



XENIX 软件开发指南

王 晓 龙 译

孙 玉 方 校

北京科海总公司培训中心

中国科学院软件研究所

一九八七年四月

IBM PC/AT硬件和XENIX系统资料汇编

(之五)

XENIX 软件开发指南

王 晓 龙 译

孙 玉 方 校

北京科海总公司培训中心

中国科学院软件研究所

一九八七年四月

编辑: 科海培训中心教材部
发行: 科海培训中心资料组
地址: 北京2725信箱 科海培训中心
资料组

(北京海淀路332路黄庄站旁)

印刷: 河北省蔚县印刷厂

编 者 序

IBM PC已从PC、PC/XT推进到PC/AT。PC/AT以Intel80286为主CPU，具有丰富的硬件资源。鉴于目前DOS系统基本上是一种单用户系统，许多硬件资源未得到充分利用，许多用户都要求在PC/AT上配备多用户多任务的XENIX系统。

XENIX系统是UNIX系统在以Intel为主CPU的微机上的实现，该系统由Microsoft公司开发。目前在PC/AT上运行的XENIX相当于UNIX的System III或System V。

为了更好地在国内推广PC/AT及其兼容机，中国科学院软件研究所在其雄厚的技术力量基础上，积多年研究、开发UNIX系统之经验，开发成功多种XENIX中西文信息处理系统并移植到几乎所有PC/AT的兼容机和部分386机上。为了更好地推广XENIX中英文信息处理系统，科海培训中心和中国科学院软件研究所组织了一批专家和技术人员，收集并编译整理了有关XENIX及IBM PC/AT的全部技术资料。

这些资料包括以下几类：

- I. IBM PC/AT硬件资料，包括硬件安装及组装手册、技术手册和维护手册。
- II. XENIX基本系统的安装、基本用户指南、命令参考手册、系统管理手册和直观shell手册。
- III. XENIX开发系统方面的软件开发手册、库函数程序员手册、系统调用和子程序手册。
- IV. XENIX系统上运行的汇编语言和各种高级语言（C、Fortran、Cobol、Basic）的用户指南和参考手册。
- V. XENIX正文格式化处理手册。
- VI. XENIX系统上配备的最新版本INFORMIX和UNIFY数据库管理系统用户及参考手册。
- VII. 中西文兼容的C—XENIX系统安装和基本使用手册。

全套资料约400万字，分装成20本。

全书的主要译校任务由中国科学院软件研究所的专家、技术人员承担，科海培训中心负责编辑、印刷和发行工作。

由于时间仓促，本资料汇编中必有不少错漏之处，敬请读者批评指正，以便再版时更正。

主编 孙玉方
董洪皋

本书概述

本指南解释如何用XENIX软件开发系统来建立、维护C语言和汇编语言程序。本指南面向熟悉C语言和XENIX系统的程序员。

C语言程序员可以阅读第二、三、四、七章，它们解释如何编译、调试C语言程序。希望将程序的编译过程自动化的程序员可以阅读第五章，它解释make程序。希望对一个程序的多个版本进行组织、维护的程序员可以阅读第六章，它解释SCCS命令。

需要有一种便利的方法生成词法分析程序和语法分析程序的特殊项目的程序员可以阅读第八、九章，它们解释lex和yacc程序生成程序。

本书结构如下：

第一章 引论

介绍XENIX软件开发系统。

第二章 CC：C编译程序

介绍如何使用CC命令来编译C语言程序。

第三章 XENIX到DOS：交叉开发系统

为如何建立在DOS下运行的程序提供信息。

DOS交叉开发系统允许在XENIX系统上建立、编译、连接DOS程序，并将这些程序传送到DOS系统。

第四章 lint程序：C程序的检查程序

解释如何用C语言程序的检查程序lint来对C语言程序的语法和语义的正确性进行检查。

第五章 程序的维护程序：make

解释如何使用make程序使程序和其它项目的开发过程自动化。

第六章 SCCS，源代码控制系统

解释如何用SCCS命令来控制和维护一个项目的源文件的所有版本。

第七章 adb程序调试器

解释如何用调试程序adb对C语言程序和汇编语言程序进行调试。

第八章 lex程序：词法分析器

解释如何用程序生成程序lex来生成词法分析程序。

第九章 yacc程序产生器：编译程序的编译程序

解释如何用程序生成程序yacc来生成语法分析程序。

第十章 C Shell

解释如何使用命令解释程序C Shell。C Shell比标准的XENIX Shell(sh) 提供了更大的灵活性和更多的功能。

附录A C语言的可移植性

解释如何书写可在别的XENIX系统上编译的C语言程序。

附录B M4宏处理程序

解释如何用M4 宏处理程序来建立和处理宏。

附录C XENIX设备驱动程序指南

解释如何为XENIX系统书写设备驱动程序。

附录D 连接程序错误信息

按字母顺序列出连接程序所有的错误信息，解释每一条错误信息的含义。

与本手册有关的IBM PC XENIX出版物有：

IBM PC XENIX Installation Guide

IBM PC XENIX 安装指南

IBM PC XENIX System Administration

IBM PC XENIX 系统管理手册

IBM PC XENIX Command Reference

IBM PC XENIX 命令参考手册

IBM PC XENIX Programmer's Guide to Library Functions

IBM PC XENIX 库函数程序员手册

IBM PC XENIX Assembler Reference

IBM PC XENIX 汇编程序参考手册

IBM PC XENIX Visual Shell

IBM PC XENIX 直观 shell

IBM PC XENIX Basic Operation Guide

IBM PC XENIX 基本操作指南

IBM PC XENIX C Compiler Reference

IBM PC XENIX C 编译程序参考手册

IBM PC XENIX Software Command Reference

IBM PC XENIX 软件命令参考手册

目 录

第一章 引论	(1)
1.1 概述	(1)
1.2 建立C语言程序	(1)
1.3 建立其它程序	(2)
1.4 库的建立和维护	(2)
1.5 程序源文件的维护	(2)
1.6 用shell命令建立程序	(3)
第二章 CC: C编译程序	(4)
2.1 引言	(4)
2.2 调用C编译程序	(4)
2.3 由C源文件建立程序	(4)
2.3.1 编译一个C源文件	(4)
2.3.2 编译若干个源文件	(5)
2.3.3 输出文件命名	(6)
2.4 建立大、中、小型程序	(6)
2.4.1 -Ms选择项	(7)
2.4.2 -Ms和-i选择项	(7)
2.4.3 -Mm选择项	(7)
2.4.4 -Ml选择项	(7)
2.4.5 建立小型程序	(7)
2.4.6 建立纯正文的小型程序	(7)
2.4.7 建立中型程序	(8)
2.4.8 建立大型程序	(8)
2.5 使用目标文件和库	(8)
2.5.1 建立目标文件	(8)
2.5.2 由目标文件建立程序	(9)
2.5.3 程序与库中函数的连接	(9)
2.6 建立较小较快的程序	(9)
2.6.1 建立优化的目标程序	(10)
2.6.2 删除符号表	(10)
2.6.3 从程序中删去栈检测	(10)
2.7 准备调试用程序	(11)
2.7.1 产生汇编语言清单	(11)
2.7.2 程序的直方图统计	(11)

2.8 控制C预处理程序	(11)
2.8.1 定义宏	(12)
2.8.2 定义包含目录	(12)
2.8.3 忽略缺省包含目录	(12)
2.8.4 保存预处理过的源文件	(13)
2.9 错误信息	(13)
2.9.1 C编译程序信息	(13)
2.9.2 设置警告的级别	(13)
2.10 使用高级选择项	(14)
2.10.1 由汇编语言源文件建立程序	(14)
2.10.2 使用关键字near和far	(14)
2.10.3 设置栈的大小	(15)
2.10.4 使用模块、段和组	(15)
2.10.5 建立DOS程序	(16)
2.11 编译程序小结	(16)
2.11.1 cc选择项	(16)
2.11.2 存贮模式	(17)
2.11.3 指针和整数的大小	(18)
2.11.4 段名和模块名	(18)
第三章 XENIX到DOS: 交叉开发系统	(19)
3.1 引言	(19)
3.2 建立源文件	(19)
3.3 编译DOS源文件	(20)
3.4 使用汇编语言源文件	(20)
3.5 建立连接目标文件	(20)
3.6 DOS程序的运行和调试	(21)
3.7 系统间程序的传送	(21)
3.8 建立DOS库	(21)
第四章 lint程序——C程序检查器	(22)
4.1 引言	(22)
4.2 调用lint	(22)
4.3 检查无用的变量和函数	(23)
4.4 检查局部变量	(23)
4.5 检查不可到达语句	(24)
4.6 检查无限循环	(24)
4.7 检查函数返回值	(25)
4.8 检查无用的返回值	(25)
4.9 类型检查	(25)
4.10 检查类型强制转换	(26)

4.11	检查不可移植字符的使用	(26)
4.12	检查long到int的赋值	(26)
4.13	检查奇怪的结构	(27)
4.14	检查老的C 文法的使用	(27)
4.15	检查指针排列	(28)
4.16	检查表达式计值顺序	(28)
4.17	嵌入伪指令	(29)
4.18	检查库相容性	(29)
第五章	程序的维护程序: make	(31)
5.1	引言	(31)
5.2	建立makefile文件	(31)
5.3	调用make	(32)
5.4	使用伪目标名	(33)
5.5	使用宏	(34)
5.6	使用Shell环境变量	(36)
5.7	使用内部规则	(36)
5.8	改变内部规则	(37)
5.9	使用库	(38)
5.10	故障查找	(39)
5.11	使用make: 一个例子	(40)
第六章	SCCS: 源代码控制系统	(42)
6.1	引言	(42)
6.2	基本信息	(42)
6.2.1	文件和目录	(42)
6.2.2	Delta和SID	(42)
6.2.3	SCCS工作文件	(43)
6.2.4	SCCS命令参数	(44)
6.2.5	文件管理者	(44)
6.3	建立和使用S文件	(44)
6.3.1	建立S文件	(44)
6.3.2	为读入恢复文件	(45)
6.3.3	为编辑恢复文件	(45)
6.3.4	保持文件的新版本	(46)
6.3.5	恢复某一特定版本	(47)
6.3.6	改变文件的发行号	(47)
6.3.7	建立分版本	(48)
6.3.8	恢复分版本	(48)
6.3.9	恢复最新版本	(48)
6.3.10	显示一个版本	(48)

6.3.11	保存新版本的拷贝	(49)
6.3.12	显示帮助信息	(49)
6.4	使用标识关键字	(49)
6.4.1	往文件中插入关键字	(49)
6.4.2	对关键字赋值	(50)
6.4.3	强置关键字	(50)
6.5	使用S文件标志	(50)
6.5.1	设置S文件标志	(51)
6.5.2	使用i标志	(51)
6.5.3	使用d标志	(51)
6.5.4	换用v标志	(51)
6.5.5	删去S文件标志	(51)
6.6	修改S文件信息	(51)
6.6.1	加入注释	(52)
6.6.2	改变注释	(52)
6.6.3	加入修改请求	(52)
6.6.4	改变修改请求	(53)
6.6.5	加入描述正文	(53)
6.7	打印S文件	(53)
6.7.1	使用数据说明	(53)
6.7.2	打印特定版本	(54)
6.7.3	打印较新的和较早的版本	(54)
6.8	由若干用户编辑	(54)
6.8.1	编辑不同版本	(54)
6.8.2	编辑一个版本	(54)
6.8.3	保存一特定版本	(55)
6.9	保护S文件	(55)
6.9.1	向用户表中加入一个用户	(55)
6.9.2	从用户表中删去一个用户	(56)
6.9.3	设置底层标志	(56)
6.9.4	设置顶层标志	(56)
6.9.5	对一版本加锁	(56)
6.10	订正SCCS文件	(56)
6.10.1	检查S文件	(56)
6.10.2	编辑S文件	(57)
6.10.3	改变S文件的校验和	(57)
6.10.4	为编辑重新产生G文件	(57)
6.10.5	恢复受损的P文件	(57)
6.11	使用其它命令选择项	(57)

6.11.1	用SCCS命令获取帮助	(57)
6.11.2	用标准输入建立文件	(58)
6.11.3	从特定发行号开始	(58)
6.11.4	向第一版中加入注释	(58)
6.11.5	禁止正常输出	(58)
6.11.6	包含和排斥Delta	(58)
6.11.7	列出版本的Delta	(59)
6.11.8	映射行到Delta	(59)
6.11.9	行命名	(59)
6.11.10	显示差异表	(59)
6.11.11	显示文件信息	(60)
6.11.12	删去Delta	(60)
6.11.13	对串进行搜索	(60)
6.11.14	比较SCCS文件	(60)
第七章	adb 程序调试器	(62)
7.1	引言	(62)
7.2	启停adb	(62)
7.2.1	用程序文件启动	(62)
7.2.2	用内存映象文件启动	(62)
7.3	用写选择项启动	(63)
7.4	用提示选择项启动	(63)
7.5	退出adb	(63)
7.6	显示指令和数据	(64)
7.6.1	形成地址	(64)
7.6.2	形成表达式	(64)
7.6.2.1	十、八、十六进制整数	(64)
7.6.2.2	符号	(64)
7.6.2.3	adb 中的变量	(65)
7.6.2.4	当前地址	(65)
7.6.2.5	寄存器名	(66)
7.6.2.6	运算符	(66)
7.6.3	选择数据格式	(67)
7.7	使用等于命令	(68)
7.7.1	使用 (?) 命令和反斜线命令	(68)
7.7.2	一个例子: 简单格式化	(69)
7.8	调试程序的执行	(70)
7.9	运行一程序	(70)
7.9.1	设置断点	(71)
7.9.2	显示断点	(71)

7.9.3	继续执行	(71)
7.9.4	用Interrupt和Quit停止程序	(71)
7.9.5	单步执行程序	(72)
7.9.6	抹去一程序	(72)
7.10	删去断点	(72)
7.10.1	显示C语言栈回溯踪迹	(72)
7.10.2	显示CPU 寄存器	(72)
7.10.3	显示外部变量	(73)
7.10.4	一个例子: 跟踪多个函数	(73)
7.11	使用adb内存映象	(76)
7.11.1	显示内存映象	(76)
7.11.2	改变内存映象	(77)
7.11.3	建立新的映象条目	(77)
7.12	验证地址	(78)
7.13	其它特点	(78)
7.13.1	在一行中组合命令	(78)
7.13.2	建立adb原本文件	(78)
7.13.3	设置输出宽度	(79)
7.13.4	设置最大偏移量	(79)
7.14	设置缺省输入格式	(79)
7.15	使用XENIX命令	(80)
7.16	计算数值和显示正文	(80)
7.17	一个例子: 目录和I节点	(80)
7.18	修补二进制文件	(81)
7.19	在文件中确定值的位置	(82)
7.20	写文件	(82)
7.21	修改内存	(82)
第八章	lex 程序: 词法分析器	(83)
8.1	引言	(83)
8.2	lex源格式	(84)
8.3	lex正则表达式	(85)
8.4	调用lex	(85)
8.5	说明字符类	(86)
8.6	说明一任意字符	(86)
8.7	说明可选表达式	(86)
8.8	说明重复表达式	(86)
8.9	说明选择和分组	(87)
8.10	说明上下文相关性	(87)
8.11	说明表达式重复	(87)

8.12	说明定义	(88)
8.13	说明动作	(88)
8.14	处理模糊源规则	(90)
8.15	说明左上下文相关性	(92)
8.16	说明源定义	(94)
8.17	程序lex和yacc	(95)
8.18	说明字符集	(98)
8.19	源格式	(98)
第九章	yacc程序产生器: 编译程序的编译程序	(101)
9.1	引言	(101)
9.2	规范说明	(103)
9.3	动作	(104)
9.4	词法分析	(106)
9.5	语法分析程序的工作过程	(107)
9.6	二义性和冲突	(110)
9.7	优先级	(113)
9.8	出错处理	(115)
9.9	yacc的环境	(117)
9.10	准备规范说明	(118)
9.11	输入风格	(118)
9.12	左递归	(118)
9.13	词法上的标注	(119)
9.14	处理保留字	(119)
9.15	在动作中模拟出错处理和接收	(120)
9.16	在封闭规则中访问值	(120)
9.17	对任意值类型的支持	(120)
9.18	一个小型台式计算器	(122)
9.19	yacc的输入语法	(122)
9.20	一个高级例子	(124)
9.21	早先的特性	(128)
第十章	C shell	(130)
10.1	引言	(130)
10.2	调用C shell	(130)
10.3	使用shell变量	(131)
10.4	使用C shell历史列表	(132)
10.5	使用别名	(134)
10.6	改向输入和输出	(135)
10.7	建立后台和前台作业	(135)
10.8	使用内部命令	(136)

10.9	建立命令原本	(137)
10.10	使用argv变量	(137)
10.11	替代shell变量	(137)
10.12	使用表达式	(139)
10.13	使用C shell: 一个实例原本	(139)
10.14	使用其它控制结构	(141)
10.15	提供命令的输入	(142)
10.16	捕捉中断	(142)
10.17	使用其它特性	(143)
10.18	在终端上启动一个循环	(143)
10.19	使用带有参数的花括号	(144)
10.20	替换命令	(144)
10.21	特殊字符	(144)
10.21.1	句法元字符	(144)
10.21.2	文件名元字符	(144)
10.21.3	引用元字符	(145)
10.21.4	输入/输出元字符	(145)
10.21.5	扩展/替换元字符	(145)
10.21.6	其它元字符	(145)
附录A	C语言的可移植性	(146)
A.1	引言	(146)
A.2	程序的可移植性	(146)
A.3	机器硬件	(147)
A.3.1	字节长度	(147)
A.3.2	字长	(147)
A.3.3	存储器排列	(147)
A.3.4	字中的字节顺序	(148)
A.3.5	位段	(148)
A.3.6	指针	(149)
A.3.7	地址空间	(149)
A.3.8	字符集	(149)
A.4	编译程序的差异	(150)
A.4.1	有符号/无符号字符, 符号扩展	(150)
A.4.1	移位操作	(150)
A.4.3	标识符长度	(150)
A.4.4	寄存器变量	(150)
A.4.5	类型转换	(151)
A.4.6	带有可变个参数的函数	(151)
A.4.7	副作用, 计值顺序	(152)

A.5	程序环境差异	(153)
A.6	数据的可移植性	(153)
A.7	lint C程序检查器	(154)
A.8	字节顺序小结	(154)
附录B	m4宏处理程序	(155)
B.1	引言	(155)
B.2	调用m4	(155)
B.3	定义宏	(155)
B.4	加引号	(156)
B.5	使用参数	(157)
B.6	使用内部算术运算	(158)
B.7	文件操作	(159)
B.8	使用系统命令	(159)
B.9	使用条件	(159)
B.10	使用串	(160)
B.11	打印	(161)
附录C	XENIX设备驱动程序指南	(162)
C.1	引言	(162)
C.2	预备知识	(162)
C.3	字符设备	(164)
C.4	块设备	(165)
C.5	配置与安装	(166)
C.6	警告	(167)
C.7	行式打印机驱动程序实例	(167)
C.7.1	lpopen ()	(168)
C.7.2	lpclose ()	(168)
C.7.3	lpwrite ()	(168)
C.7.4	lpstart ()	(168)
C.7.5	lpintr ()	(168)
C.8	终端驱动程序实例	(170)
C.8.1	tdopen ()	(171)
C.8.2	tdclose ()	(171)
C.8.3	tdread () tdwrite ()	(171)
C.8.4	tdparam ()	(171)
C.8.5	tmodem ()	(172)
C.8.6	tdintr ()	(172)
C.8.7	tdxint ()	(172)
C.8.8	tdrint ()	(172)
C.8.9	tdmint ()	(172)

C.8.10	tdioctl ()	(172)
C.8.11	tdproc ()	(172)
C.9	磁盘驱动程序实例	(181)
C.9.1	hdstrategy	(182)
C.9.2	hdstart ()	(182)
C.9.3	hdintr ()	(182)
C.9.4	hdread ()	(182)
C.9.5	hdwrite ()	(182)
附录D	连接程序错误信息	(187)

第一章 引 论

1.1 概述

IBM PC XENIX软件开发系统为帮助你设计、开发应用程序和系统程序提供了一系列的程序和命令。这些程序和命令使你能建立在XENIX系统上运行的C语言程序和汇编语言程序。还让你调试这些程序、自动建立程序，并维护你所开发的不同版本。make程序可以节省许多步骤，因为它可以使你不必人工地去做每一步。

本指南用一些特殊符号来描述XENIX命令的文法。下面列出这些符号及其含义：

- [] 方括号表示可选择的命令参数。
- ... 省略号表示前面的参数可重复一次或多次。
- 粗体字** 粗体字表示命令名和程序名。
- 斜体字* 斜体字表示命令参数的位置占据者。打入命令时，这个位置必须代之以适合的文件名、数字或选择项。

下面各节介绍IBM PC XENIX软件系统的程序和命令，解释为IBM PC XENIX系统开发程序所可能采取的步骤。本章的几节中所介绍的大部分命令和程序将在本指南的后续各章中详细解释。这里提及的一些命令是IBM PC XENIX分时系统的一部分。在《IBM PC XENIX基本操作指南》和《IBM PC XENIX系统管理手册》中有它们的解释。

1.2 建立C语言程序

所有的C语言程序都是以源文件中C程序语句的集合开始。XENIX系统提供了许多正文编辑程序，使你方便有效地建立源文件。最方便的编辑程序是面向屏幕的编辑程序vi，屏幕编辑程序vi提供了许多命令使你方便地对正文作增、删、改、查询。所有的命令都可从命令键或从命令行调用。编辑程序vi还有各种选择项使你修改它的操作。

一旦将一个C语言程序写入源文件中，便可用cc命令建立一个可执行程序。CC命令调用XENIX的C编译程序，对源文件进行编译。这个命令还调用其它的XENIX程序，如连接程序和汇编程序，为已编译的程序的执行作准备。

可用XENIX的调试程序adb，对可执行的C程序进行调试。调试程序adb为修改可执行程序的机器指令提供直接接口。

如果希望在编译前对程序检查，可使用lint——XENIX C程序检查程序。lint程序检查C语言程序的内容和结构，查出句法上的和逻辑上的错误。它还实施一套合适的C程序设计风格的严格规则。lint程序主要是在程序开发的早期阶段用来检查C语言的非法的和不恰当的使用。

另一种检查程序的方法是使用cb-XENIX C程序美化程序。cb程序改善C程序的可读性，使之易于检查其逻辑错误。