UNIX 平台下 C 语言编程

- UNIX 基础、文件系统与 I/O
- 进程、信号、进程间通信
- Berkeley Socket
- 多线程编程、脚本编程
- 编译和调试

喻志虎 编著

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

UNIX 平台下 C 语言编程

喻志虎　编著

清 华 大 学 出 版 社

(京)新登字 158 号

内 容 简 介

本书详细而深入地介绍了在 UNIX 操作系统下利用 C 语言进行应用程序设计所需要的知识。

本书的主要内容包括：UNIX 基础知识、文件系统和文件 I/O、高级文件操作、录、终端及其他各种 I/O、进程的环境、进程控制、守护进程、进程之间的通信、信号及其机制、基于 SOCKET 的网络编程以及 Client/Server 编程、CGI 编程语言 perl 和多线程编程等。

本书内容丰富，概念清晰，在叙述上深入浅出，主要面向 UNIX 操作系统下的 C 程序设计人员，同时也适合于高等院校相关专业的师生借鉴。

书　　名：UNIX 平台下 C 语言编程
作　　者：喻志虎　编著
出 版 者：清华大学出版社(北京清华大学学研大厦，邮编 100084)
http://www.tup.tsinghua.edu.cn
责任编辑：欧振旭
印 刷 者：清华大学印刷厂
发 行 者：新华书店总店北京发行所
开　　本：787×1092 1/16　**印张**：27.75　**字数**：644 千字
版　　次：2001 年 10 月第 1 版　2001 年 10 月第 1 次印刷
书　　号：ISBN 7-302-04795-2/TP・2845
印　　数：0001～5000
定　　价：38.00 元

UNIX 是一种十分优秀的操作系统，它具有功能强大、系统稳定及开放性等特点。从计算机操作系统的发展史来看，UNIX 超过了以往所有的操作系统，取得了前所未有的成功。

UNIX 系统的发展已经有 30 多年了。许多操作系统在这期间早已销声匿迹，然而没有任何计算机工作者会预言 UNIX 在今后可见的一段时间中会走下坡路。UNIX 不仅在操作系统发展的历史上具有里程碑的作用，而且，在学术上和教学中也是人们首先研究和探讨的为数不多的操作系统之一。UNIX 极大地推动了操作系统理论的发展。

UNIX 不仅在学术上取得了成功，而且在商业上也极为成功。以前因为只有在工作站上以及更高性能的计算机上才提供 UNIX 系统，使得 UNIX 系统对普通用户来说总有一种高不可攀的感觉。随着 PC 机性能的不断提高，现在已经有了多种 UNIX 操作系统运行其上。例如 SCO 的 SCOUNIX、SUN 的 Solaris x86 以及现在流行的免费的 Linux 和 FreeBSD。在我国 PC 机普及范围比较广的情况下，这使得许多程序员可以用廉价的微机使用 UNIX 操作系统。尤其这几年，随着互联网的迅猛发展，越来越多的地方需要提供稳定快速的网络服务，因为 UNIX 具有优秀的稳定性、对网络良好的支持和众多的网络产品，凡是重要的互联网上的站点，几乎都把 UNIX 作为首选的 Web 服务器操作系统。从中可以看出 UNIX 系统的重要性。

与其他系统不同的是，UNIX 的成功与 C 语言的成功紧密相关。UNIX 自身就是 C 高级程序设计语言的编程典范。UNIX 推动了 C 语言的应用和普及，而 C 语言又反过来促进了 UNIX 在更广阔领域内的成功。

在 UNIX 环境下进行编程的程序设计人员需要学习并掌握以下两方面的内容和技术：一方面是要清楚地了解 UNIX 程序设计界面所提供的各种服务，以及正确应用它们的基本技术；另一方面是在开发较复杂的软件时，提高综合应用各种系统服务的能力，并找到疑难问题的解决方法。

本书详细而深入地介绍了在 UNIX 操作系统下进行应用程序设计所需要的各种知识。

本书的主要内容包括：UNIX 基础知识、文件系统和文件 I/O、高级文件操作、录、终端及其他各种 I/O、进程的环境、进程控制、守护进程、进程之间的通信、信号及其机制、基于 SOCKET 的网络编程以及 Client/Server 编程、CGI 编程语言 perl 和多线程编程等。

本书面向 UNIX 操作系统下的 C 程序设计人员。对读者的要求是，对 C 语言已经有了

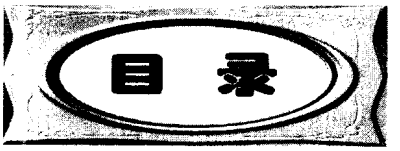
一定程度的了解。

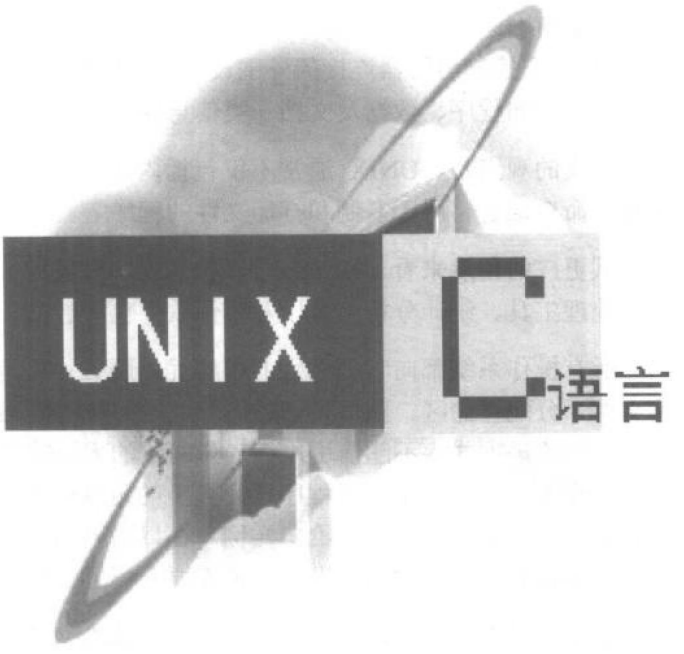
学习程序设计的最关键点是：在学习书本知识的同时，进行上机实践，才能巩固业已学到的理论知识。所以希望读者务必多上机进行实际的设计。

由于作者的水平有限，如有错误和疏漏之处，敬请读者批评、指正。

作者

2001 年 8 月





第1章

UNIX 基础

对于 UNIX 的理解通常有狭义的和广义的两种观点：狭义的观点是，它是一个分时操作系统内核，即一个控制将计算机的资源并发地分配给用户的程序。它让用户运行其程序；它控制与机器连接的外围设备，如硬盘、终端和打印机等；它提供一个文件系统用以管理诸如程序、数据及文档一类信息的长期存储。

广义的观点是，UNIX 通常不仅包括内核，而且包括一些基本程序，比如编译器、编辑器、命令语言、用于复制和打印文件的程序等。

从更广的角度来看，UNIX 可以包括用户开发的、用于运行用户定制系统的程序，如文档处理工具、统计分析工具以及图像软件包等。

所有操作系统都向它们运行的程序提供服务。典型的服务有执行新程序、打开文件、读文件、分配存储区、获得当前时间等。本书集中阐述了 UNIX 操作系统所提供的服务。以严格的步进方式、不超前引用尚未说明过的术语的方式来说明 UNIX 几乎是不可能的，甚至是令人厌烦的。本章从程序设计人员的角度快速浏览 UNIX，并对书中引用的一些术语和概念进行简要的说明并给出实例。在以后各章中，将对这些概念作更详细的说明。本章也对不熟悉 UNIX 的程序设计人员简要介绍了 UNIX 提供的各种服务。

如果用过 UNIX 且对 UNIX 比较了解，那么对本章的多数内容应该比较熟悉，可以直接跳到第 2 章。

如果打算阅读本章，需要准备一本 UNIX 程序员手册。遇到某些问题时，查手册也许会更加方便一些；另外，要弄清楚所使用的系统，以及和本书所叙述的内容的差别；最后，最好一边阅读一边进行上机实践。

1.1 登　　录

UNIX 系统可以适用于多种类型的终端，但通常习惯于小写字体的设备。要特别注意的是，UNIX 区分大小写。要确认设备的开关设置为大小写、全双工以及速率等。登录时，系统会出现：

Login:

如果出现乱码，则可能是速率不对，那么就需要检查速率和其他设置。如果仍不成功，可以试试按几次 Break 或 Interrupt 键。

出现 login：登录提示后，就可以进行登录了。登录 UNIX 系统时，先键入登录名，然后键入口令。系统在其口令文件，通常是/etc/passwd 文件中查看登录名。口令文件中的登录项由 7 个以冒号分隔的字段组成：登录名，加密口令，数字用户 ID(224)，数字组 ID(20)，注释字段，起始目录(/home/stevens)，以及 shell 程序(/bin/ksh)。很多比较新的系统已将加密口令移到另一个文件中。在以后的章节中将说明这种文件以及存取它们的函数。

登录成功后，系统先显示一些典型的系统信息，然后就可以向 shell 程序键入命令了。

1.2 shell

shell 是一个命令行解释器，是用户与 UNIX 之间的层，它读取用户输入，然后执行命令。用户通常用终端，有时则通过文件(称为 shell 脚本)向 shell 进行输入。

Shell 有三个主要的优点：

- 可以把任何程序的输出送到一文件中而不是终端上，并当作来自文件的而不是终端的输入。甚至可以将输入和输出连接到其他程序上。
- 可以通过指令文件名的模式来选取一套文件名作为程序的变量。Shell 会找出匹配该模式的文件名。
- 用户无需深入内核便可方便地控制自己的界面，可以自定义一些命令和同义词。

常用的 shell 也有三种：

- Bourne shell，/bin/sh
- C shell，/bin/csh
- KornShell，/bin/ksh

系统从口令文件中登录项的最后一个字段中了解到应该执行哪一个 shell。

自第 7 版的 shell 以来，Bourne shell 得到了广泛应用，几乎每一个现有的 UNIX 系统都提供 Bourne shell。C shell 是在伯克利开发的，所有 BSD 版本都提供这种 shell。另外，AT&T 的系统 V/386 R3.2 和 SVR4 也提供 C shell。

KornShell 由 SVR4 提供，它是在 Bourne shell 的基础上发展起来的。KornShell 在大多数 UNIX 系统上运行，但是在 SVR4 之前，它通常需要另行购买，所以目前还没有 Bourne shell 和 C shell 流行。要想对此了解得更多一些，可以参考相关资料。

本书将应用 Bourne shell 和 KornShell 都具有的功能，来执行已经开发出来的程序和一些 shell 实例。

Bourne shell 是 Steve Bourne 在贝尔实验室中开发的，其控制流结构使人想起 Algol 68。

C shell 是在伯克利由 Bill Joy 完成的，其基础是第 6 版 shell(不是 Bourne shell)。其控制结构很像 C 语言，它支持一些 Bourne shell 没有提供的功能，如作业控制、历史机制和命令行编辑。

KornShell 是 David Korn 在贝尔实验室中开发的，它兼容 Bourne shell，并且也包含了使 C shell 非常流行的一些功能，如作业控制、命令行编译等。

1.3 文件和目录

1.3.1 文件系统

文件系统控制用户文件数据的存取与检索。在 UNIX 中，设备也是作为一种特殊的文件处理的。UNIX 的文件系统管理文件空间的分配，管理文件系统的空闲空间。进程通过系统调用来对文件进行操作，也可以通过 C 语言的函数调用来操作文件，而 C 的函数库使用系统调用来实现这些函数调用。

或者更通俗地说，UNIX 文件系统就是对目录和文件的一种层次安排。目录的起点称为根(root)，用一个字符“/”来表示。

目录(directory)是一个包含目录项的文件，在逻辑上，可以认为每个目录项都包含一个文件名，同时还包含说明该文件属性的信息。文件属性是：文件类型、文件长度、文件所有者、文件的许可权(例如，其他用户能否访问该文件)、文件最后的修改时间等。stat 和 fstat 函数返回一个包含所有文件属性的信息结构。在后面的章节中还将详细说明文件的各种属性。

文件名(filename)是出现在目录中的各个目录项的名字。某些 UNIX 文件系统限制文件名的最大长度为 14 个字符，而 BSD 版本则将这种限制扩展为 255 个字符。

需要注意的是，有一些字符是不能出现在文件名中的，例如斜线符“/”和空操作符(null)。斜线分隔构成路径名(在下面说明)的各文件名，空操作符则终止一个路径名。同样，不可显示的控制字符也是不能出现在文件名中的。其他禁止使用的字符还有* ? “ ‘ ` [ ]和其他控制字符。

如果在文件名中使用了某些 shell 特殊字符，则必须使用 shell 的引号机制来引用文件名。所以，建议只使用印刷字符的一个子集作为文件名字符。

此外，还需要注意文件和目录名的大小写。

当创建一个新目录时，自动创建了两个文件名：.(称为“点”)和 ..(称为“点-点”)。“点”引用当前目录，“点-点”则引用父目录。在最高层次的根目录中，“点-点”与“点”相同。

以斜线分隔的文件名序列(可以任选地以斜线开头)构成路径名(pathname)，以斜线开头的路径名称为绝对路径名(absolute pathname)，如/space/oracle；否则称为相对路径名(relative pathname)，如 oracle。

下面的例子是列出一个目录中所有文件的名字，即 ls(1)命令的源码。

```
#include <sys/types.h>
#include <dirent.h>
```