



## 第2章 图形界面与命令行

主讲：王洪泊





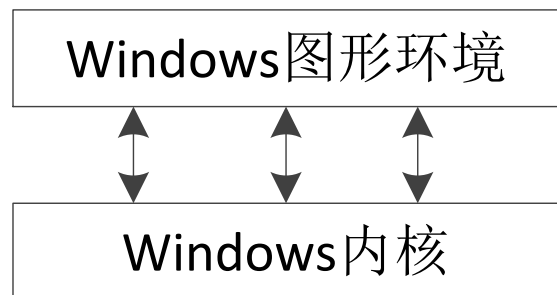
# 本章学习要点

- (1) **Linux**图形界面
- (2) **Linux**命令行界面
- (3) **Shell**基础
- (4) **Linux**命令行使用
- (5) 使用**vi**编辑器



## 2.1 Linux图形界面

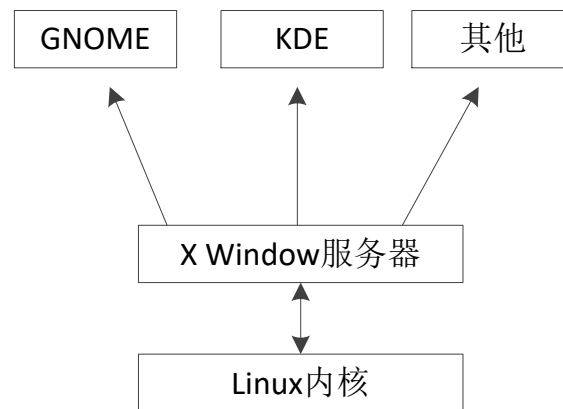
### Windows与 Linux 图形架构



X客户端

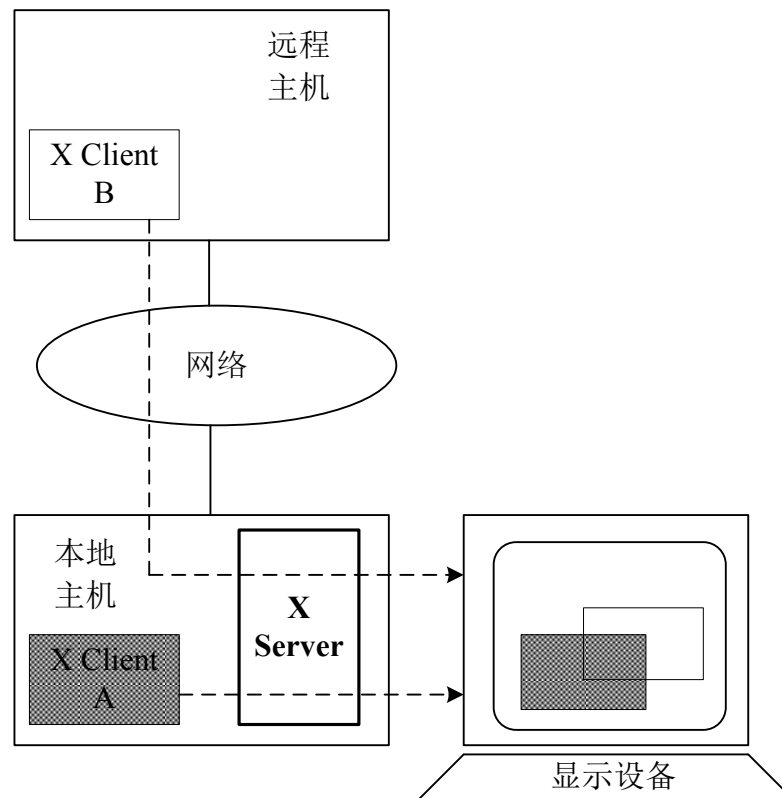
X协议

X服务器



## 2.1 Linux图形界面

### 2.1.1 X Window System





## 2.1 Linux图形界面

### 2.1.2 窗口管理器 (Window Manger)

X Window System只是提供了建立窗口的一个标准，具体的窗口形式由窗口管理器决定。

窗口管理器是X Window System的组成部分，用来控制窗口的外观，并提供与用户交互的方法。

作为在X Server上运行的应用程序，窗口管理器为用户提供操作窗口程序的方法。

窗口管理器是与X Server进行交互控制的服务器端程序，也可以将它看作是一种特殊的X客户端程序，其功能都是通过向X服务器发送命令来实现的。





## 2.1 Linux图形界面

### 2.1.3 显示管理器 (Display Manager)

显示管理器提供图形界面用户登录，用于验证用户身份，让用户登录到系统。

图形界面由显示管理器启动，包括X Server。

系统启动之后第一个要启动的X Client程序是DM，DM也是一类软件，如XDM、KDM、GDM。





## 2.1 Linux图形界面

### 2.1.4 Widget库

Widget库定义一套图形用户界面的编程接口。

应用程序开发人员通过调用该库来实现具体的用户界面，包括按钮、文本框、菜单、滚动条等。

程序员不需要理解X Server的语言，由Widget库将调用提交给X Server即可。



## 2.1 Linux图形界面

### 2.1.5 桌面环境 (Desktop Environment)

开发人员在窗口管理器的基础上，增加各种功能和应用程序，提供更为完善的图形用户环境，这就是桌面环境。

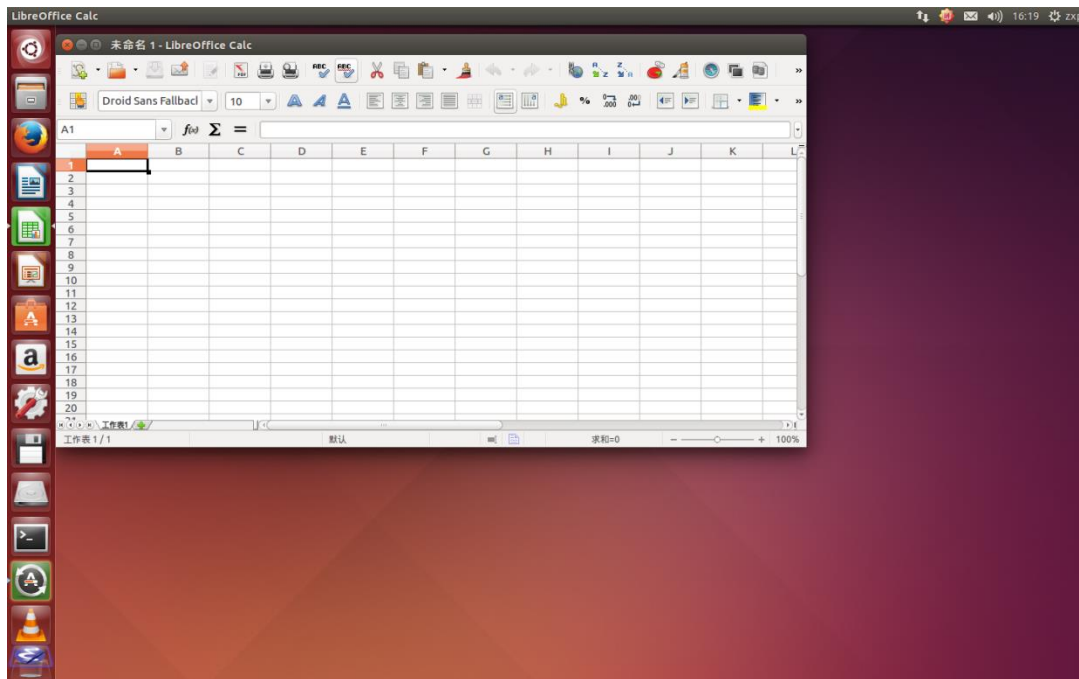




## 2.1 Linux图形界面

### 2.1.6 Unity桌面环境

与GNOME、KDE不同，Unity并非一个完整桌面程序安装包，主要实现的是桌面环境的面板部分，窗口管理器等其他桌面环境要素采用了现有的方案。



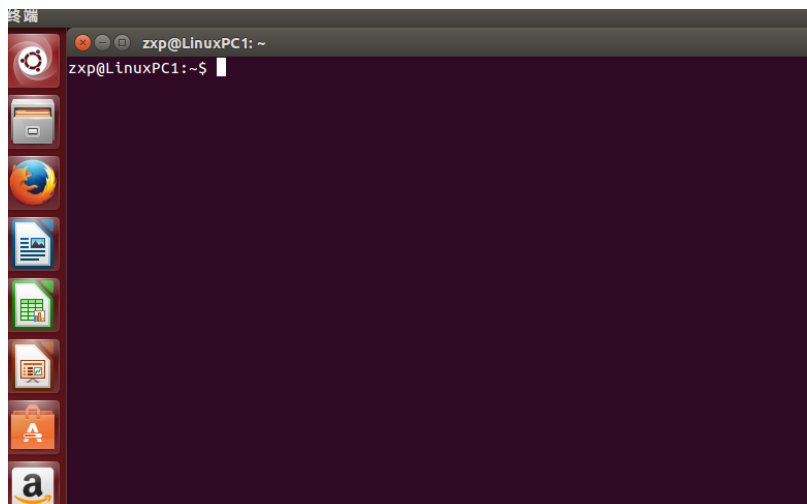
## 2.2 Linux命令行界面

### 2.2.1 使用仿真终端窗口

- 打开终端控制台

- 使用组合键<Ctrl>+<Alt>+<T>。
- 在Dash中浏览搜索“终端”程序。
- 在文件系统中搜索“gnome-terminal”。

建议将终端应用程序图标锁定到启动器，以便于今后通过快捷方式运行。



## 2.2 Linux命令行界面

### 2.2.1 使用仿真终端窗口

#### ● 配置终端





## 2.2 Linux命令行界面

### 2.2.2 使用文本模式

Ubuntu 桌面版启动之后直接进入图形界面，然后可以根据需要切换到文本模式，此时需要登录到Linux系统。

Linux系统允许用户同时打开6个虚拟控制台（tty1~tty6）进行操作，每个控制台可以让不同用户身份登录，运行不同的应用程序。

```
Ubuntu 14.10 LinuxPC1 tty1
LinuxPC1 login: zxp
Password:
Welcome to Ubuntu 14.10 (GNU/Linux 3.16.0-23-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

67 packages can be updated.
67 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

zxp@LinuxPC1:~$ _
```



## 2.2 Linux命令行界面

### 2.2.3 使用命令行关闭和重启系统

**Linux**只有**root**用户才能执行关机或重启命令。

- 执行**reboot**命令重启

`sudo reboot`

- 执行**shutdown**命令关机

要立即关机，执行以下命令

`shutdown -h now`

执行以下命令表示10分钟之后关机，并向用户给出提示

`shutdown +10 "System will shutdown after 10 minutes"`



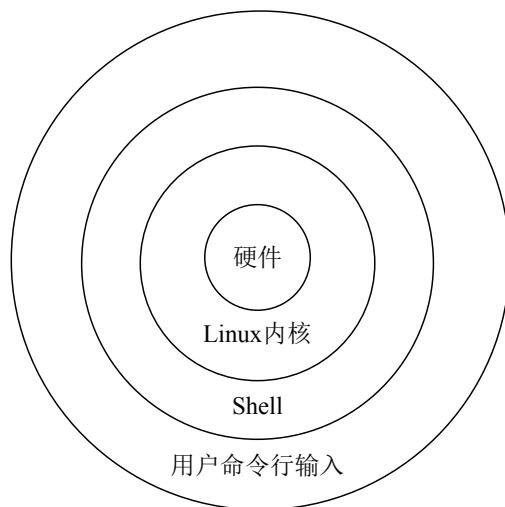


## 2.3 Shell基础

### 2.3.1 什么是Shell

#### ● Linux Shell主要功能

- 解释用户在命令行提示符下输入的命令。这是最主要的功能。
- 提供个性化的用户环境，通常由Shell初始化配置文件实现。
- 编写Shell脚本，实现高级管理功能。





## 2.3 Shell 基础

### 2.3.2 使用Shell

用户进入Linux命令行时，就已经自动运行一个默认的Shell程序。

用户可看到Shell的提示符，用户在提示符后输入一串字符，Shell将对这一串字符进行解释。输入的这一串字符就是命令行。

Ubuntu默认使用的Shell程序是bash。

要改变当前Shell程序，只需在命令行中输入Shell名称。





## 2.3 Shell 基础

### 2.3.3 正则表达式

- 通配符

\*（星号）：表示任何字符串。

?（问号）：表示任何单个字符。

[]（一对方括号）：表示一个字符序列，字符序列可以直接包括若干字符，也可以是由“-”连接起止字符形成的序列。

！（感叹号）：在[]中使用!表示排除其中任意字符。

^（幂符号）：只在一行的开头匹配字符串。

\$（美元符号）：只在行尾匹配字符串，它放在匹配单词的后面。







## 2.3 Shell 基础

### 2.3.3 正则表达式

- 模式表达式符

\*: 匹配任意多个模式。

+: 匹配1个或多个模式。

? : 匹配模式表中任何一种模式。

@: 仅匹配模式表中一个给定模式。

!: 除给定模式表中的一个模式之外，它可以匹配其他任何字符串。





## 2.3 Shell 基础

### 2.3.4 Shell 中的特殊字符

- 引号

单引号 (') 括起来的字符串视为普通字符串。

双引号 (") 括起来的字符串，除 \$、\、单引号和双引号仍作为特殊字符并保留其特殊功能外，其他都视为普通字符对待。

反引号 (`) 起来的字符串被 **Shell** 解释为命令行，在执行时首先执行该命令行，并以它的标准输出结果替代该命令行。





## 2.3 Shell 基础

### 2.3.4 Shell 中的特殊字符

- 其他符号

- #（注释）

- \（跳转符号，将特殊字符或通配符还原成一般字符）

- |（分隔两个管道命令）

- ；（分隔多个命令）

- /~（用户的主目录）

- \$（变量前需要加的变量值）

- &（将该符号前的命令放到后台执行）





## 2.3 Shell 基础

### 2.3.5 环境变量

- 常用的环境变量

PATH（可执行命令的搜索路径）

HOME（用户主目录）

LOGNAME（当前用户的登录名）

HOSTNAME（主机名）

PS1（当前命令提示符）

SHELL（用户当前使用的Shell）

- 使用**env**命令可显示所有的环境变量
- 要引用某个环境变量，在其前面加上\$符号
- 要修改某个环境变量，则不用加上\$符号





## 2.4 Linux命令行使用

### 2.4.1 命令语法格式

- 命令格式如下：  
提示符 命令 选项 参数
- 有的命令不带任何选项和参数。Linux命令行严格区分大小写，命令、选项和参数都是如此。
- 选项是包括一个或多个字母的代码，前面有一个“-”连字符，主要用于改变命令执行动作的类型。
- 参数通常是命令的操作对象，多数命令都可使用参数。
- 同时带有选项和参数的命令，通常选项位于参数之前。





## 2.4 Linux命令行使用

### 2.4.2 命令行基本用法

- 编辑修改命令行
- 调用历史命令
- 自动补全命令
- 一行多条命令和命令行续行
- 强制中断命令运行
- 获得联机帮助





## 2.4 Linux命令行使用

### 2.4.3 命令行输入与输出

- 输入重定向

命令 < 文件名

- 输出重定向

命令 > 文件名

- 管道

命令1 | 命令2 ..... | 命令n

- 命令替换

命令1 `命令2`





## 2.4 Linux命令行使用

### 2.4.4 执行Shell脚本

- **Shell脚本**

Shell脚本是指使用Shell所提供的语句所编写的命令文件。

Shell脚本最基本的功能就是汇集一些在命令行输入的连续指令，将它们写入脚本中，而由直接执行脚本来启动一连串的命令指令。

- **Shell脚本执行**

将Shell脚本的权限设置为可执行，然后在提示符下直接执行它。

直接编辑生成的脚本文件没有执行权限，要当做命令执行，就需要利用命令`chmod`将它置为具有执行权限。例如

```
chmod +x example1
```



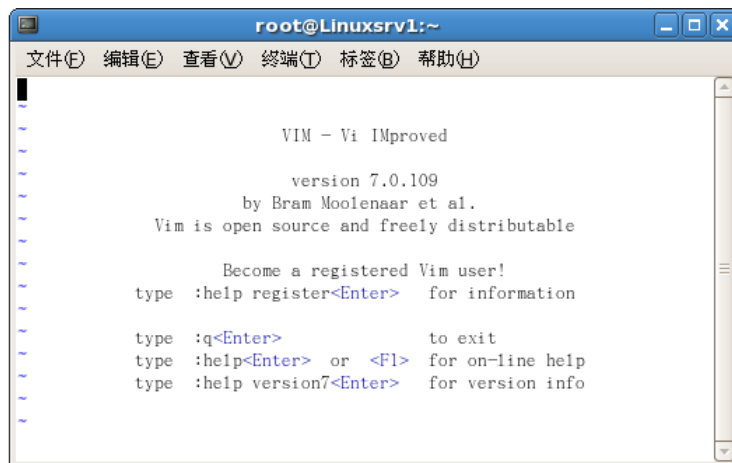




## 2.5 使用vi编辑器

### vi操作模式

- 命令模式（Command mode）：输入的任何字符都作为命令处理
- 插入模式（Insert mode）：输入的任何字符都作为插入字符处理
- 末行模式（Last line mode）：执行文件级或全局性操作，如保存文件、退出编辑器，设置编辑环境等



```
root@Linuxsrv1:~  
文件(E) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)  
  
      VIN - Vi IMproved  
  
      version 7.0.109  
      by Bram Moolenaar et al.  
      Vim is open source and freely distributable  
  
      Become a registered Vim user!  
type  :help register<Enter>   for information  
  
type  :q<Enter>               to exit  
type  :help<Enter> or <F1>    for on-line help  
type  :help version7<Enter>  for version info
```



## 2.5 使用vi编辑器

### Vi基本操作

- 打开vi编辑器
- 编辑文件
- 保存文件和退出
- 其他全局性操作
- 多文件操作





# 预习与巩固思考题

1. 简述X Windows System 的工作原理。
2. 窗口管理器有什么作用？
3. 完整的图形桌面环境至少包含哪些程序？
4. Unity 桌面环境有什么优点？
5. 为什么要学习命令行？
6. 什么是Shell, 它有什么作用？
7. 简述命令行命令语法格式。
8. 如何强制中断命令运行？
9. 管道有什么作用？
10. 使用vi编辑器编辑一个文本文件，熟悉基本编辑方法。

