



张学工 刘业新 编著

X WINDOW/MOTIF 编程速成



清华大学出版社 <http://www.tup.tsinghua.edu.cn>



TP316
1270

X Window/Motif 编程速成

张学工 刘业新 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

X Window/Motif 是计算机工作站、大型机和采用 UNIX 操作系统的微机上最主要的窗口系统,目前已经成为工作站应用软件的用户界面标准。本书从广大 UNIX 用户和软件开发者的实际需要出发,采用循序渐进的方式介绍了 X Window 和 Motif 应用程序的编程方法,内容按照一般程序员学习的自然方式编排,采用了大量根据实际应用程序改编的程序实例,使读者可以在较短的时间内就能够开始设计自己的应用程序。为了兼顾初学者的需要,书中还对 UNIX 系统的有关基本知识和 UNIX 环境下 C 语言程序的设计和调试方法进行了扼要的介绍。本书的读者对象主要是各行业的工作站和 UNIX 系统用户以及 UNIX 系统下的应用软件开发人员,也包括需学习 X Window 和 Motif 的高等院校学生和科研人员。本书既可用于学习 X Window/Motif 编程的速成教材,也可作为实用的编程参考手册。

版权所有,翻印必究。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

X Window/Motif 编程速成/张学工,刘业新编著. — 北京:清华大学出版社,1998
ISBN 7-302-02865-6

I. X… II. ①张… ②刘… III. 窗口软件,X-程序设计 IV. TP316

中国版本图书馆 CIP 数据核字(98)第 02693 号

出版: 清华大学出版社(北京清华大学校内,邮编 100084)

因特网地址: www.tup.tsinghua.edu.cn

印刷: 清华园胶印厂

发行: 新华书店总店北京科技发行所

开本: 787×1092 1/16 **印张:** 26 **字数:** 648 千字

版次: 1998 年 3 月第 1 版 1998 年 7 月第 2 次印刷

书号: ISBN 7-302-02865-6/TP·1512

印数: 3001~6000

定价: 32.00 元

前 言

目前,X窗口(X Window)和 Motif 系统已成为计算机工作站和大型计算机上主要的图形用户界面系统,在微机上也有广泛应用。本书是一本介绍如何在 X 窗口和 Motif 下编写图形界面应用程序的实用教材和工具书,其中如何编写 Motif 应用程序是本书的重点。通过阅读本书,读者可了解和掌握有关 X 窗口和 Motif 的基本知识和基本编程方法,并能够编写常用的图形界面程序。X 窗口和 Motif 是一个十分复杂的系统,本书并不试图完全系统地介绍这一复杂系统,而是从实用角度出发,以帮助读者快速入门为目的,按照多数用户和程序开发者的接受习惯,采用剖析大量实例程序的方法由浅入深地教读者用 X 窗口和 Motif 编写自己的图形界面程序。

从实例中学习,是多数计算机用户和程序开发者所喜欢的和习惯的方式,因此,本书中尽量采用了这种方式,无论是介绍一些基本的操作和使用方法,还是介绍如何编写某类应用程序,都采用一些实例,力求使读者从本书中学习如同向一位热心的朋友学习一样。

本书力求实用性的另一特点是,除了一般地介绍编程方法外,书中还针对一些学习编程中容易出现的实际问题给出了及时的提示,这些问题包括不同系统和不同版本下的一些细微的差别、编程中容易出现的一些常见错误和初学者容易忽视的一些细节、更有效地学习编程的方法建议等等,这些内容是一般 X 窗口和 Motif 编程教材中所不包括的。另外,书中采用的大量源程序中,除部分是 X 窗口系统和 OSF/Motif 的标准例子或根据标准例子改编的实例外,其余多数是作者从自己的实际工作中提炼整理出来的,且均已在多种常用工作站平台上通过了调试,读者可以直接在此基础上开发自己的程序,这也是本书与其它同类教材的一个重要不同。

为了那些希望更系统地学习 X 窗口和 Motif 的读者的方便,书后给出了本书的主要参考文献和对读者进一步阅读的建议。同时,在本书的最后,我们专门给出了关于本书中介绍和涉及的关键概念和函数的索引,读者可以快速地查到希望了解的内容,因此,本书也可以作为一般 X Window/Motif 编程的简明参考手册使用。

各章节内容简介

本书分为七章,基本内容如下:

X 窗口和 Motif 应用程序是在 UNIX 系统下用 C 语言编写的,为了对 UNIX 系统不十分熟悉的读者能够尽快地掌握所需的有关基础知识,本书首先在第一章中简要介绍了与 X 窗口/Motif 编程关系较密切的一些 UNIX 系统和 C 语言的使用知识。这些知识分为两大部分:第一部分包括 UNIX 系统的常用命令、用户的基本环境设置、如何使用 vi 进行文本编辑、如何使用 man 随时获得联机帮助信息和有关文档等;第二部分则介绍 C 语言的基本知识、如何在 UNIX 系统下进行程序的编译连接、如何用 dbx 对程序进行调试、如何混合使用 C 语言和 FORTRAN 语言以及如何编写较好的 C 语言程序等。这些都是读者在学习用 X Window/Motif 编程时首先遇到的问题。

第二章介绍 X Window 和 Motif 的基本概念和使用方法,以及基本的 Motif 图形界面风格等。除了可以帮助对 X Window/Motif 系统不太了解的读者能够快速掌握基本操作方法及有关术语外,本章所介绍的关于如何使用缺省文件定义界面环境特性等内容对有效地使用和编写 Motif 应用程序是十分有用的。

Motif 应用程序中涉及到的画图功能是用 X Window 系统的基本库 Xlib 实现的,第三章将介绍如何用 Xlib 编写画图程序,读者在这里可以从一些实例程序和程序片断中学到如何用 Xlib 提供的功能在窗口内画点、画线、图形填充、写字及使用各种颜色等,以及如何用这些基本的功能组合出自己需要的图形。

第四章到第七章着重介绍如何用 Motif 编程,是本书的重点。一个 Motif 应用程序是由多个基本组件组成的,第四章就专门介绍 Motif 组件(Widget)的概念、分类和各种组件的功能、外观、特点等,这些知识是用 Motif 编程的基础。第五章以若干实例程序由浅入深地介绍 Motif 应用程序的结构、编程方法和各种 Motif 组件的使用方法,在学习了这章之后,读者即能够编写自己的 Motif 程序了。

UIL 是“用户界面语言”的缩写,是 Motif 下提供的一种专门用于编写用户图形界面模块的语言。用 UIL 和 C 语言联合编写 Motif 应用程序,比单纯用 C 语言效率更高。第六章介绍如何用 UIL 编写 Motif 程序。灵活利用第五章和第六章介绍的内容,读者就可以比较高效地编写出各种复杂的 Motif 图形界面程序了。

为了让读者更完整地掌握实际的 X Window 和 Motif 应用程序的编写方法,第七章专门向读者介绍一些 Motif 程序实例,其中除了比较专门的一些小程序外,还包括了多种常见界面风格的较复杂的程序。利用这些程序进行一定的修改,读者能很快地编写出符合自己需要的实际应用程序。

在本书最后有本书的参考文献和索引,读者可以从这里很快地查到希望了解的内容所在的页码,这在实际编程中是十分有用的。

本书的读者对象

本书的读者对象主要是各行业广大计算机工作站用户、微机 UNIX 系统用户和 UNIX 系统下的应用软件开发人员,也包括需学习 X Window 和 Motif 的高等院校学生和科研人员。读者需有初步的计算机知识,对工作站使用或微机使用有一定的了解,具备基本的 C 语言知识。同时,由于 X Window 和 Motif 的库函数都是以英文为基础写成,读者也需有初步的英语理解能力。

建议阅读顺序

我们在编写本书时尽量使所涉及的内容比较完整和具有系统性,但本书的特点并不在于强调内容的系统性和全面性,而是根据我们在工作中了解到的各行业计算机工作站用户的实际需要,从应用角度出发,试图给读者提供一本能够较快地学习用 X Window 和 Motif 编程的实用教材,同时也力求使已经入门的读者可以将本书作为实际工作中的一本有用的参考手册。因此,读者在使用本书时,没有必要像使用其它教材那样从头到尾地阅读,而可以根据自己的需要和所掌握的基础知识,采用灵活的阅读顺序。根据读者的不同情况和需要,我们建议以下几种阅读顺序:

• II •

1. 对于一般希望学习用 X Window/Motif 编程的初学者,可基本按本书的自然章节安排阅读,但第三章中的部分内容和第四章中的部分 Widgets 可以先匆匆掠过,待实际编程用到时再反过来查阅。

2. 对于较熟悉 UNIX 系统和 X Window/Motif 环境的读者,可以略去第一、二章的大部分内容直接阅读后面的章节(但 2.5 节中介绍 Motif 界面中的一些基本单元的概念和名词等,建议首先浏览一下)。

3. 对于对 X Window/Motif 编程已有一定了解的读者,可以直接阅读第五、六、七三章,需要查阅其它内容时再利用目录或索引在其它章节中查找。

4. 对于只需用 Motif 编写用户交互界面,并不需要自己在窗口内画图的读者,可以不看第三章内容;对于只需了解 X Window 编程的读者,可只重点阅读第三章,其它章节只作简要了解即可,第四章及以后各章均可略去不看。

5. 对于已经基本掌握了 X Window/Motif 编程的读者,可以直接根据需从第七章及其它章节中寻找可用作参考的程序实例,也可以将本书作为一本简要参考手册,需要时通过索引查找有关内容。

6. 本书虽是针对如何用 X Window/Motif 编程来编写的,但对于只需要使用工作站上的软件的读者也有很好的参考价值。本书第一章介绍的 UNIX 的一些基本问题和第二章介绍的 X Window 和 Motif 系统的基本使用方法等,对使用工作站上的很多软件都是必不可少的参考资料。

本书使用的若干约定

由于本书中涉及大量的实际操作方法和源程序例子,为方便读者阅读,我们特将各种英文字符和符号进行一些基本的约定,这些约定与多数计算机软件手册中的习惯是一致的。

普通字(normal characters): 系统中规定的字符,如命令或函数名、数据结构定义等,读者在使用时需按书中的字符敲入计算机。

斜体字(*italic characters*): 系统中的变量、参数等,使用中读者需根据实际情况提供。很多 Motif 和 X Window 的库函数例程都有通用的命名规则,其名字的一部分是固定的,另一部分根据具体函数发生变化,在遇到这种情况时,我们也采用斜体字表示需根据实际情况替代的部分。比如 `vi filename` 表示使用此命令时 `vi` 必须照打,而 *filename* 则应根据实际情况用要编辑的文件的文件名替代,又如 `XmCreateWidget()` 表示函数名以 `XmCreate` 开头,后面接具体的组件名。

等宽体(constant width characters): 书中提供的源程序或源程序片断等均用等宽体排印,系统显示出的信息也用等宽体排印。

中括号[]: 在命令或格式、句法说明中,包含在[]中的内容为可选内容,即用户可根据实际情况选择或不选括号中的内容。比如 `[a [b]]` 表示可以采用或不采用选项 a,而如果采用了选项 a,则还可以采用或不采用选项 b。

大括号{ }: 在命令或句法、格式说明中,包含在{ }中的内容为一个备选列,读者在使用时必须从中选择一项内容。

竖杠|: 在命令或句法、格式说明中,竖杠用来分开两项可供选择的内容。比如 `{a|b|c}` 表示必须选用选项 a、b 或 c 中的一个。

省略号 …：在命令或句法、格式说明中，省略号表示可以重复前一项内容或可以有其它可选内容。

功能键：在本书中，我们用[]或< >表示某功能键，比如用[Esc]或< Esc >表示键盘上标有 Esc 字样的功能键。

组合键 < Ctrl - x >：表示在按住 Ctrl 键后再按下 x 键，其它组合键也用类似方法表示。< Ctrl - x > 有时也用 ^x 表示。

回车 < Return >：回车表示按下回车键 < Return > 或按下输入键 < Enter >，习惯上可以用“(CR)”表示，但通常，因为在每输入一条命令后均要回车，因此，在本书中介绍的每一条命令后一般不再注明要回车。

鼠标操作：严格来说，工作站上的鼠标键一般定义为第一键、第二键和第三键等，对于用右手使用鼠标的人来说，它们分别对应于鼠标的左键、中键和右键。在本书中，为了简单起见，我们只采用左键、中键和右键的说法，左手鼠标使用者需将它们反过来。缺省情况下，鼠标左键是用来进行选择的，因此，在本书中，我们说用鼠标选中或激活(或简单说选中)界面上的某键或菜单，即是指用鼠标指向该键或菜单，然后按一下鼠标左键(称作单击)；如果说双击某按键或菜单，则是指用鼠标指向该键或菜单，然后快速地按两下鼠标左键。对于一些其它类型的界面成份，选中有时也指鼠标指向该成份使它成为当前操作的焦点。

注意：在 UNIX 系统中，字母的大小写的意义是有严格区别的，习惯使用微机的读者应特别注意，在使用本书时，应注意有关命令、操作和函数中的每一个字母应该是小写还是大写。UNIX 系统的命令和 C 语言的库函数多数都使用小写字母，而 X Window 和 Motif 中的很多函数是部分字母小写、部分字母大写的。

本书使用的名词和术语

由于目前使用的计算机编程语言都是以英文为基础的，X Window 和 Motif 的库函数也不例外，因此，X Window/Motif 编程涉及到的大量名词和术语都是英文的。虽然它们在英文中多是有实际含义的，在中文中也有对应的解释，但是，一方面有些名词术语现在还没有通用的翻译方法或者很难找到一个简洁的中文词与之对应，另一方面，这些名词术语往往是直接与有关库函数或程序变量、数据类型等相对应的，翻译成中文则失去了这一优势。有一些关于 X Window/Motif 的书籍大量采用了中文翻译的名词术语，结果反而给读者使用有关函数和查阅手册等带来了很大不便。因此，在本书中，虽然在解释有关概念时我们使用一些中文名词或用中文对一些英文术语进行解释，但在很多与编程关系比较密切的地方我们将直接使用其原文的名词术语，以方便读者直接与程序中的库函数和变量等对应。实际上，对于多数名词术语，读者只需知道其含义即可，并不必追究其确切的中文翻译。

说明

X Window 和 Motif 有几种不同的版本，在本书中，对 X Window 我们基本按照其 Version 11, Release 5(即 X11R5)版本进行介绍，对 Motif 我们主要按照其较基本的 1.1 版本介绍，部分内容参照了 1.2 版。对于大部分基本内容来说这些版本是相同或兼容的，在区别较大的地方书中将尽量专门作介绍。由于一般系统均具有向上兼容性，采用较基础的版本进行介绍有利于拥有不同版本的读者共同使用。

本书中给出的源程序和源程序片断,除了书中注明是其它来源的外,版权均为清华大学自动化系所有。本书作者在此允许读者对它们进行复制、使用和修改,但读者在这样做时必须明确注明其原作者和版权所有。对书中给出的程序和程序片断,作者已在几种计算机平台上进行了调试和检验,但它们仅供读者学习时参考,作者在此并不作出它们不包含任何错误或一定能够符合读者要求的承诺。关于有关程序的其它版权事宜参见程序前的说明语句。

本书第三章主要由刘业新编写,其余各章由张学工编写。

X Window System 是麻省理工学院(MIT)的商标和产品。

Motif 是开放软件基金会(Open Software Foundation, Inc. 或 OSF)的注册商标。

UNIX 是 UNIX 系统实验室(UNIX System Laboratories, Inc.)的注册商标。

Sun, SunOS, Solaris 等是 Sun 微系统公司(Sun Microsystems, Inc.)的商标或注册商标。

SGI, IRIX, IRIS 是 SGI 公司(Silicon Graphics Inc.)的商标或注册商标。

其它涉及到的商标和注册商标在此不再一一说明。

目 录

第一章	UNIX 和 C 语言的基础知识	1
1.1	UNIX 系统使用简介	1
1.1.1	UNIX 系统的目录结构	1
1.1.2	部分基本的 UNIX 命令	2
1.1.3	Shell 简介	4
1.1.4	常用的用户环境设置	5
1.2	vi——UNIX 下的基本编辑工具的使用	9
1.2.1	vi 的基本知识	9
1.2.2	vi 的使用举例	11
1.2.3	vi 的常用命令小结	14
1.2.4	使用 vi 时常见问题的处理	16
1.3	man——UNIX 下随时获得帮助的途径	17
1.3.1	man——UNIX 下的基本联机帮助工具	17
1.3.2	xman——X Window 下的 man	21
1.3.3	其它在线帮助工具	21
1.4	C 语言提示	22
1.4.1	编写 C 语言程序时的几个注意问题	22
1.4.2	编写 C 语言程序容易出现的错误举例	23
1.4.3	编写良好 C 语言程序的建议	24
1.4.4	有关 C 语言程序的实用工具	25
1.5	UNIX 环境下 C 语言程序的编译连接	27
1.5.1	一般的编译连接方法	27
1.5.2	用 make 进行程序的编译连接	28
1.6	C 和 FORTRAN 的混合编程	33
1.7	UNIX 下程序调试工具 dbx 的使用	34
1.7.1	dbx 简介	34
1.7.2	使用 dbx 调试程序举例	35
1.7.3	dbx 的命令	37
1.7.4	dbx 的常用命令一览表	43
第二章	X Window 和 Motif 基础	45
2.1	X Window 和 Motif 简介	45
2.2	X Window 的基本概念	46
2.3	使用 X 窗口和 Motif 窗口管理器	48
2.3.1	启动 X 窗口系统和 Motif 窗口管理器	48

2.3.2	Motif 界面的外观和基本使用方法	49
2.3.3	如何通过网络运行和显示 X 应用	53
2.3.4	中文 Motif 简介	54
2.4	按自己的爱好设置 Motif 特性	55
2.4.1	设置缺省的窗口安排	55
2.4.2	定制自己的菜单和其它 Mwm 特征	57
2.4.3	X 资源	60
2.4.4	建立自己的资源文件	61
2.5	OSF/Motif 界面风格	63
2.5.1	用户界面设计的一般原则	63
2.5.2	Motif 界面单元	64
第三章	X Window 编程入门	73
3.1	建立一个基本的 X 窗口程序	73
3.1.1	包含文件与变量说明	74
3.1.2	连接服务器	74
3.1.3	获取根窗口信息	75
3.1.4	建立窗口	76
3.1.5	位图和图标	78
3.1.6	颜色策略	78
3.1.7	与窗口管理系统的通信	78
3.1.8	选择事件类型	79
3.1.9	建立服务器资源	80
3.1.10	窗口映象与可见性	80
3.1.11	设置事件循环	80
3.1.12	出错处理	81
3.1.13	例程序的源码	81
3.2	图形上下文(GC)	90
3.2.1	创建和设置 GC	90
3.2.2	画图像素点选择的控制	93
3.2.3	其它	96
3.3	颜色的使用	97
3.3.1	基本概念	97
3.3.2	X 的颜色表(colormap)	98
3.3.3	分配共享颜色	98
3.3.4	分配私有颜色	100
3.4	事件	104
3.4.1	X 事件结构	104
3.4.2	选择事件	106
3.4.3	针对不同事件类型进行处理	106

3.4.4	各种事件类型	107
3.5	X 画图	110
3.5.1	X 画点和画线	111
3.5.2	X 画文字	113
3.5.3	X 填充图案	114
3.5.4	X 画图像	115
3.6	X 程序实例	117
3.6.1	画拖动框的程序例子	117
3.6.2	画中文字的程序例子	123
3.6.3	填充的程序例子	127
3.6.4	画图像的程序例子	132
第四章	Motif 程序的基本组件	136
4.1	Motif 编程基础	136
4.1.1	Xlib、Xt 和 Motif	136
4.1.2	Widgets 概念	137
4.1.3	事件和事件处理	139
4.1.4	Motif 的复合字符串	140
4.1.5	Motif 编程:使用 Xlib、Xt 和 Xm	141
4.2	组件类的层次关系	141
4.3	Motif 的各类组件	144
4.3.1	外壳组件(Shell Widgets)	144
4.3.2	显示组件(Display Widgets)	146
4.3.3	容器组件(Container Widgets)	147
4.3.4	对话(Dialogs)	149
4.3.5	菜单组件(Menu Widgets)	151
4.4	Gadgets 简介	151
4.5	建立组件的基本方法——便利函数	152
4.6	如何利用 Motif 的手册	153
第五章	Motif 编程方法	155
5.1	Motif 应用程序的基本结构	155
5.1.1	一个基本 Motif 程序的例子: xmbutton.c	155
5.1.2	包含头文件	157
5.1.3	程序中的初始化	159
5.1.4	设置组件的参数	160
5.1.5	创建组件	160
5.1.6	回调函数	161
5.1.7	实现组件和进入主循环	164
5.1.8	程序的编译和连接	164

5.1.9	Motif 应用程序结构小结	165
5.2	创建组件	165
5.3	Motif 程序中的组件资源	169
5.3.1	初始资源设置	169
5.3.2	应用程序的资源文件	171
5.3.3	在程序中定义资源值	172
5.3.4	一些常用的基本资源	174
5.4	一系列带有 Va 的 Xt 和 Motif 函数	175
5.5	一个复杂一些的 Motif 程序例子	178
5.5.1	例子的源程序及解释	178
5.5.2	例子的运行结果和程序的组件结构	192
5.6	Motif 应用程序设计的一般过程	194
5.7	一些常用组件的编程使用方法	195
5.7.1	可以充当界面主窗口的组件	195
5.7.2	用来显示和输入文字的组件	200
5.7.3	按键	204
5.7.4	列表和选择	206
5.7.5	弹出的小对话框	215
5.7.6	用来包容其它组件的组件	216
5.7.7	菜单	225
5.7.8	其它组件	234
5.8	复合字符串和字体	237
5.8.1	复合字符串	237
5.8.2	字体	241
5.8.3	复合字符串和字体函数一览	245
5.8.4	一个使用多种字体的例子	246
5.9	在 Motif 界面程序中用 Xlib 画图	248
5.10	编写 Motif 程序需注意的一些问题	259
第六章	用 UIL 编写 Motif 程序	261
6.1	UIL 和 MRM 简介	261
6.2	使用 UIL 编程的好处	262
6.3	一个简单的 UIL 程序例子	263
6.3.1	UIL 模块	264
6.3.2	对应的 C 程序	265
6.4	UIL 模块的编译和使用	269
6.5	UIL 模块的写法	271
6.5.1	UIL 语言的句法	271
6.5.2	UIL 文件的结构	279
6.6	C 程序和 Mm 函数	285

6.6.1	使用 UIL 模块的 C 程序结构	285
6.6.2	常用 Mm 函数简介	285
6.6.3	在 C 程序中使用 UIL 定义的组件	289
6.7	用 UIL 实现界面中的图形键钮	292
第七章	典型的 Motif 应用程序举例	297
7.1	几个专门的 Motif 小应用程序	297
7.1.1	C 程序中使用图案标签	297
7.1.2	用标尺改变颜色	303
7.1.3	一个日程序	306
7.2	一个较全面的 Motif 应用程序	310
7.2.1	应用程序 DrawLine 介绍	310
7.2.2	DrawLine 的界面结构和对应的 UIL 模块	312
7.2.3	DrawLine 的 C 程序	331
7.2.4	DrawLine 的 X 资源文件和编译连接方法	345
7.3	在 Motif 应用程序中使用汉字界面	348
7.3.1	在 UIL 模块中采用汉字标签	348
7.3.2	在 C 程序中定义汉字标签	350
7.3.3	在资源文件中定义汉字标签	354
7.4	另一个较完整的应用程序例子	355
后记		382
主要参考文献		384
索引		385

第一章 UNIX 和 C 语言的基础知识

UNIX 系统因其自身的优秀特点,现在已成为计算机工作站的标准操作系统,在微机上也有相当的应用。UNIX 系统是 X Window/Motif 系统的基础,要学习用 X Window 和 Motif 编程,必须首先学会使用 UNIX 操作系统。本章将向读者简要介绍 UNIX 系统常用的命令,包括编写 X Window/Motif 程序时最常用到的编辑器 vi 和系统联机帮助工具 man 的用法,为读者学习 UNIX 环境下的编程奠定基础。

X Window/Motif 应用程序是用 C 语言(或 C++)编写的,程序通过调用一系列的 X 和 Motif 库函数实现有关的功能。因此,要学习用 X 和 Motif 编程,会用 C 语言编程是前提。在本章中,除了介绍 UNIX 系统的知识外,我们还将简要回顾 C 语言的基本知识,并重点介绍 UNIX 环境下如何编译和连接 C 语言程序、如何使用 dbx 调试 C 语言程序以及如何联合使用 C 语言程序和 Fortune 语言程序等。这些知识,对于在 UNIX 环境下具有较少开发程序经验的读者来说是十分必要的。

1.1 UNIX 系统使用简介

1.1.1 UNIX 系统的目录结构

UNIX 系统采用层次结构的文件系统,如一个较典型的 UNIX 系统的目录结构如图 1-1 所示。其中,“/”表示一级目录(注意,是正斜线“/”而不像微机上那样是反斜线“\”)。每一个用户的注册目录的位置是由系统超级用户在建立这个用户时确定的。在 Sun 的各种工作站上,通常有一个/home 目录,各个用户的注册目录一般在这个目录下,如/home/zhang 等。如果系统安装有多个硬盘,一般习惯将它们再划出一些其它的 home 目录,如/home1,/home2 等,作为用户目录使用。在 SGI 的各种工作站上,按系统缺省情况建立的用户注册目录通常在/usr/people 下,如/usr/people/xyz 等。用户注册到机器上后所在的目录就是自己的注册目

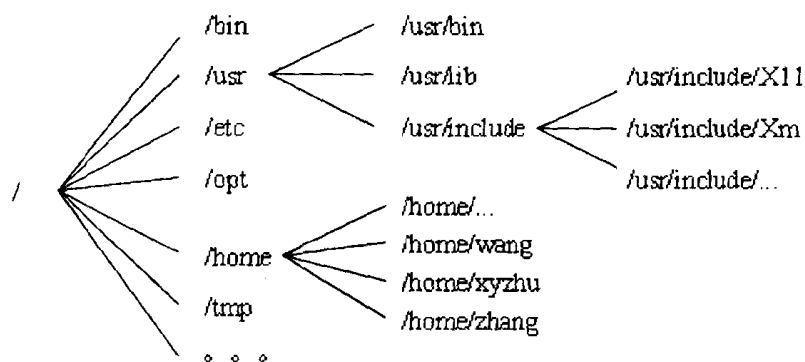


图 1-1 UNIX 的典型文件系统结构举例

录,用户可以在这里建立自己的子目录,用以存放自己的各种文件。用户的注册目录有时也称为用户的根目录,英文称作 home directory 或 home。

1.1.2 部分基本的 UNIX 命令

在本节及以后的讨论中,除非特殊说明,我们假定用户使用的是 C-Shell(见下节),其缺省提示符为“%”。

UNIX 系统命令的基本格式为

% *command* [*arguments*] [*target*](回车)

其中,*command* 为所发的命令,*arguments* 为命令的变元,*target* 为命令的执行对象,比如文件名等,命令中每一项之间需用空格隔开。在本节中,我们只简单列出一些常用的 UNIX 命令和它们的基本功能,其详细使用方法可参考有关的 UNIX 教材,或者按照本章第 1.3 节介绍的方法查阅系统联机帮助。

显示当前所在的目录:

% pwd

列文件清单:

% ls [选项] [文件名]

其中,如选项为“-l”,则以长格式显示包括文件的访问权限、文件属主(owner)、文件所属的组名、文件长度、最后修改日期和文件名等在内的各种信息;如选项为“-F”,则显示出文件的性质,如在可执行文件名后加“*”,在目录名后加“/”,在符号连接后加“@”等等;如选项为“-a”,则表示列出的文件包括隐藏文件(文件名以.开头的文件);其它选项参见系统联机帮助。

建立子目录:

% mkdir 目录名

改变当前目录:

% cd [目录名]

如不给出目录名,则回到本用户的根目录,否则到目录名指定的目录。

在屏幕上显示文件内容:

% cat 文件名

按页在屏幕上显示文件内容:

% more 文件名

用 more 显示出一屏文件内容后,左下角会显示“more(nn%)”的字样,表示已经显示了文件内容的 nn%。这时如按空格键,则显示下一页;如按回车键,则屏幕向上滚动一行;如果按“q”,则终止显示,回到系统提示符状态。

如果在执行其它命令时,系统列出的信息太多,超出了一屏的显示范围,则可通过以下方式使信息在屏幕上分页显示:

% *command* [*arguments*] | more

其中的“|”在这里为管道线,详见有关 UNIX 教材中关于管道的介绍。

复制文件:

% cp [*选项*] 原文件名 目标文件名

移动文件:

% mv [*选项*] 原文件名 目标目录名或文件名

删除文件:

% rm [*选项*] 文件名 ...

删除目录:

% rmdir 目录名 ...

(目录必须是空的才能用这种方法删除,否则应先删掉目录下的所有文件和子目录,或者改用“rm -r 目录名”来一次删除该目录及其下的所有文件和子目录。)

将若干文件存档或读存档文件:

% tar [*选项*] 存档文件名 文件名 ...

tar 是个十分有用的命令,尤其是在与软盘、磁带等外部存储设备打交道时更常要用它。其选项有很多,这里只给出较常用的几种组合的例子,其它可参见系统联机帮助。

将 *file1*、*file2* 写入存档文件 *files12.tar*:

% tar cvf *files12.tar file1 file2*

列出存档文件 *files12.tar* 中包含的文件:

% tar tvf *files12.tar*

从存档文件 *files12.tar* 中读出所包含的文件:

% tar xvf *files12.tar*

当使用外部存储设备进行存档时,tar 命令行中的存档文件名应为对应的外部存储设备的系统文件,如在 Sun 的工作站上,1.44MB 软盘的设备文件通常为/dev/rfd0,磁带机的设备文件通常为/dev/rst0 或/dev/rmt0 等,具体情况因不同机器而不同,可向系统管理员询问。

通过网络登录到一台远程计算机:

% rlogin 远程机器名 [-l 用户名]

或

% telnet 远程机器名

(在向远程机器登录时,远程机器应已经设置成允许本地机器登录的状态。远程机器名也可用远程机器的网络地址代替。详细情况可向系统管理员询问。)

通过网络传送文件:

% ftp [远程机器名或远程机器地址]

进入 ftp 后,系统首先要求远程机器的用户登录,然后系统显示 ftp 的提示符“>”,用户可以通过 ftp 命令传送文件。FTP 中一些常用的命令有:?(列出 ftp 可用的命令),ascii(设置 ASCII 传送方式),bin(设置二进制传送方式),cd(改变远程机上的当前目录),ls(列出远程机

上的文件),lcd(改变本机上的当前目录),get/mget(从远程机向本机传送一个或多个文件),put/mput(从本机向远程机传送一个或多个文件),quit(从 ftp 中退出)。

UNIX 命令中可能使用的一些常用符号的含义:

“*”: 字符串通配符,用以表示任意个字符,比如“ls a*.c”命令将列出当前目录下所有以 a 开头、以 .c 结尾的文件。

“?”: 字符串通配符,用以表示一个任意字符,比如“ls a??.c”命令将列出当前目录下所有以 a 开头、后跟两个字符、以 .c 结尾的文件。

“..”: 表示上一级目录。比如“cd ..”命令是回到上一级目录。(注意,cd 命令后必须有空格,这与微机上 DOS 的 cd 命令用法不同。)

“.”: 表示本级目录。如在 cp,mv 等命令的目标文件名中单独使用或接在目录后使用,则表示目标文件仍使用与原文件相同的文件名,如“cp abc.c ../.”表示将当前目录下的文件 abc.c 拷贝到上一级目录下也叫 abc.c 的文件。

“~/”: 表示当前用户的注册目录,如“vi ~/.cshrc”命令表示用 vi 编辑当前用户注册目录下的 .cshrc 文件。

“\”: 反斜线,在某些情况下表示转义。如果某个字符在某一命令中已有其固定的含义,而我们需要将它作为一般字符使用,这时就需用“\”对它进行转义,使其原有的定义临时不起作用,这种转义在进行字符串查找时较常用到。另外,如果我们用 alias 对某条 UNIX 命令进行了重命名(参见下节),当需要使用未经重命名的原命令时,也可在命令前加反斜线。

1.1.3 Shell 简介

Shell 的英文本意为“壳”,它是 UNIX 系统与用户之间的界面。我们注册到一个 UNIX 系统的机器上以后,面对的字符窗口或终端就是一个 Shell 进程,从 X Window 或 Motif 系统中开出一个新的命令窗口,也是在窗口中启动了一个新的 Shell 进程。Shell 负责交互地接受和执行用户输入的命令,并将执行结果显示给用户。Shell 有几种,其一是 Bourne Shell,人们也称之为标准 Shell,通常有关 UNIX 系统的书中讲的 Shell 多是指这种 Shell,它由运行/bin/sh 而得,提示符为“\$”。另一种 Shell 是 C-Shell,因其句法与 C 语言类似而得名,它由运行/bin/csh 而得,缺省的提示符为“%”,也称作 csh。两种 Shell 在基本功能上是兼容的,但又各有自己的特色,有人认为 C-Shell 用于交互使用更方便,而 B-Shell 更适合于 Shell 程序设计。除了这两种基本的 Shell 外,还有其它类型的 Shell,如 K-Shell,T-Shell 等,它们在基本功能上都是兼容的。

用户注册到系统后进入的是哪种 Shell 是由建立这个用户时的设置决定的,在系统的/etc/passwd 文件中,每个用户占有一行,如下例:

```
xyz: *:202:200:Real World Name:/home/xyz:/bin/csh
```

其中,最后一项即说明了用户 xyz 缺省的 Shell 类型,如为/bin/csh 则是 C-Shell,如为/bin/sh 则是 B-Shell,还可能为其它 Shell,如/bin/tsh 等。因为 C-Shell 是当前各种机器上较流行的 Shell,也较为广大用户所喜欢,因此,我们在本书中除非特别注明外,均假设用户使用的是 C-Shell,如果用户使用的是其它 Shell,只需对有关内容做少量修改即可。