

©2009 Pearson Education, Inc. All rights reserved.

SCD Developer 开发指南

编程工具指南

图 1-1 展示了 SCD Developer 的
主界面。在左侧的“工程树”中，
可以看到当前工程的结构。在右侧
的“设计器”中，可以看到当前
工程的设计视图。

图 1-2 展示了 SCD Developer
的“工程树”视图。在左侧的
“工程树”中，可以看到当前
工程的结构。在右侧的“设计
器”中，可以看到当前工程的设计
视图。

SCO OpenServer 5 实用大全系列丛书之五

SCO OpenServer 开发系统

编程工具指南

熊胜峰 施运梅 编译
李国华 吴 健
郭逢荣 审校

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书阐述了如何用 SCO OpenServer 开发系统创建和维护 C 程序,它提供了编程工具及如何用它们开发软件的方法。本书包括:随开发系统提供的与 ANSI 一致的 C 编译程序的实现细节,调试工具以及如何用它们来发现和排除 C 语言与汇编语言中的错误方面的信息,以及如何用 dbXtra 在基于 OSF/Motif 的窗口环境下调试 C 语言与 C++ 语言程序的信息等。

本书的读者对象为使用 SCO OpenServer 开发系统及其它 UNIX 系统开发应用软件的开发人员,以及计算机专业的大学高年级学生或研究生。

版权所有,翻印必究。本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

SCO OpenServer 开发系统编程工具指南/熊胜峰等编译. 北京:清华大学出版社,1998. 10
ISBN 7-302-03124-X

I. S... II. 熊... III. 软件工具-指南 IV. TP311.56

中国版本图书馆 CIP 数据核字(98)第 31546 号

出 版 者: 清华大学出版社(北京 清华大学校内, 邮政编码: 100084)

因特网址: <http://www.tup.tsinghua.edu.cn>

责任编辑: 柳秀丽 刘小峰

印 刷 者: 北京市清华园胶印厂

发 行 者: 新华书店总店北京科技发行所

开 本: 787×1092 1/16 印张: 34.5 字数: 818 千字

版 次: 1999 年 1 月第 1 版 1999 年 3 月第 2 次印刷

书 号: ISBN 7-302-03124-X/TP·1664

印 数: 2001—4000

定 价: 168.00 元

前 言

UNIX 系统自 1969 年踏入计算机世界以来已近 30 年。虽然目前市场上面临某种操作系统(如 Windows NT)强有力的竞争,但是它仍然是笔记本电脑、PC、PC 服务器、中小型机、工作站、大巨型机及群集、SMP、MPP 上全系列通用的操作系统,至少到目前为止还没有哪一种操作系统可以担此重任。而且以其为基础形成的开放系统标准(如 POSIX)也是迄今为止唯一的操作系统标准,即使是其竞争对手或者目前还尚存的专用硬件系统(某些公司的大中型机或专用硬件)上运行的操作系统,其界面也是遵循 POSIX 或其它类 UNIX 标准的。从此意义上讲,UNIX 就不只是一种操作系统的专用名称,而成了当前开放系统的代名词。君不见许多公司“开放系统”的旗帜不是 UNIX 在世界上流行并形成了开放的潮流之后,自愿或不自愿地举起来的吗?就此一点 UNIX 就功不可没。我们在 80 年代初就说过下列的话:UNIX 这一名字可能会在电脑世界消失,但其精神永存,今日看起来还是这样。

在技术不断进步、产品日益丰富、市场激烈竞争的今天,我们每一个人都希望快速而有效地掌握最新技术并应用到实际工作中去。而科学技术知识浩如烟海,总是根据需要有所选择,学习 UNIX 系统也应如此。

除了一些特定的群集、SMP 和 MPP 上的大型系统之外,其余硬件平台上使用的 UNIX 系统目前主要分两大部分。一部分是各硬件厂家根据 UNIX 基本系统和相关标准开发并配备在不同平台上的系统,其代表有 IBM 的 AIX、HP 的 UX、NCR 的 RAS、SUN 的 Solaris 等;另一部分是专业软件厂家提供的系统,其代表有 SCO UNIX。SCO UNIX 是目前在 Intel 芯片构成的硬件平台上运行得最为普遍的 UNIX。AT&T 在 1989 年推出了 UNIX 系统 V 第 4 版(SVR4),之后 AT&T 把 UNIX 系统及其商标卖给了 Novell,Novell 以此为基础开发了 UnixWare。之后,Novell 又把 UnixWare 卖给了 SCO,SCO 以此为基础结合其以前的 XENIX 和 UNIX 版本于 1997 年开发了新的 UNIX 系统 V 第 5 版(SVR5)。

综合了以前的 OpenServer 和 UnixWare,SCO 推出了 OpenServer 5.0 版。所以 OpenServer 5.0 是目前为止在市场上最为流行的也是最新的商品系统。由于以 Intel 芯片为 CPU 的 PC、PC 服务器、笔记本电脑等在市场上极为普及,因此学习以这些电脑为载体的 SCO OpenServer 5.0 就成了掌握 UNIX 系统知识最便捷的途径。

本丛书的编译者系中国科学院软件研究所的有关专家学者,1979 年曾率先在国内引进了 UNIX 系统,近 20 年来基于 UNIX 作了大量的研究开发工作。

本丛书在 SCO 公司提供的技术资料的基础上,根据编译者的实践经验,增加了部分新的内容。

全套丛书共有六本:

丛书之一《SCO OpenServer 系统手册》重点在系统安装及配置、系统的基本操作(如启、停系统)以及硬件设备。

丛书之二《SCO OpenServer 用户指南》向用户全面介绍系统的使用,特别是 shell 命令及其编程,系统命令使用及一般工作的使用。

丛书之三《SCO OpenServer 系统管理指南》则侧重于系统管理,这只有在学习了丛书之一和之二后,对系统有一个完整的了解并懂得使用之后,才能有条件学习该书内容,也只有在掌握了这部分知识之后才有能力管理好系统。

丛书之四《SCO OpenServer 网络指南》侧重于 SCO 网络工具的讲述,目的是帮助读者访问各种网络服务。利用这些工具 SCO 用户可将 OpenServer 系统与各种相似或不同的硬件平台,如专用微机、工作站、运行 DOS 或 OS/2 的 PC 机以及软件平台(如其它 UNIX 系统)连接。

丛书之五《SCO OpenServer 开发系统编程工具指南》则不像前面几本书只要求读者学会使用和管理系统,而是对读者提出了更高的要求:利用开发工具和系统来编程、调试和组装构成系统。利用 UNIX 系统的各种编程工具进行系统开发是 UNIX 系统开发人员必须掌握的知识,而且掌握越多就越“自由”,而本书介绍的这些工具是 UNIX 系统最重要、最有使用价值的,读者应不断深入掌握这些知识,使自己成为真正的 UNIX 行家里手。

丛书之六《SCO OpenServer 程序员技术精粹》主要是在丛书之五基础上对程序员提出的更高要求。书中对 SCO OpenServer 新增加的一些特征进行了说明,并辅以实例。对于程序员来说掌握这些知识对于其在 UNIX 系统上的开发会有更多的好处。

这套丛书涉及的内容很多,读者可以根据需要选读或通读全套丛书中的几本或一本中的几章、几节。希望这套丛书的出版对读者掌握 UNIX,特别是 SCO UNIX 有所帮助。

由于编译这套丛书资料量大,参与编译的工作人员又同时承担其它多项任务,为了尽快将此套丛书奉献给读者,急切之中不妥之处在所难免,敬请广大读者不吝赐教。

编译者

1998 年 4 月 29 日

序

UNIX 是当前世界上使用普遍、影响深远的主流操作系统。在 UNIX 世界中,SCO UNIX 一直以其植根于 Intel 平台,接近于大众的特色而名满天下。这其中最新的产品就是 SCO 的 OpenServer 系列,也就是摆在您面前的这一套丛书想要向您介绍的内容。根据国际数据集团(IDG)的统计,在 1996 年,SCO UNIX 在全球 UNIX 市场占有 35% 的市场。1997 年,SCO UNIX 的市场占有率提高到 40%。这其中大多数的产品是 SCO OpenServer。

UNIX 从诞生到现在已经有了近 30 年的历史。在这 30 年的风风雨雨中,UNIX 有过它的徘徊,也有过它的辉煌。在 20 世纪行将结束的时候,UNIX 再次面临市场的考验。来自于其它操作系统产品的竞争使人们产生了许多疑问和误解,对此我们用个时髦词称之为“误区”。在您开始阅读这套丛书之前,我认为有必要把这些误区澄清一下,以免在您学习的过程中还要考虑为什么要学习 UNIX。

误区之一: UNIX 只适用于大中型核心系统

虽然在群集、SMP、MPP 领域 UNIX 是基本的甚至是唯一的操作系统,但是 UNIX 同时也是唯一可以运行在笔记本电脑、PC、PC 服务器、小型机乃至大型机上的操作系统。从以上 IDG 的统计数字可以看出,有四成的 UNIX 系统是 SCO UNIX,而 SCO UNIX 是只能运行在 Intel 平台上的。加上其它种类的 UNIX,可以说,世界上一大半 UNIX 系统是运行在 PC 上的。而今天的 SCO UNIX 也同样具备了支持群集和 SMP 的能力。

UNIX 可支持多用户核心系统,同时也支持分布式网络系统。对网络的支持是 UNIX 能够长期发展的关键所在。在那句“网络就是计算机”的名言背后,UNIX 功不可没。因为有了 UNIX,才有了 TCP/IP,才有了 Internet/Intranet。懂了 UNIX,您就能更深刻地理解它们的精髓。

误区之二: UNIX 与我们的日常生活无关

UNIX 虽然不像 Windows 在每个办公室都可以见到。然而您的生活是离不开 UNIX 的。就我国而言,目前大部分关键性的业务是运行在 UNIX 平台上的。这其中包括银行、保险、税务、邮政和电信、铁道、海关、气象及政府机构等。所有这些都在您的身边。相信许多了解 SCO UNIX 的人对此都深有体会。大部分关键应用部门的高层技术人员和管理人员都懂 UNIX 甚至是 UNIX 专家。如果您希望自己成为一位合格的信息主管(CIO),UNIX 恐怕是您的必修课程。

误区之三: UNIX 是阳春白雪,很难学且学而无用

UNIX 的技术含量很高,专家们说,当今计算机软件技术中的大部分技术都可以从

UNIX中找到它们的来源和影子。正因为如此,UNIX 才具有如此的高性能、可靠性、稳定性、安全性、互操作性、可用性、可伸缩性等。也正因为如此,UNIX 要学习的东西比其它的多。同时,也要承认由于历史的原因,UNIX 的用户界面不够友好。但是这种状况现在已经大大改观。从这套丛书中您就可以看到,SCO 的 OpenServer 5 系统的界面已经变得非常友好。学了 UNIX,您会感觉到走入了一个新的境界,这种境界不是那些只满足于对软件技术浅尝辄止、一知半解的人所能够达到的。

还没有遇到过学了 UNIX 而无用武之处的。遇到的都是对 UNIX 感叹“书到用时方恨少”的。甚至没有见到买了 UNIX 系统而搁置不用的,虽然这种情况对于其它产品并不少见。如果您留心招聘软件工程师的广告,您能发现多少是不需要懂 UNIX 的?

误区之四: UNIX 没有发展

UNIX 自从诞生那天起就没有停止过发展。我们只看最近几年就能明了。

- 1995 年,新标准 UNIX'95 问世
- 1996 年,两大 UNIX 集团 X/Open 和 OSF 合并成为“开放集团”
- 1997 年,新的 UNIX 核心 SVR5(UNIX 系统 5,版本 5)在 SCO 完成
- 1998 年,基于 SCR5 的综合了 OpenServer 和 UnixWare 的 UnixWare7 版本由 SCO 推出,并计划推出 OpenServer 5.0.5 版本

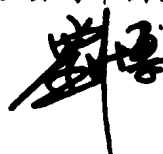
这些新的标准的产品发展和更多新的技术融合,使得它们能为客户提供更强有力的功能和性能,对客户更友好,与 Internet/Intranet 结合的更紧密。这也是为什么 UNIX 仍然被 Intel 所钟爱,被广大软、硬件厂商包括 IBM,HP,Compaq,SUN,NCR 等作为标准操作系统平台的原因。

误区之五: UNIX 将被其它操作系统所取代

如果我们谈一个产品或者一种技术是否将被其它的产品或技术取代,或者直截了当地说它是否会消亡,那么我想必须有两个前提:其一是这种技术或产品已经不能适应市场的需要,属于被市场淘汰;其二是有一种比它更好、更先进的技术或产品问世。但是认真研究 UNIX,我们没有发现上述两个前提的存在。不但如此,UNIX 的技术到目前为止仍然是没有任何一种操作系统可以与之比拟的,而且基于 UNIX 的大量的应用是一笔不可估量的财富。UNIX 将会在相当长的时间里与其它操作系统共存,但是两年前谈 UNIX 会消亡的“预言家”面对 UNIX 目前的发展已经噤若寒蝉,现在预言 UNIX 在不久将被取代的惊世骇俗之言也同样会在历史面前成为茶余饭后的笑料。

现在您是不是对学好这套丛书很有信心了? 希望如此。

美国 SCO 公司中国区总经理



一九九八年春

目 录

关于本书	(1)
如何使用这本指南	(1)
符号约定	(2)
相关文档	(3)
其它参考资料	(5)
第 1 章 使用编程工具	(6)
1.1 创建源代码	(6)
1.2 将代码存档	(8)
1.3 分析代码	(9)
1.3.1 使用 cb	(10)
1.3.2 使用 cscope	(10)
1.3.3 使用 lint	(12)
1.3.4 使用 cflow	(12)
1.3.5 使用 cxref	(13)
1.4 将分析过程中所做的修改归档	(15)
1.5 编译代码	(15)
1.5.1 使用 make	(16)
1.6 测试代码	(17)
1.7 调试代码	(18)
1.7.1 使用 dbxtra	(18)
1.7.2 测试修正的结果	(22)
1.8 程序结构和指导方针	(22)
1.8.1 程序结构	(23)
1.8.2 前导文件	(26)
1.8.3 出错处理	(26)
第 2 章 C 编译系统	(27)
2.1 编译和链接	(27)
2.1.1 基本 cc 命令行语法	(29)
2.1.2 cc 命令行选项的一般使用	(32)
2.1.3 链接	(33)

2.2 库和前导文件.....	(48)
2.2.1 前导文件.....	(48)
2.2.2 如何使用库函数.....	(49)
2.2.3 C 库(libc)	(51)
2.2.4 数学库(libm)	(55)
2.2.5 常规库(libgen)	(56)
2.2.6 标准 I/O	(58)
第 3 章 C 语言编译程序	(63)
编译模式	(63)
翻译阶段	(64)
3.1 源文件和单词化.....	(65)
3.1.1 记号.....	(65)
3.1.2 标识符.....	(65)
3.1.3 关键字.....	(65)
3.1.4 常量.....	(66)
3.1.5 字符串正文.....	(68)
3.1.6 宽字符串正文.....	(68)
3.1.7 标点.....	(68)
3.1.8 注释.....	(69)
3.2 预处理.....	(69)
3.2.1 三字符序列.....	(69)
3.2.2 预处理单词.....	(69)
3.2.3 预处理指令.....	(70)
3.3 声明和定义.....	(75)
3.3.1 基本类型.....	(75)
3.3.2 作用域.....	(77)
3.3.3 存储持续时间.....	(78)
3.3.4 存储类型说明.....	(78)
3.3.5 声明.....	(79)
3.3.6 函数定义.....	(81)
3.4 转换和表达式.....	(82)
3.4.1 隐含转换.....	(82)
3.4.2 表达式.....	(83)
3.4.3 操作符.....	(84)
3.4.4 操作符的结合性和优先级.....	(89)
3.4.5 常量表达式.....	(89)
3.4.6 初始化.....	(90)

3.5 语句.....	(92)
3.5.1 表达式语句.....	(92)
3.5.2 复合语句.....	(92)
3.5.3 选择语句.....	(92)
3.5.4 循环语句.....	(93)
3.5.5 跳转语句.....	(94)
3.6 可移植性考虑.....	(95)
第4章 COFF 链接编辑程序	(97)
4.1 段.....	(97)
4.2 存储器配置.....	(98)
4.2.1 地址.....	(98)
4.2.2 连接.....	(98)
4.2.3 目标文件.....	(98)
4.3 链接编辑程序的命令语言.....	(99)
4.3.1 表达式.....	(99)
4.3.2 赋值语句	(100)
4.3.3 指向存储器配置	(101)
4.3.4 段定义指令	(103)
4.4 改变入口点	(111)
4.5 使用归档库	(112)
4.6 分配算法	(113)
4.6.1 增量链接编辑	(114)
4.6.2 DSECT,COPY,NOLOAD,INFO 和 OVERLAY 段	(115)
4.6.3 输出文件分块	(116)
4.6.4 不可重定位输入文件	(116)
4.7 输入指令语法图	(117)
第5章 lint 分析程序	(121)
5.1 lint 做些什么	(122)
5.1.1 一致性检查	(122)
5.1.2 可移植性检查	(123)
5.1.3 可疑的结构	(124)
5.2 使用 lint	(125)
5.2.1 lint 库	(126)
5.2.2 lint 过滤程序	(127)
5.2.3 选项和指令	(127)
5.3 理解“lint-特有”的消息	(131)

第 6 章 dbXtra 和 dbxtra (156)

6.1 从命令行调用 dbXtra 和 dbxtra (156)

6.2 dbXtra 界面 (157)

6.2.1 配置 dbXtra X11 和 Motif 资源 (159)

6.3 dbxtra 界面 (159)

6.3.1 屏幕方式 (160)

6.3.2 dbxtra 键盘接口 (161)

6.4 dbxtra 辅导 (166)

6.4.1 为 dbxtra 准备文件 (166)

6.4.2 启动 dbxtra (167)

6.4.3 dbxtra 屏幕 (167)

6.4.4 dbxtra 求助机制 (168)

6.4.5 打印变量值 (168)

6.4.6 查看源代码和别名 (168)

6.4.7 查看调用栈 (169)

6.4.8 遍历调用栈 (169)

6.4.9 监视变量内容 (169)

6.4.10 分析结果 (170)

6.4.11 检查不同函数 (170)

6.4.12 使用断点 (171)

6.4.13 从断点后继续执行 (171)

6.4.14 退出 dbxtra (172)

6.5 dbXtra 和 dbxtra 命令 (172)

6.5.1 File 选单:保存和退出 (172)

6.5.2 Events 选单:管理执行时事件 (172)

6.5.3 eXecute 选单:执行被调试程序 (178)

6.5.4 Source 选单:存取源文件 (180)

6.5.5 Info 选单:调试寄存器、存储单元和表达式 (181)

6.5.6 Options 选单:控制调试环境 (183)

6.5.7 help 选单 (185)

6.5.8 curses 界面命令 (185)

6.5.9 默认命令别名 (187)

6.5.10 dbXtra 和 dbxtra 表达式和条件 (187)

6.6 C++ 专用调试特性 (189)

6.6.1 使用需求 (189)

6.6.2 重载的函数和操作符 (189)

6.6.3 dbXtra 和 dbxtra cc 命令 (190)

6.6.4	调试子进程	(191)
6.7	连接到运行进程	(192)
第 7 章	sdb:符号调试程序	(193)
7.1	使用 sdb	(193)
7.1.1	以一个程序文件启动 sdb	(193)
7.1.2	以一个内存映像启动 sdb	(194)
7.1.3	显示一个栈的轨迹	(195)
7.1.4	诊断变量	(195)
7.1.5	规定变量格式	(196)
7.2	显示和操作源文件	(197)
7.2.1	显示源文件	(197)
7.3	控制程序的执行	(198)
7.3.1	设置和删除断点	(198)
7.3.2	程序的单步跟踪	(199)
7.3.3	运行程序	(200)
7.3.4	调用函数和过程	(200)
7.4	调试机器语言程序	(201)
7.4.1	显示机器语言语句	(201)
7.4.2	寄存器操作	(201)
7.5	其它命令	(201)
7.6	样本 sdb 对话	(202)
第 8 章	adb 调试程序	(205)
8.1	启动 adb	(205)
8.1.1	使用程序文件启动	(205)
8.1.2	使用内存映像文件启动 adb	(206)
8.1.3	退出 adb	(206)
8.2	显示指令和数据	(207)
8.2.1	形成地址	(207)
8.2.2	形成表达式	(207)
8.2.3	选择数据格式	(211)
8.2.4	使用“=”命令	(213)
8.2.5	使用“?”和“/”命令	(213)
8.2.6	一个例子:简单格式化	(214)
8.3	调试程序执行	(215)
8.3.1	执行一个程序	(215)
8.3.2	设置断点	(216)

8.3.3	显示断点	(216)
8.3.4	继续执行	(217)
8.3.5	使用中断键和退出键停止程序	(217)
8.3.6	单步执行程序	(217)
8.3.7	杀死程序	(218)
8.3.8	删除断点	(218)
8.3.9	显示 C 栈回溯	(218)
8.3.10	显示 CPU 寄存器	(218)
8.3.11	显示外部变量	(219)
8.3.12	一个例子:跟踪多个函数	(219)
8.4	使用 adb 内存映像	(223)
8.4.1	显示内存映像	(223)
8.4.2	改变内存映像	(224)
8.4.3	建立新的映像项	(225)
8.4.4	验证地址的合法性	(225)
8.5	其它特点	(226)
8.5.1	在一行中组合命令	(226)
8.5.2	建立 adb 命令文件	(226)
8.5.3	设置输出宽度	(227)
8.5.4	设置最大位移量	(227)
8.5.5	设置默认输入格式	(228)
8.5.6	使用 UNIX 命令	(228)
8.5.7	计算数值和显示正文	(228)
8.5.8	一个例子:卸出目录和 i 节点	(229)
8.6	修补二进制文件和内存	(230)
8.6.1	在文件中确定值的位置	(230)
8.6.2	写文件	(231)
8.6.3	修改内存	(231)
第 9 章	C 程序员的效率工具	(232)
9.1	prof	(232)
9.1.1	建立程序的剖视版本	(232)
9.1.2	运行被剖视的程序	(233)
9.1.3	PROFOPTS 环境变量	(233)
9.1.4	使用 PROFOPTS 的例子	(233)
9.2	解释剖视输出	(235)
9.2.1	查看剖视源列表	(236)
9.2.2	为 lprof 指定程序和数据文件	(238)
• X •		

9.2.3	lprof 所需文件	(238)
9.2.4	文件子集的源列表	(239)
9.2.5	总计选项	(239)
9.2.6	归并选项	(240)
9.2.7	使用 lprof 的注意事项	(240)
9.2.8	使用 prof 和 lprof 提高性能	(242)
9.2.9	使用 lprof 扩大测试覆盖范围	(242)
9.3	cscope	(244)
9.3.1	配置环境	(244)
9.4	使用 cscope	(245)
9.4.1	运行 cscope	(247)
9.4.2	交叉引用文件	(247)
9.4.3	一个辅导例子:查找出错信息源	(248)
9.4.4	条件编辑指令	(253)
9.5	使用 cscope 的例子	(253)
9.5.1	改变文本串	(253)
9.5.2	函数增加实参	(255)
9.5.3	改变变量的值	(256)
第 10 章	make	(257)
10.1	基本功能	(257)
10.2	并行 make	(259)
10.3	递归 makefile	(259)
10.4	makefile 和替换	(260)
10.4.1	依赖行语法	(260)
10.4.2	依赖关系信息	(260)
10.4.3	宏定义	(261)
10.4.4	可执行命令	(262)
10.4.5	输出转换	(263)
10.5	后缀和转换规则	(264)
10.5.1	隐含规则	(264)
10.5.2	归档库	(266)
10.5.3	SCCS 文件名中的波浪线	(267)
10.5.4	空后缀	(269)
10.5.5	创建新的后缀规则	(269)
10.6	包含文件	(270)
10.7	动态依赖参数	(270)
10.8	环境变量	(271)

10.9	建议和警告	(272)
10.10	Makefile 例子	(273)
10.11	内部规则	(274)
第 11 章	源代码控制系统	(279)
11.1	概述	(279)
11.2	基础知识	(280)
11.2.1	文件和目录	(280)
11.2.2	delta 和 SID	(281)
11.2.3	SCCS 工作文件	(282)
11.2.4	文件管理员	(283)
11.3	建立和使用 s-文件	(283)
11.3.1	建立 s-文件	(283)
11.3.2	为读而复原文件	(284)
11.3.3	编辑和修改 SCCS 文件	(285)
11.3.4	使用多个版本	(286)
11.3.5	比较版本之间的差异	(290)
11.3.6	简化版本结构	(291)
11.3.7	显示 s-文件	(292)
11.3.8	显示 delta 版本的信息	(295)
11.3.9	包含与排除 delta	(296)
11.4	使用标识关键字	(297)
11.5	使用 s-文件标志	(300)
11.6	修改 s-文件信息	(302)
11.6.1	加入注释	(302)
11.6.2	修改注释	(303)
11.7	保护和修复 s-文件	(303)
11.7.1	检查 s-文件	(305)
11.7.2	修复 SCCS 文件	(305)
11.8	同时编辑一个 s-文件	(306)
11.9	SCCS 的求助功能	(307)
11.10	make 与 SCCS	(307)
第 12 章	词法分析程序的生成程序 lex	(310)
12.1	lex 的工作原理	(310)
12.2	lex 源程序中的正则表达式	(311)
12.2.1	字符的集合	(311)
12.2.2	闭包运算	(312)

12.2.3 选择和字符组	(312)
12.2.4 重复和辅助定义	(312)
12.2.5 其它	(312)
12.3 lex 源程序中的动作	(312)
12.4 lex 源程序的格式	(314)
12.5 lex 程序的输入和输出	(316)
12.6 二义性和上下文相关性的处理	(319)
12.6.1 二义性的处理	(319)
12.6.2 上下文相关性的处理	(319)
12.7 其它	(321)
12.8 常用的一些运算符	(322)
12.9 lex 与 yacc 的配合使用	(323)
第 13 章 编译程序的编译程序 yacc	(324)
13.1 yacc 的工作原理	(324)
13.2 一个简单的例子	(325)
13.3 yacc 源程序中的规则	(328)
13.4 yacc 源程序中的动作	(329)
13.4.1 语义值的定义	(330)
13.4.2 复杂的语义动作	(330)
13.5 yacc 源程序的格式	(331)
13.5.1 说明部分的写法	(331)
13.5.2 运算符优先级和结合性的定义	(333)
13.5.3 程序段部分的写法	(334)
13.6 yacc 和 lex 的配合使用	(335)
13.7 yacc 源程序的风格	(336)
13.8 对输入的进一步讨论	(336)
13.9 再论二义性的处理	(337)
13.10 语法分析中错误的处理	(338)
13.11 其它	(339)
13.12 例子	(339)
13.12.1 计算器 hoc	(339)
13.12.2 计算器 ec	(341)
13.12.3 汉语拼音流分析程序	(360)
第 14 章 共享库	(372)
14.1 什么是共享库	(372)
14.1.1 共享库例子	(373)

14.1.2	建立 a.out 文件	(373)
14.2	决定是否使用共享库	(373)
14.3	空间考虑	(374)
14.3.1	节省空间	(374)
14.3.2	提高内存空间的利用率	(376)
14.3.3	编写应用程序	(376)
14.3.4	标识使用共享库的 a.out 文件	(376)
14.3.5	调试使用共享库的 a.out 文件	(377)
14.4	实现共享库	(377)
14.4.1	主库和目标库	(377)
14.4.2	转移表	(378)
14.5	建立共享库	(379)
14.5.1	建立过程	(379)
14.5.2	编写共享代码的原则	(382)
14.6	选择库成员	(382)
14.6.1	包含大的、使用频繁的例程	(382)
14.6.2	排除不常使用的例程	(383)
14.6.3	排除使用很多静态数据的程序	(383)
14.6.4	排除维护复杂的例程	(383)
14.6.5	包含库本身所需例程	(383)
14.7	改变共享库的已有代码	(383)
14.7.1	将全局数据减至最少	(384)
14.7.2	使用说明文件实现兼容性	(386)
14.7.3	传入符号	(387)
14.7.4	提供和非共享库的兼容性	(392)
14.7.5	调整共享库代码	(393)
14.7.6	检查兼容性	(394)
14.7.7	示例	(397)
附录 A	ANSI 定义实现的行为	(404)
A.1	翻译	(404)
A.1.1	识别诊断结果	(405)
A.2	环境	(405)
A.2.1	main()的实参	(405)
A.2.2	交互设备	(405)
A.3	标识符	(405)
A.3.1	无外部链接的有效字符	(406)
A.3.3	有外部链接的有效字符	(406)