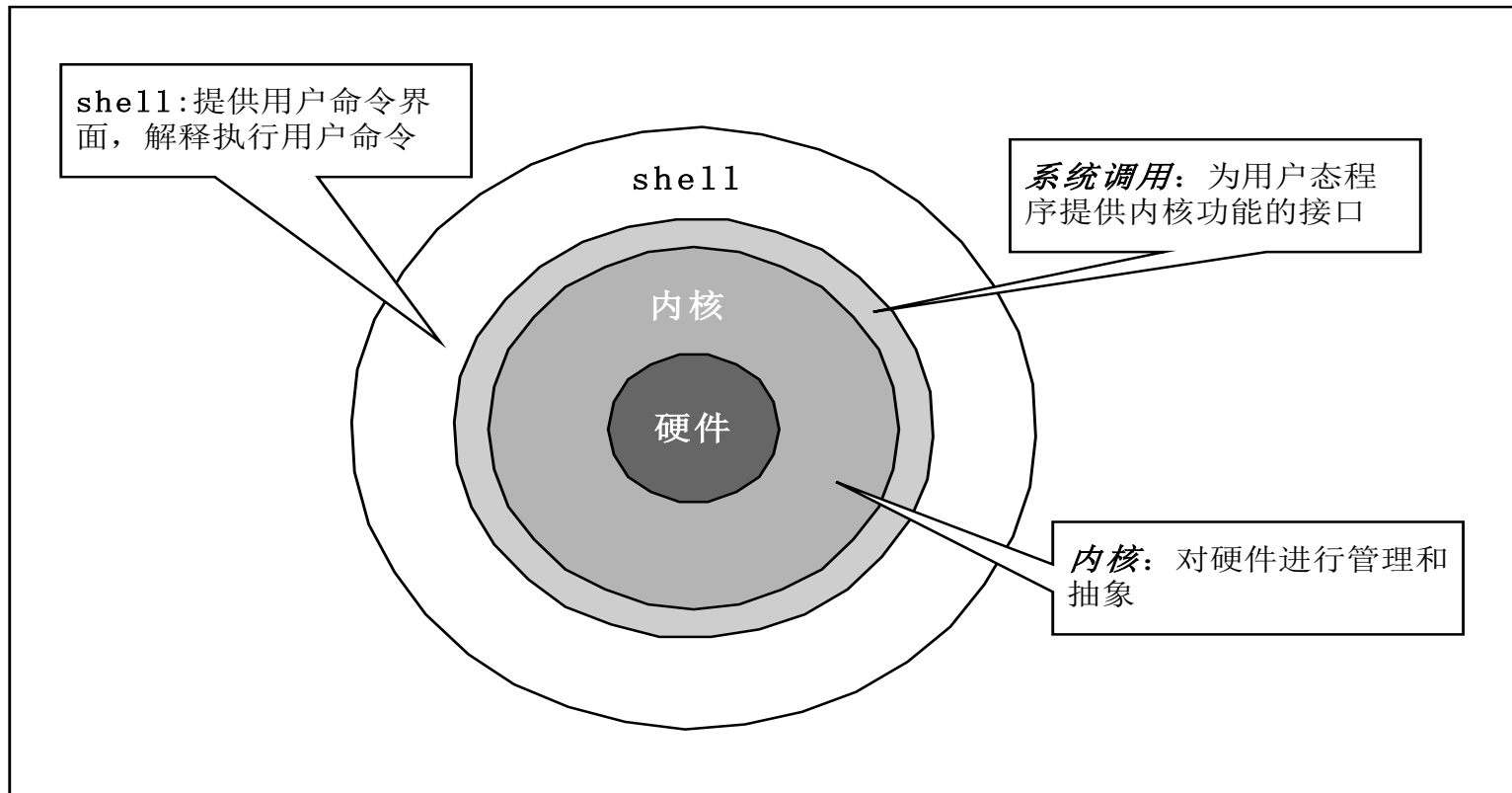




shell简介

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

■ **shell**也被称为Linux的命令解释器（**command interpreter**）



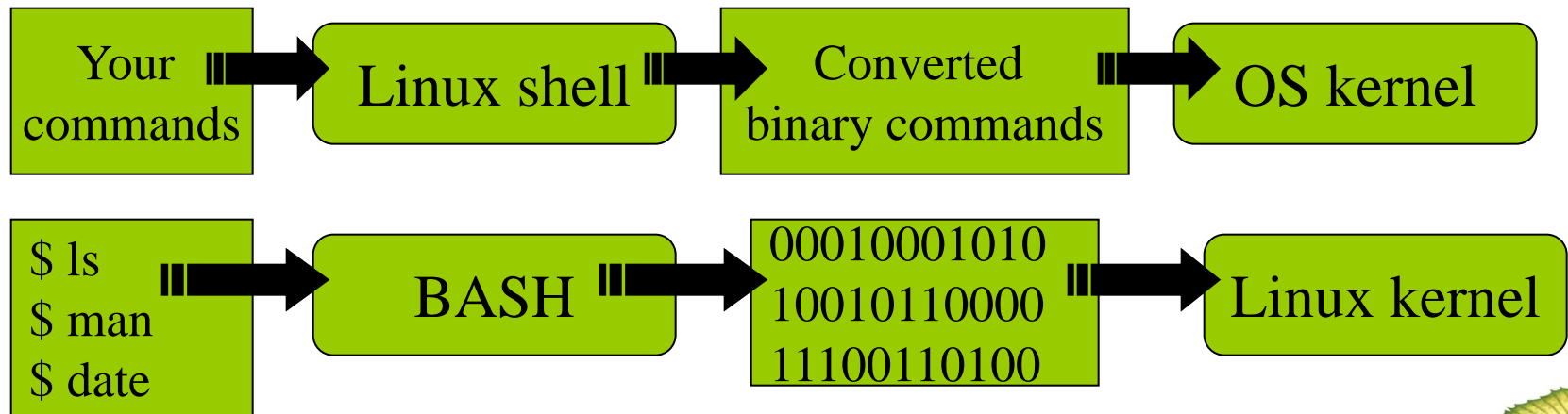


Linux Shell



■ Why Shell?

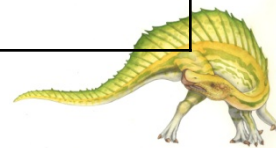
- Computer only realize the command in binary form which is difficult for most of human
- So OS provides a special program call 'shell' accepts human's command in 'readable' form and translates them into 1 and 0 stream





常用的shell

shell名称	相关历史
sh (Bourne)	源于UNIX早期版本的最初的shell
csch,tcsh, zsh	C shell及其变体，最初是由Bill Joy在Berkeley UNIX上编写的。它可能是继bash和Korn shell之后第三个最流行的shell
ksh, pdksh	korn shell和它的公共域兄弟pdksh (public domain korn shell) 由David Korn编写，它是许多商业版本UNIX的默认shell
bash	来自GNU项目的bash或Bourne Again Shell是Linux的主要shell。它的优点是 <code>可以免费获取其源代码</code> ，即使你的UNIX系统目前没有运行它，它也很可能已经被移植到该系统中。bash与Korn shell有许多相似之处





UNIX类系统常见的shell

Common shells used on UNIX systems *					
Name	Path	FreeBSD 5.2.1	Linux 2.4.22	Mac OS X 10.3	Solaris 9
Bourne shell	/bin/sh	√	link to bash	link to bash	√
Bourne-again shell	/bin/bash	optional	√	√	√
C shell	/bin/csh	link to tcsh	link to tcsh	link to tcsh	√
Korn shell	/bin/ksh		√		√
TENEX C shell	/bin/tcsh	√	√	√	√

*: Advanced Programming in the UNIX® Environment: Second Edition





常用的shell

shell名称	存放的位置	程序名
Bourne shell	/bin/sh->bash	bash
Bourne Again shell	/bin/bash	bash
C shell	/bin/csh->tcsh	tcsh
TC shell	/bin/tcsh	tcsh
Korn shell	/bin/ksh	ksh

■ Shell命令可以被分为内部(内置)命令和外部命令。

- 内部命令是shell本身包含的一些命令，这些内部命令的代码是整个shell代码的一个组成部分；
- 内部命令，shell是通过执行自己代码中相应的部分来完成的
- 外部命令的代码则存放在一些二进制的可执行文件或者shell脚本中
- 外部命令，shell会到文件系统结构（file system structure）中的一些目录去搜索那些文件名与外部命令的名字相同的文件，因为shell认为这些文件中就存放了将要执行的代码。





Shell命令搜索路径

- Shell搜索的目录的名字都保存在一个shell变量**PATH**（在TC shell中是path）中。
- 变量**PATH**（或者path）中的目录名用一些特定的符号分开。在**bash shell**中，目录名用**冒号**分开。
- **\$ echo \$PATH**
`/usr/local/globus/bin:/usr/local/globus/sbin:/usr/java/j2sdk1.4.1_01/bin:/usr/local/apache-ant-1.5.4/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/opt/hdf5-oscar-1.4.4-post2/bin:/opt/pbs/bin:/opt/pbs/lib/xpbs/bin:/opt/pvm3/lib:/opt/pvm3/lib/LINUX:/opt/pvm3/bin/LINUX:/opt/env-switcher/bin:/opt/lam-6.5.9/bin:/root/bin`
- 变量**PATH**（或者path）保存在**主目录**中的隐藏文件（hidden file）**.profile**或者**.login**中





shell的环境变量

- 对于普通计算机用户，控制台窗口的属性和外观，以及解释命令的环境，都是由系统管理员设定的。环境设置是由**环境变量**控制的。环境变量在用户登录时得到，或事先设定默认值。当用户键入命令或执行其他重要操作时，环境决定了使用哪种**shell**或命令行解释器。

- 显示**shell**的路径：

echo \$SHELL

注意大小写

- 查看环境变量值：

set

- 设置环境变量值：

变量名=值





shell的环境变量

- 在Linux系统的终端中有几个最有用的**bash**变量
 - **HISTFILE**: 用于贮存历史命令的文件。
 - **HISTSIZE**: 历史命令列表的大小。
 - **HOME**: 当前用户的主目录。
 - **OLDPWD**: 前一个工作目录。
 - **PATH**: **bash**寻找可执行文件的搜索路径。
 - **PS1**: 命令行的一级提示符。
 - **PS2**: 命令行的二级提示符。
 - **PWD**、**cwd**: 当前工作目录。
 - **SECONDS**: 当前**shell**开始后所流逝的秒数





bash中的一些启动文件

表 bash中的一些启动文件	
文件名	功能描述
/etc/profile	登录时自动执行
~/.bash_profile, ~/.bash_login, ~/.profile	登录时自动执行
~/.bashrc	Shell登录时自动执行
~/.bash_logout	退出时自动执行
~/.bash_history	记录最近会话中的命令
/etc/passwd	



这些文件
都是系统
启动、用
户登录、
退出时自
动执行的

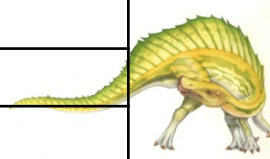




shell元字符

- **shell元字符(shell metacharacters)**：除了字母和数字，其它有特殊的含义的大部分字符。

表 Shell 中的元字符		
元字符	作用	例子
回车换行	结束一个命令行	
空格	分割命令行中的元素	ls /etc
Tab	分割命令行中的元素	ls /etc
#	开始一行注释	# this is a comment line
“	引用多个字符但是允许替换（15章）	“\$file”.bak
\$	表示一行的结束以及显示变量的值	\$PATH
&	让一个命令在后台执行	command &
‘	引用多个字符	‘\$100,000’
()	在子shell中执行命令	(command1;command2)
*	匹配0个或者多个字符	chap*.ps
[]	插入通配符	[a-s] 或者 [1,5-9]
^	表示一行的开始以及作为否定符号	[^3-8]
`	替换命令	PS1=`command`
{ }	在当前shell中执行命令	{command1;command2}
	创建命令间的管道	command1 command2
;	分割顺序执行的命令	command1;command2





shell元字符(续)

<	重定向命令的输入	command<file
>	重定向命令的输出	command>file
?	匹配单个字符	lab.?
/	用作根目录或者路径名中的分割符	/usr/bin
\	转义字符；转义回车换行字符，允许在下一行中继续shell命令	command arg1\ arg2 arg3 \?
!	启动历史记录列表中的命令和当前命令	!! , !4
%	TC shell的提示符，或者指定一个任务号时作为起始字符	% 或者 %3
~	代表主目录	~/.profile





通配符

- 常用通配符：*、?、[]、-
 - “[]-”用于构成字符组模式，如：
 - ▶ [abc] 表示匹配a、b、c中一个字符
 - ▶ [a-z] 表示匹配小写英文字母中的一个字符
- 通配符扩展:花括号{ }通配符,允许将任意的字符串分组放在一个集合中，以供shell进行扩展。
 - 还可以使用花括号扩展用相关的名字创建子目录：

```
$ ls -F
file1 fi1e2 fi1e3
$ mkdir dir{A,B,C,D,E}
$ ls -F
file1 fi1e2 fi1e3 dirA/ dirB/ dirC/ dirD/ dirE/
```





学会使用<tab>键

■ <tab>键:命令行补全

`cd d<tab>` ==> `cd dir1`

- 把当前目录设置为当前目录下d开头的子目录





例

- 例：字符串“**?.txt**”可以用来表示一个字符后跟“**.txt**”的所有文件名，如：**a.txt**, **1.txt**, **@.txt**。
- 例：字符串**[0-9].c**用来表示所有文件名为单个数字后跟“**.c**”形式的文件，如：**1.c**和**3.c**。
- 例：字符串**lab1 \ / c**表示**lab1/c**。注意，在这里，我们用反斜线号（\）来处理“消除了特殊意义”的斜线号（/）。
- 例：下面的这条命令显示当前目录中所有由**2**个字符组成，且以**.html**为结尾的文件。而且这些文件名的第一个字符是数字，第二个字符是大写或者小写的字母。
 - **\$ ls [0-9][a-zA-Z].html**





命令语法结构

■ 命令行输入的语法结构：

\$ command [[-]option(s)] [option argument(s)] [command argument(s)]

- \$是来自操作系统的**提示符**，你的计算机可能不一样。
- 任何括在[]中的内容都不是必需的
- **command**是**LINUX**命令（不同的**shell**有所区别），小写（**命令名**）
- **[-option(s)]**是定制命令动作的一个或多个修饰符号（**选项**）
- **[option argument(s)]**是定制选项动作的一个或多个修饰符号（**选项的参数**）
- **[command argument(s)]** 受命令影响的一个或者多个对象（**命令的参数**）





例

- **\$ ls**
- **\$ ls -la**
- **\$ ls -la m***
- **\$ lpr -Pspr -n 3 proposal.ps**
 - 命令**lpr**，两个选项：**P**和**n**,作为选项的操作对象的两个选项参数：**spr**和**3**，以及命令参数：**proposal.ps**。
 - 这些项都是区分大小写的。
 - 一个选项和它的参数间是有空格的，但是另一个选项和它的参数间是没有空格的





常用的几个命令

- **passwd** :修改密码命令
- **su** **改变**用户身份
- **sudo** 以**root**用户权限运行命令
- 与用户有关的其他命令
 - **useradd/adduser** 添加用户
 - **userdel/deluser** 删除用户
- **man、info** : 获取帮助命令
 - **man passwd** #passwd命令的帮助手册出现在屏幕上
 - **info passwd**
 - **man read** #read系统调用的帮助手册
 - **ls --help**





常用的几个命令

- **whatis:** 得到任何**LINUX**命令的更短的描述命
- **whoami:** 显示用户名
- **hostname:** 显示登录上的主机的名字
- **uname:** 显示关于运行在计算机上的操作系统的信息
- **id:** 显示用户**id**和组**id**等信息

Linux常用命令查询: <http://zh.wikipedia.org/zh-cn/Unix实用程序列表>





常用命令

- 列出文件名和显示工作目录
 - **ls *.c**
- 路径设置
 - **PATH=~/bin:\$PATH:.** 搜索路径中增加~/bin和.目录
- 创建和显示目录
 - **pwd** print working directory, 显示工作目录
 - **mkdir** make directory, 创建目录
 - **rmdir** remove directory, 删除目录
 - **cd** change directory, 改变当前的工作目录
- 两个特殊目录
 - “.”代表当前工作目录
 - “..”代表当前目录的父目录





例:

```
$ pwd
/home/faculty/sarwar
$ ls -aC
. .. .cshrc .login ece 231 ece345 ece441 ece445 ece446 personal linuxbook
$ cd linuxbook
$ cd examples
$ pwd
/home/faculty/sarwar/linuxbook/examples
$ ls -C
chapter1 chapter2 chapter3 chapter4
$ mkdir dir1
$ ls -C
chapter1 chapter2 chapter3 chapter4 dir1
$ cd dir1
$ pwd
/home/faculty/sarwar/linuxbook/examples/dir1
$ cd ..
$ rmdir dir1
$ ls -C
chapter1 chapter2 chapter3 chapter4
$ cd ~
$ pwd
/home/faculty/sarwar
```





常用命令（续）

■ 显示文件内容

- **cat** 同时显示一个或多个文件的内容

\$ **cat sample** 显示sample文件的内容

- **more**、**pg** 一次显示一个屏幕的内容

\$ **more sample phones** 一次显示一屏幕的sample文件的内容，然后以同样的方式显示phones文件的内容

■ 显示日历命令

- **cal [[month] year]**





常用命令（续）

■ 为命令创建假名（别名）

- **alias** **[name[=string]...]** 为“name”命令建立别名“string”
- **unalias** 删除别名

■ 显示系统运行时间命令 **uptime**

\$ uptime

11:39 AM up 7:04, 11 users, load average: 0.08, 0.12, 0.17

■ 清楚屏幕命令 **clear**

■ 改变用户的身份

- 命令语法: **su** **[-i][-c <command>]** **[username]**
- 常用选项/参数:

-c < command > 执行完指定的指令后，即恢复原来的身份。

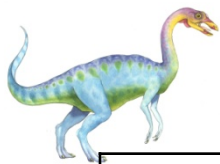
- 改变身份时，也同时变更工作目录，及HOME、SHELL、PATH变量等
username 指定要变更的用户名。若不指定此参数，则为root用户。

- 例：下面su命令将你的用户身份转换为root，当然，你必须输入root密码。

\$ su

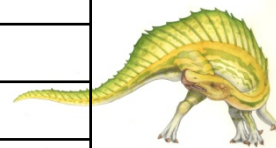
Ubuntu隐含root账号，使用**su passwd root**
命令设置root的密码后，才能使用su命令





常用命令

命令	功能
<Ctrl-C>	终止当前的命令或程序
<Ctrl-D>	结束输入，或退出Linux系统，或从上层shell返回
<Ctrl-Z>	暂停当前命令执行
cp	拷贝文件
echo \$SHELL	显示正在运行的shell名字
exit	结束当前的shell
hostname	显示你所登录的主机的名字
login	用一对正确的用户名/密码登录到计算机系统
logout	退出登录
ls	显示文件和目录的信息
man	浏览关于一个命令或主题的帮助手册
mv	移动或重命名文件
passwd	修改密码
set	在bash中显示和修改环境变量
uname	显示关于计算机正运行的操作系统的信息
w	比who命令更加详细地列出系统上用户的信息
whatis	显示一个命令的简要描述
whereis	在标准路径（非用户指定的路径）下搜索与指定命令相关文件的全路径
which	当某个工具或程序有多个副本时，用which来识别哪个副本在运行
who	显示现在正在使用系统的用户的信息
whoami	显示你的用户名





Linux的编辑器

■ 文本界面下，常用的文本编辑器有：

- **vi**编辑器：**UNIX**类操作系统通用的全屏幕编辑器，只要你习惯于操作，你会觉得它比任何的编辑器都好用，且功能强大。
- **vim**编辑器：**vi**的增强版本，是**vi**的克隆，是**基于GUN**软件。
- **pico**和**nano**编辑器：一种风格很像**Microsoft DOS**的文本编辑器。一些发行版没有安装。
- **emacs**编辑器：**GNU**编辑器，功能强大的全屏幕编辑器。

■ 图形界面下，常用的文本编辑器有：

- **emacs**编辑器：编程编辑器
- **gedit**或**kedit**编辑器：全屏幕文本编辑程序

操作方便，
建议使用

- ## ■ 要区分文本编辑器和排版工具不同，文本编辑器不象Openoffice、Word或WPS那样可以对字体、格式、段落等其他属性进行编排。





vi 编辑器

- **vi**是Linux/Unix世界里最常用的全屏编辑器，所有的Linux系统都提供该编辑器，而Linux也提供了vi的加强版——vim，同vi是完全兼容，存放路径为/usr/bin/vim，vim软件及有关信息可以从www.vim.org获得。
- 多数的Linux系统中vi命令是vim的别名，你可以通过alias命令或which vi命令查看一下，所以，当您启动vi命令时，实际运行是vim程序。在本节内容中，我们不对vi和vim加以区别，统一使用vi命令。





vi 编辑器(续)

■ vi有两种操作方式，分别是：

- 命令模式（**command mode**），由击键命令序列（**vi编辑器命令**）组成，完成某些特定动作；
- 插入模式（**insert mode**），允许你输入文本。

图 vi文本编辑器的操作模式

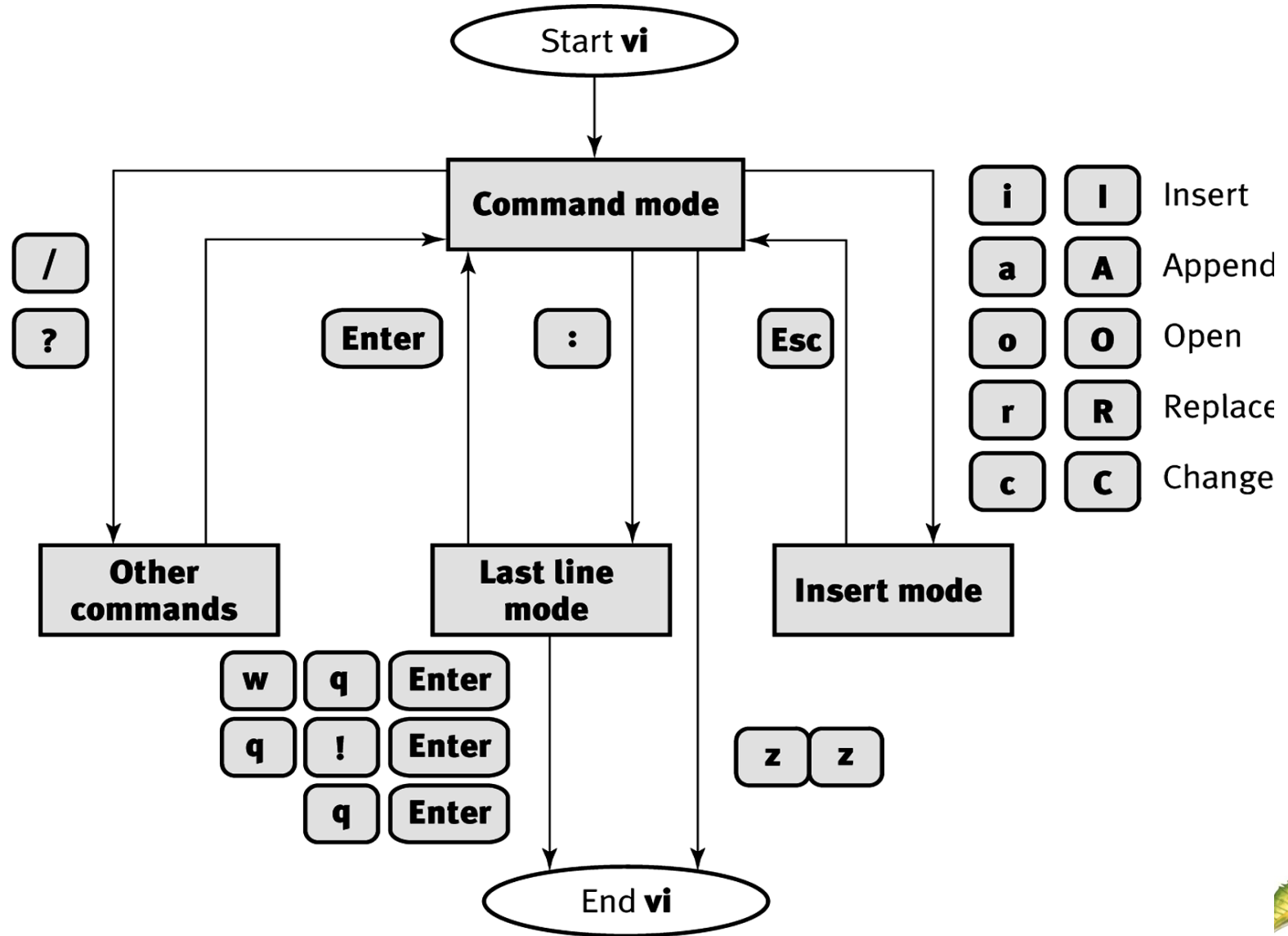
■ vi的进入与离开

- 在系统提示符“\$”下键入命令**vi**，后面跟上想要编辑（或者建立）的文件名，**vi**可以自动载入所要编辑的文件或是开启一个新文件。
- **vi**的退出，可以在命令模式使用命令“**:wq**”或者“**:q!**”，前者的功能是写文件并从**vi**中退出，后者的功能是从**vi**中退出，但不保存所作的修改（注意冒号）。





vi 编辑器(续)



图vi文本编辑器的操作模式





vi 编辑器(续)

■ vi的插入模式:

- 在命令模式下正确定位光标之后，可用一下命令切换到插入模式：表 插入模式下的重要按钮。
- 如果用户想利用已有的文件内容，可以使用命令“: i filename”，则vi将指定文件的内容输入当前光标的下一行，且vi仍处于命令模式。
- 退出插入模式的方法是，按ESC键或组合键Ctrl+l

■ vi的命令模式:

- 表 命令模式下的重要命令
- 表 光标移动和键盘编辑命令
- 表 复制和粘贴文本命令yank和put

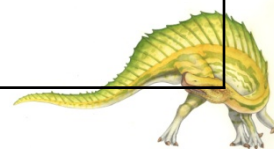




vi 编辑器(续)

- 表列出了在**命令模式**和**插入模式**下都可用的一般语法及其变形的具体示例。

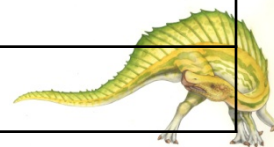
表	vi命令语法示例
命令	动作
5dw	从当前光标位置开始删除 5 个字
7dd	从当前行开始删除 7 行
7o	在当前行后面开辟 7 个空行
7O	在当前行前面开辟(插入) 7 个空行
c2b	修改光标前面 2 个字
d7,14	将缓冲区中第 7 行至 14 行删除
1G	将光标置于文件首行
10yy	将后面 10 行（从当前行开始）拷贝到临时缓冲区中





vi 编辑器(续)

表	插入模式下的重要按键
按键	行为
<a>	在光标所在字符后添加文本
<A>	在当前行最后一个字符后添加文本
<c>	开始修改操作，允许你更改当前行文本
<C>	修改从光标位置开始到当前行末尾范围内的内容
<i>	在光标所在字符前插入文本
<I>	在当前行开头插入文本
<o>	在当前行下方开辟一空行并将光标置于该空行行首
<O>	在当前行上方开辟一空行并将光标置于该空行行首
<R>	开始覆盖文本操作
<s>	替换单个字符
<S>	替换整行





vi 编辑器(续)

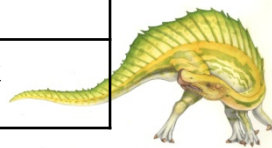
表	命令模式下的重要命令
命令	动作
d	删除字、行等
u	撤销最近一次编辑动作
p (小写)	在当前行后面粘贴 (插入) 此前被复制或剪切的行
P (大写)	在当前行前面粘贴 (插入) 此前被复制或剪切的行
:r filename	读取 filename 文件中的内容并将其插入在当前光标位置
:q!	放弃缓冲区内容, 并退出 vi
:wq	保存缓冲区内容, 并退出 vi
:w filename	将当前缓冲区内容保存到 filename 文件中
:w! filename	用当前文本覆盖 filename 文件中的内容
ZZ	退出 vi , 仅当文件在最后一次保存后进行了修改, 才保存缓冲区内容





vi 编辑器(续)

表	光标移动和键盘编辑命令
命令	动作
<1G>	将光标移到文件首行
<G>	将光标移到文件末行
<0>(数字0)	将光标移到当前行首个字符
<Ctrl-G>	以行列号形式报告光标位置
<\$>	将光标移到当前行最后一个字符
<w>	将光标每次前移一字
	将光标每次倒退一字
<x>	删除光标位置上的字符
<dd>	删除当前光标所在行
<u>	撤销最近一次所做的修改
<r>	用随后键入的一个字符替换当前光标位置处的字符





vi 编辑器(续)

表	yank 和put命令的语法示例
命令语法	完成的动作
y2w	从当前光标位置开始向右复制2个字
4yb	从当前光标位置开始向左复制4个字
yy或Y	复制当前行
p (小写)	在当前光标位置后插入复制的文本
P (大写)	在当前光标位置前插入复制的文本
5p	在当前光标位置后将缓冲区中复制的文本粘贴5次





vi 编辑器(续)

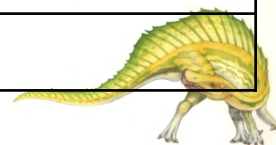
■ **Substitute (搜索和替换), Substitute command format:**

: [range]s/old_string/new_string[/option]

- ▶ []中的部分是可选的;
- ▶ : 是状态行命令的冒号前缀;
- ▶ range是缓冲区中有效行的范围指定 (如果省略, 当前行就是命令的作用范围);
- ▶ s 代表**substitute**命令;
- ▶ / 是查找的分隔符;
- ▶ old_string 是你想要替换掉的文本;
- ▶ / 是替换的分隔符;
- ▶ new_string 是替换上去的新文本;
- ▶ /option 是命令的修饰选项, 通常用g代表全局。

● **Practice Session 5.5**

Table	Subtitute命令的语法示例
命令语法	完成的动作
:s/john/jane/	在当前行用字 <i>jane</i> 替换字 <i>john</i> , 只替换一次
:s/john/jane/g	在当前行用字 <i>jane</i> 替换所有的字 <i>john</i>
:1,10s/big/small/g	在第1至第10行用字 <i>small</i> 替换所有的字 <i>big</i>
:1,\$s/men/women/g	在整个文件中用字 <i>women</i> 替换所有的字 <i>men</i>





vi 编辑器(续)

■ 从vi中执行shell命令：

- 命令模式下通过在命令前加:**!** 来实现。执行完一个**shell**命令后，**vi**回到它的命令模式。
- 例如：
 - ▶ 键入:**! pwd**会显示你当前目录的路径名
 - ▶ 键入:**! ls**会显示你当前目录下的所有文件名。





emacs编辑器

- **emacs**文本编辑器可以用来编辑文本、剪辑和粘贴文本内容、提供个人日历和日记，阅读**usenet**新闻、发送电子邮件，同时还是一种程序语言解释器，可以编辑**C**、**Lisp**、**Tev**源代码文件、以及**Linux**的**Shell**。
- **emacs**是由 **Richard Stallman**发明的。是一个**GNU**的编辑器。**emacs**的主页为
www.gnu.org/software/emacs/emacs.html。
- 最初的**emacs**是用来编辑宏命令的，现已进一步扩充为**UNIX**用户中装机用户数量最大、功能最齐全的免费文本编辑器了。





小结

■ 本章需要掌握：

- 学会安装Linux系统
- Linux的登录和退出
- 要求在文本界面和图形界面下分别掌握一种编辑器的简单使用
- 命令： **man、info、su、whoami、hostname、uname、who、whatis、whereis、uptime、cal、echo、pwd、reboot**等
- 知识： **shell**

■ 本章需要了解：

- 命令： **cp、exit、login、ls、mv**
- 知识： 元字符、环境变量





思考题

- 使用**Linux**的终端（命令行界面）有很多多种方式，请查找相关资料依次列出，并尽可能实践之。





实验和作业

■ 实验1



End of chapter

