



UNIX 系统 V/386 第 4 版
集成软件开发指南

UNIX[®] SYSTEM V/386
RELEASE 4

Integrated Software Development Guide



江苏工业学院图书馆
藏书章

UNIX Software Operation

電子工業出版社

目 录

第一章 引论	(1)
1.1 引言	(1)
1.1.1 本指南的目的	(1)
1.1.2 本指南的内容	(1)
1.1.3 如何使用本指南	(2)
1.2 符号约定	(2)
1.2.1 有关的文档	(3)

第二章 应用软件打包	(5)
2.1 软件打包概述	(5)
2.2 软件包的内容	(5)
2.3 软件包的结构生命周期	(6)
2.4 打包工具	(6)
2.4.1 pkgmk 命令	(6)
2.4.2 pkgtrans 命令	(7)
2.4.3 pkgproto 命令	(7)
2.4.4 prototype 文件	(7)
2.4.4.1 描述行	(8)
2.4.4.2 命令行	(9)
2.5 安装工具	(10)
2.6 软件包信息文件	(10)
2.6.1 pkginfo 文件	(11)
2.6.2 pkgmap 文件	(11)
2.6.3 compver 文件	(12)
2.6.4 depend 文件	(12)
2.6.5 space 文件	(13)
2.6.6 copyright 文件	(14)
2.6.7 request 文件	(14)

2.7 安装脚本 (14)

2.7.1 脚本处理 (15)

2.7.2 安装参数 (16)

2.7.3 为脚本获取软件包信息 (16)

2.7.4 脚本的出口码 (17)

2.7.5 请求脚本 (17)

2.7.5.1 请求脚本的命名约定 (17)

2.7.5.2 请求脚本的使用规则 (18)

2.7.6 分类动作脚本 (18)

2.7.6.1 分类动作脚本的命名约定 (18)

2.7.6.2 分类动作脚本的使用规则 (18)

2.7.6.3 类的安装 (19)

2.7.6.4 类的删除 (20)

2.7.7 特别系统类 (20)

2.7.7.1 sed 类脚本 (20)

2.7.7.2 awk 类脚本 (21)

2.7.7.3 build 类脚本 (22)

2.7.8 过程脚本 (22)

2.7.8.1 过程脚本的命名约定 (22)

2.7.8.2 过程脚本的使用规则 (23)

2.8 打包的基本步骤 (23)

2.8.1 第一步 指定软件包的简称 (24)

2.8.2 第二步 定义一个软件包实例 (24)

2.8.2.1 标识软件包实例 (24)

2.8.2.2 在脚本中引用实例标识符 (25)

2.8.3 第三步 将对象分类 (25)

2.8.4 第四步 使软件包对象可重定位 (26)

2.8.4.1 定义群体可重定位对象 (26)

2.8.4.2 定义个体可重定位对象 (26)

2.8.5 第五步 书写安装脚本 (27)

2.8.6 第六步 定义软件包的相关性 (27)

2.8.7 第七步 书写版权消息 (28)

2.8.8 第八步 创建 pkginfo 文件 (28)

2.8.9 第九步 创建 prototype 文件 (28)

2.8.9.1 手工创建 prototype 文件 (28)

2.8.9.2 用 pkgproto 创建 prototype 文件 (30)

2.8.10 第十步 将软件包分布在多个卷上	(32)
2.8.11 第十一步 用 pkgmk 创建一个软件包	(32)
2.8.11.1 pkgmk 创建一个软件包实例	(33)
2.8.11.2 帮助 pkgmk 确定软件包内容的位置	(33)
2.8.12 第十二步 用 pkgtrans 创建一个软件包	(33)
2.8.12.1 创建一个数据流软件包	(34)
2.8.12.2 转换一个软件包实例	(34)

第三章 设备驱动程序

3.1 设备驱动程序	(35)
3.1.1 什么是 UNIX 的设备驱动程序	(35)
3.1.2 一般的 UNIX 驱动程序	(36)
3.2 驱动程序的任务及责任	(36)
3.2.0.1 系统缓冲区	(36)
3.2.0.2 系统与用户空间之间的数据传输	(37)
3.2.0.3 进程的睡眠和唤醒	(37)
3.2.0.4 核心计时器	(38)
3.2.0.5 驱动程序的同步段和中断段	(38)
3.2.0.6 中断处理	(39)
3.2.0.7 驱动程序的临界区	(39)
3.2.0.8 如何在核心和设备之间移动数据	(40)
3.2.0.9 DMA 分配例程	(42)
3.3 UNIX 系统的驱动程序特点	(42)
3.3.0.1 设备的类型	(42)
3.3.0.2 特别文件	(42)
3.3.0.3 主设备号和次设备号	(43)
3.3.0.4 /dev 目录	(43)
3.3.0.5 主文件和系统文件	(44)
3.3.1 设备驱动程序源文件的结构	(44)
3.3.1.1 包含文件	(44)
3.3.1.2 通用系统数据结构	(45)
3.3.1.3 驱动程序专用的数据结构	(46)
3.3.1.4 设备标志 devflag	(46)
3.4 函数说明(驱动程序入口点)	(47)

3.4.0.1	Poll	(47)
3.4.0.2	Kenter	(47)
3.4.0.3	Kexit	(48)
3.4.0.4	中断处理程序	(48)
3.4.0.5	共享中断与 DMA 通道	(49)
3.4.0.6	函数命名约定	(49)
3.5	系统实用函数	(49)
3.5.0.1	睡眠与唤醒	(49)
3.5.0.2	延迟函数	(50)
3.5.0.3	块驱动程序的 biowait/biodone 事件的同步	(51)
3.5.0.4	设置处理器优先级	(51)
3.5.0.5	中断优先级	(53)
3.5.0.6	睡眠优先级	(53)
3.5.0.7	Timeout	(55)
3.5.1	出错报告	(56)
3.5.2	分配缓冲区空间	(56)
3.5.2.1	核心内存分配	(56)
3.5.2.2	缓冲区池	(57)
3.5.3	可安装驱动程序的实现	(58)
3.5.4	ID 概要	(58)
3.6	控制器接口基础	(58)
3.6.0.1	中断	(59)
3.6.0.2	I/O 地址和控制器存储地址	(60)
3.6.0.3	DMA 控制器操作	(60)
3.6.1	用户接口	(61)
3.6.1.1	用户特权	(61)
3.6.1.2	与 SVR 4.0/386 中的其它进程间的相互影响	(61)
3.6.2	可安装的驱动程序个数	(61)
3.7	对 ID 的修改	(62)
3.7.0.1	主文件	(62)
3.7.0.2	系统文件	(62)
3.7.0.3	space-c	(62)
3.7.0.4	ID 目录结构	(62)
3.7.0.5	由配置进程生成的设备 #define	(64)
3.8	安装驱动程序并重构 UNIX 操作系统核心的命令	(65)
3.8.0.1	ldcheck	(65)

3.8.0.2	Idinstall	(65)
3.8.0.3	Idbuild	(65)
3.9	驱动程序软件包	(66)
3.9.0.1	Driver.o(必需的)	(67)
3.9.0.2	Master(必需的)	(67)
3.9.0.3	System(必需的)	(67)
3.9.0.4	Space.c(任选的)	(68)
3.9.0.5	Node(任选的)	(68)
3.9.0.6	Init(任选的)	(68)
3.9.0.7	Rc(任选的)	(69)
3.9.0.8	Shutdown(任选的)	(69)
3.9.0.9	Postinstall(必需的)	(69)
3.9.0.10	Preremove(必需的)	(70)
3.9.0.11	模块小结	(71)
3.9.1	基本系统驱动程序	(72)
3.9.2	更新驱动程序软件包	(73)
3.9.3	安装/删除小结	(73)
3.9.4	可调的系统参数	(74)
3.9.5	修改一个现有的核心参数	(74)
3.9.6	定义一个新的核心参数	(75)
3.9.7	重新配置核心以启用新参数	(75)
3.9.8	设备驱动程序开发方法	(75)
3.9.9	驱动程序开发过程	(75)
3.9.9.1	紧急状态下的恢复(新的核心无法引导)	(77)
3.10	驱动程序的调试	(78)
3.10.0.1	核心打印语句	(78)
3.10.0.2	跟踪驱动程序	(79)
3.10.0.3	系统的致命错误	(79)
3.10.0.4	将系统卸出	(79)
3.10.0.5	postinstall 脚本	(81)
3.10.0.6	preremove 脚本	(81)
3.10.0.7	如何为驱动程序的安装编制文档	(81)
3.10.1	将 XENIX 系统 V/386 设备驱动程序转换为 SVR 4.0/386 的设备驱动程序	(82)

第四章 兼容性方面的考虑	(85)
4.1 引言	(85)
4.2 UNIX Release 3.0、3.1 和 3.2	(85)
4.2.1 打包中的变化	(85)
4.2.2 文件系统	(86)
4.2.2.1 文件系统类型	(86)
4.2.2.2 /proc 文件系统类型	(89)
4.2.2.3 目录的布局	(89)
4.2.2.4 符号连结	(90)
4.2.3 STREAMS 驱动程序	(93)
4.2.4 网络服务	(94)
4.2.5 国际化	(95)
4.2.6 存储分配	(99)
4.2.6.1 核心存储分配程序	(99)
4.2.6.2 增强的加锁功能	(99)
4.2.7 标准 C 开发环境 (SCDE)	(100)
4.2.7.1 编译程序的变化	(100)
4.2.7.2 浮点变化	(100)
4.2.7.3 排错变化	(101)
4.2.7.4 可扩展的连结格式 (ELF)	(101)
4.2.7.5 动态共享库	(102)
4.2.8 可调参数: 动态及静态	(103)
4.2.9 扩展的基本类型: EFT	(104)
4.2.9.1 基本的 EFT 数据类型	(104)
4.2.9.2 EFT 兼容性问题	(105)
4.2.10 其它	(107)

第五章 视象接口	(109)
5.1 引言	(109)
5.2 视象显示程序设计的概念性概述	(110)
5.2.1 视象适配器板	(110)
5.2.2 存储区	(110)

5.2.3	寄存器	(110)
5.2.4	正文方式和图形方式	(110)
5.2.4.1	正文方式	(111)
5.2.4.2	图形方式	(111)
5.2.5	程序设计的步骤	(111)
5.2.5.1	开始之前	(111)
5.2.5.2	正文方式	(112)
5.2.5.3	图形方式	(112)
5.3	视象显示程序设计的技术性概述	(113)
5.3.1	视象适配器板	(113)
5.3.1.1	确定适配器类型	(113)
5.3.1.2	方式的获取与设置	(115)
5.4	正文方式下的程序设计	(117)
5.4.1	正文是如何存放的	(117)
5.4.2	字符集	(117)
5.4.3	选择正文方式	(118)
5.4.4	转义序列	(118)
5.4.5	正文方式程序设计的例子	(119)
5.4.6	正文程序设计存储管理	(121)
5.4.6.1	存储布局	(121)
5.5	存取视象存储区的程序设计	(124)
5.5.0.1	取得物理地址	(125)
5.5.0.2	映射视象存储区	(125)
5.5.0.3	映射举例	(126)
5.5.0.4	与 MAPCONS 的关系	(126)
5.6	图形方式	(126)
5.6.1	存取图形方式的视象存储区	(126)
5.6.1.1	在 EGA 图形方式下对视象存储区寻址	(131)
5.6.1.2	VGA 独有的寻址方式	(133)
5.6.1.3	方式切换	(133)
5.7	存取视象控制器寄存器	(133)
5.7.0.1	利用寄存器提高效率	(135)
5.7.0.2	寄存器编程举例	(135)
5.8	使用虚拟终端	(138)
5.8.1	引言	(138)
5.8.2	终端用户使用虚拟终端	(139)

5.8.3	程序设计的特点	(140)
5.8.3.1	VT 操作方式	(140)
5.8.4	书写性能良好的程序	(141)
5.8.4.1	性能良好的图形应用程序举例	(141)
5.8.5	编写虚拟终端使用管理程序	(147)
5.8.6	虚拟终端的创建和应用	(150)
5.8.7	确定 VT 状态	(153)
5.8.8	虚拟终端控制	(155)
5.9	其它功能	(161)
5.9.1	设置边界	(161)
5.9.2	键盘操作	(161)
5.9.3	声音效果	(161)
5.9.4	字体操作	(161)
5.9.4.1	替换整个字库	(162)
5.9.4.2	替换字库中的个别字符	(162)
5.9.5	对鼠标器编程	(162)

附录 A	liber, 一个图书馆系统	(169)
------	----------------------	-------

附录 B	软件包安装实例分析	(181)
------	-----------------	-------

B.1	引言	(181)
B.2	实例#1	(181)
B.2.1	技术	(181)
B.2.2	方法	(181)
B.2.3	样本文件	(182)
B.3	实例#2	(185)
B.3.1	技术	(185)
B.3.2	方法	(185)
B.3.3	样本文件	(186)
B.4	实例#3	(189)
B.4.1	技术	(189)
B.4.2	方法	(189)
B.4.3	样本文件	(190)
B.5	实例#4	(192)

B.5.1 技术	(192)
B.5.2 方法	(192)
B.5.3 样本文件	(193)
B.6 实例#5a	(194)
B.6.1 技术	(194)
B.6.2 方法	(195)
B.6.3 样本文件	(195)
B.7 实例#5b	(197)
B.7.1 技术	(197)
B.7.2 方法	(197)
B.7.3 样本文件	(198)
B.8 实例#5c	(200)
B.8.1 技术	(200)
B.8.2 方法	(200)
B.8.3 样本文件	(201)
B.9 实例#6	(202)
B.9.1 技术	(202)
B.9.2 方法	(202)
B.9.3 样本文件	(203)

附录 C 跟踪驱动程序	(207)
-------------------	-------

附录 D 一个软盘驱动程序原型	(219)
-----------------------	-------

附录 E 一个驱动程序软件包样本	(241)
------------------------	-------

附录 F 综合视象程序设计举例	(249)
-----------------------	-------

附录 G 图形方式小结	(265)
G.1 SW-BG320	(265)
G.2 SW-CG320	(265)
G.3 SW-BG640	(266)

G.4	SW-CG320-D	(266)
G.5	SW-CG640-W	(267)
G.6	SW-CG640×350	(267)
G.7	SW-ENH-CG640	(268)
G.8	SW-VGA640×480C	(268)
G.9	SW-VGA640×480E	(268)
G.10	SW-VGA320×200	(269)
G.11	SW-ATT640	(269)
G.12	SW-VDC800×600E	(269)
G.13	SW-VDC640×400V	(270)

附录 H 正文和图形方式 IOCTLs

H.1	正文方式选择 IOCTLs	(271)
H.2	图形方式选择 IOCTLs	(271)
H.3	display(7) ioctl 小结	(272)

附录 I UNIX 兼容性表

I.1	引言	(275)
I.1.1	参考表格的用法	(275)
I.2	参考表	(277)

附录 J 地区和时区环境变量

J.1	引言	(303)
J.2	LANG 环境变量	(303)
J.3	TZ 环境变量	(304)

图和表

图 2-1	#lprototype 文件样本	(9)
图 2-2	#lprototype 文件样本	(10)
图 2-3	pkginfo 文件样本	(11)
图 2-4	compver 文件样本	(12)

图 2-5	depend 文件样本	(13)
图 2-6	space 文件样本	(14)
图 2-7	copyright 文件样本	(14)
图 2-8	将参数置入安装环境	(17)
图 2-9	sed 脚本格式	(21)
图 2-10	awk 脚本格式	(22)
表 3-1	驱动程序软件包的组成模块	(71)
表 4-1	基本 EFT 数据类型	(105)
图 B-1	实例#1 pkginfo 文件	(182)
图 B-2	实例#1 prototype 文件	(183)
图 B-3	实例分析#1 请求脚本	(184)
图 B-4	实例#2 prototype 文件	(186)
图 B-5	实例#2 pkginfo 文件	(187)
图 B-6	实例#2 请求脚本	(187)
图 B-7	实例#2 postinstall 脚本	(188)
图 B-8	实例#3 pkginfo 文件	(190)
图 B-9	实例#3 prototype 文件	(190)
图 B-10	实例#3 space 文件	(191)
图 B-11	实例#3 安装类动作脚本(i•admin)	(191)
图 B-12	实例#3 删除类动作脚本(r•cfgdata)	(192)
图 B-13	实例#5 pkginfo 文件	(193)
图 B-14	实例#4 copyright 文件	(193)
图 B-15	实例#4 compver 文件	(194)
图 B-16	实例#4 depend 文件	(194)
图 B-17	实例#5a pkginfo 文件	(196)
图 B-18	实例#5a prototype 文件	(196)
图 B-19	实例#5a sed 脚本(/sbin/inittab)	(196)
图 B-20	实例#5a postinstall 脚本	(197)
图 B-21	实例#5b pkginfo 文件	(198)
图 B-22	实例#5b prototype 文件	(198)
图 B-23	实例#5b 安装类动作脚本(i•inittab)	(199)
图 B-24	实例#5b 删除类动作脚本(r•inittab)	(199)
图 B-25	实例#5b inittab 文件	(200)
图 B-26	实例#5c pkginfo 文件	(201)
图 B-27	实例#5c prototype 文件	(201)
图 B-28	实例#5c build 脚本(/sbin/inittab)	(202)

图 B-29	实例#6 pkginfo 文件	(203)
图 B-30	实例#6 prototype 文件.....	(204)
图 B-31	实例#6 安装类别动作脚本 (i•cron)	(204)
图 B-32	实例#6 删除类动作脚本 (r•cron)	(205)
图 B-33	实例#6 根 crontab 文件(与软件包一起传送)	(205)
图 B-34	实例#6 sys crontab 文件(与软件包一起传送)	(205)

第一章 引 论

1.1 引言

1.1.1 本指南的目的

有些软件厂家专门开发在 386 计算机系统上运行的 UNIX 系统应用软件，本《集成软件开发指南 (ISDG)》是为这些独立软件厂家准备的。针对 Intel 80x86 体系结构的 UNIX System V R4.0 V1.0(在后文中简称 SVR 4.0/386) 新附加的硬件，本指南提供了书写应用软件和可安装的驱动程序时所需要的信息，其中包括对键盘、屏幕、远程终端和所支持的打印子系统的使用说明。另外，本指南还提供了基于 XENIX 的软件移植到本系统上来的方法。

使用本指南，有助于确保所开发软件与现有 386 处理器以及将来扩展后的处理器相兼容。

1.1.2 本指南的内容

本指南分成如下章节：

- 第一章，“引论”，略述本指南的内容及其使用方法。
- 第二章，“UNIX 应用软件的安装”，概括了安装 UNIX 系统软件的过程，并且提供了在为计算机建立软件安装盘片时所必需了解的细节。另外，还提供了一些安装和删除 UNIX 程序的概括准则，以及一些安装和删除程序脚本的例子。
- 第三章，“设备驱动程序”，给出编写 SVR 4.0/386 设备驱动程序时所要遵循的规则和过程。你应该知道，书写设备驱动程序所担负的责任是很大的。作为 UNIX 操作系统核心的一部分，设备驱动程序必需正确执行。
- 第四章，“兼容性方面的考虑”，给出了一些必需的基本信息，以简化把 V3.2 上的应用程序转移到 SVR 4.0 上的移植过程。
- 第五章，“视象接口”，描述了如何管理正文和图形的输出。
- 附录 A，“Liber，一个图书馆系统”，用一个图书馆系统的开发，来示范在开发应用程序时如何使用 UNIX 系统程序设计工具。
- 附录 B，“软件包安装实例分析”，用一组软件包实例，来示范打包技术。
- 附录 C，“跟踪驱动程序”，描述一个被称作“跟踪驱动程序”的伪设备。利用它，UNIX 操作系统核心或其它设备驱动程序可以报告排错信息，而无需使用控制

台 printf 语句。

- 附录D,“一个软盘驱动程序原型”,这里选用了SVR 4.0/386软盘设备驱动程序的源文件的一部分。
- 附录E,“一个驱动程序软件包样本”,给出了安装一个设备驱动程序所需要的ID模块,并描述了 Install 和 Remove 脚本。
- 附录F,“视象程序设计的综合示例”,示范了使用图形适配器的大部分ioctl命令和技术。
- 附录G,“图形方式小结”,总结了各种图形方式的存储需求、存储布局以及象素和颜色布局。
- 附录H,“正文和图形方式Ioctl”,对参考手册中详细描述的所有图形ioctl命令作了一个小结。
- 附录I,“UNIX兼容性表”,表中列出所有的用户和管理命令、系统调用、库子例程、文件、文件格式和系统维护过程,说明了其中哪些内容在 V4.0 中改变了,哪些内容是 V4.0 新增加的,哪些原有的功能现在已经过时了。
- 附录J,“地区和时区环境变量”,描述了环境变量LANG和TZ的使用方法,以使应用程序适用于国际化的使用环境。

1.1.3 如何使用本指南

要有效地使用本指南,读者应具备 UNIX 操作系统的工作经验。本指南的目的在于指导读者设计设备驱动程序及其它软件。

1.2 符号约定

本指南使用下列符号约定:

黑体字	用户输入的符号,如命令、命令的选择项以及目录和文件的名字都以黑体字印出。
斜体字	必须赋值的变量名(如文件名)以斜体字印出。
等宽字	UNIX系统输出的符号,如提示符、命令的回答以及编程示例,都以等宽字印出。
< >	键入时不出现在屏幕上的输入的符号,如口令、用作命令的键或者回车及其它特殊键,都括在尖括号中。
<^ char>	控制字符括在尖括号中,因为在键入时,它们不出现在屏幕上。上尖角字符^表示控制键(通常记作 CTRL)。为了键入一个控制字符,你应在按下控制键的同时,键入由 char 指定的字符。例如,<^ d>表示在你按下控制键的同时按下 D 键;字母 D 不在屏幕上

出现。

[] 命令选择项以及可选的参数。如[-msCj]，被括在方括号中。
| 你可在由竖杠分开的参数中选择一个。例如，如果命令行的格式为：

`command [ arg1 | arg2 ]`

那么，在发出命令时你就可以选择`arg1`或者选择`arg2`。

... 参数后面的省略号表示在一个命令行上可以使用多个参数。在编程示例中一个竖直的省略号表示省略的代码部分。

`command (number)`

命令名后面的括号内加一个数字表示在某个UNIX系统参考手册中该命令所在的部分。(有两个参考手册：《用户/系统管理员参考手册》和《程序员参考手册》。)例如，`cat(1)`表示《用户/系统管理员参考手册》的第1节中描写`cat`命令的页。

1.2.1 有关的文档

UNIX 系统 V/386 有各种文档。要了解它们，可参照《产品概览和总索引》。

《产品概览和总索引》帮助你了解文档之间的一般关系，以便找出需要订购的文档。

本书将要引用《产品概况和总索引》中列出的一些文档。为方便起见，我们采用如下约定：

- 将《UNIX系统V/386第4版程序员指南》简称为《程序员指南》。
- 将《UNIX系统V/386第4版程序员参考手册》简称为《程序员参考手册》。
- 将《UNIX系统V/386第4版系统管理员参考手册》简称为《系统管理员参考手册》。
- 将《UNIX系统V/386第4版用户参考手册》简称为《用户参考手册》。
- 将《UNIX系统V/386第4版设备驱动程序界面/驱动程序-核心界面(DDI/DKI)参考手册》简称《设备驱动程序参考手册》。