



UNIX系统V第4版
程序员指南：网络界面

*UNIX® SYSTEM V
RELEASE 4*

*Programmer's Guide:
Networking Interfaces*



UNIX Software Operation

電子工業出版社

目 录

第一篇 TLI 和 Socket 编程

第一章 TLI 和 Socket 编程引论	(1)
1.1 本文档的组织	(1)
1.2 网络选择和名字到地址的映射	(1)
第二章 传输界面编程	(3)
2.1 引言	(3)
2.2 背景	(3)
2.3 本章文档的组织	(4)
2.4 传输界面概述	(5)
2.4.1 服务方式	(6)
2.4.1.1 连接方式服务	(6)
2.4.1.2 非连接方式服务	(10)
2.4.2 状态转换	(10)
2.5 连接方式服务介绍	(11)
2.5.1 本地管理	(11)
2.5.1.1 委托方	(12)
2.5.1.2 服务方	(13)
2.5.2 建立连接	(16)
2.5.2.1 委托方	(17)
2.5.2.2 事件处理	(18)
2.5.2.3 服务方	(18)
2.5.3 数据传送	(22)
2.5.3.1 委托方	(22)
2.5.3.2 服务方	(23)
2.5.4 释放连接	(25)
2.5.4.1 服务方	(26)
2.5.4.2 委托方	(26)
2.6 非连接方式服务介绍	(27)

2.6.1 本地管理	(27)
2.6.2 数据传送	(29)
2.6.3 数据报出错	(31)
2.7 读/写界面	(32)
2.7.1 write	(33)
2.7.2 read	(33)
2.7.3 close	(34)
2.8 高级专题	(34)
2.8.1 异步执行方式	(34)
2.8.2 高级编程示例	(35)
2.9 状态转换	(41)
2.9.1 传输界面状态	(41)
2.9.2 发出事件	(41)
2.9.3 发来事件	(42)
2.9.4 传输用户动作	(43)
2.9.5 状态表	(43)
2.10 协议无关性准则	(45)
2.11 示例	(46)
2.11.1 连接方式委托方	(46)
2.11.2 连接方式服务方	(48)
2.11.3 非连接方式事务服务方	(51)
2.11.4 读/写委托方	(53)
2.11.5 事件驱动服务方	(54)
2.12 词汇表	(59)
第三章 Socket 界面	(63)
3.1 背景	(63)
3.2 基础	(63)
3.2.1 Socket 的类型	(63)
3.2.2 Socket 的创建	(64)
3.2.3 联编本地名字	(65)
3.2.4 连接的建立	(66)
3.2.5 数据传输	(68)
3.2.6 关闭 Socket	(68)
3.2.7 非连接 Socket	(71)

3.2.8 输入/输出多路复用	(74)
3.3 支持的例程	(77)
3.3.1 主机名字	(78)
3.3.2 网络名字	(78)
3.3.3 协议名字	(79)
3.3.4 服务名字	(79)
3.3.5 其它	(80)
3.4 委托方/服务方模式	(82)
3.4.1 服务方	(82)
3.4.2 委托方	(85)
3.4.3 非连接的服务方	(86)
3.5 高级专题	(89)
3.5.1 不同波段数据	(89)
3.5.2 非阻塞 Socket	(91)
3.5.3 中断驱动的 Socket I/O	(92)
3.5.4 信号与进程组	(92)
3.5.5 选择具体协议	(94)
3.5.6 地址联编	(94)
3.5.7 广播和确定网络配置	(96)
3.5.8 Socket 选项	(99)
3.5.9 inetd	(100)

第四章 Socket 迁移和 Socket 到 TLI 的转换

4.1 Socket 迁移和 Socket 到 TLI 的转换	(103)
4.1.1 连接方式	(103)
4.1.1.1 建立 Socket 连接: 委托方代码	(103)
4.1.1.2 建立 TLI 连接: 委托方代码	(105)
4.1.1.3 建立 Socket 连接: 服务方代码	(106)
4.1.1.4 建立 TLI 连接: 服务方代码	(107)
4.1.2 非连接方式	(110)
4.1.2.1 基于 Socket 的数据报	(110)
4.1.2.2 TLI 数据报	(111)
4.1.3 同步和异步方式	(113)
4.1.4 出错处理	(114)
4.1.5 Socket 到 TLI 的转换	(115)

4.1.6 将 Socket 应用移入 System V R4	(116)
---------------------------------------	-------

第二篇 远程过程呼叫

第五章 远程过程呼叫引论	(121)
--------------------	-------

5.1 RPC 引言	(121)
5.1.1 技术信息的组织	(121)
5.1.2 定义	(121)
5.2 RPC 概述	(122)
5.2.1 RPC 的版本和号码	(124)
5.2.2 网络选择	(124)
5.2.2.1 名字到地址的转换	(126)
5.2.3 rpcbind 设施	(127)
5.2.3.1 地址登记	(127)
5.2.3.2 rpcinfo 命令	(128)
5.2.4 RPC 的较低层次	(128)
5.2.5 外部数据表示	(130)

第六章 rpcgen 编程指南	(131)
-----------------------	-------

6.1 引言	(131)
6.1.1 rpcgen 概述	(131)
6.1.2 技术信息的组织	(131)
6.2 rpcgen 拓导	(132)
6.2.1 把本地过程转换成远程过程	(132)
6.2.2 用 rpcgen 生成 XDR 例程	(138)
6.2.3 使用预处理伪指令	(143)
6.3 通用的 RPC 编程技巧	(144)
6.3.1 网络类型(传输选择)	(144)
6.3.2 改变延时	(145)
6.3.3 委托方认证	(145)
6.3.4 rpcgen 命令行定义语句	(146)
6.3.5 服务方对广播呼叫的应答	(147)
6.3.6 端口监控程序支持	(147)
6.3.7 分派表	(148)

6.3.8 用 rpcgen 测试	(149)
6.4 RPC 语言参考	(150)
6.4.1 定义	(150)
6.4.2 枚举	(150)
6.4.3 常量	(151)
6.4.4 类型定义	(151)
6.4.5 说明	(151)
6.4.6 结构	(152)
6.4.7 联合	(153)
6.4.8 程序	(154)
6.4.9 特殊情况	(155)
第七章 远程过程呼叫编程指南	(157)
7.1 引言	(157)
7.1.1 RPC 软件包概述	(157)
7.1.2 技术信息的组织	(159)
7.2 RPC 的简化界面	(159)
7.2.1 基于 RPC 库的网络服务	(160)
7.2.2 远程过程呼叫和登记	(161)
7.2.2.1 rpc-call() 例程	(161)
7.2.2.2 rpc-reg() 例程	(162)
7.2.2.3 传递任意的数据类型	(163)
7.3 RPC 的较低层次	(167)
7.3.1 顶层	(167)
7.3.1.1 顶层: 委托方侧	(167)
7.3.1.2 顶层: 服务方侧	(169)
7.3.2 中间层	(171)
7.3.2.1 中间层: 委托方侧	(171)
7.3.2.2 中间层: 服务方侧	(172)
7.3.3 专家层	(173)
7.3.3.1 专家层: 委托方侧	(173)
7.3.3.2 专家层: 服务方侧	(176)
7.3.4 底层	(178)
7.3.4.1 底层: 委托方侧	(178)
7.3.4.2 底层: 服务方侧	(178)

7.4 低层数据结构	(179)
7.5 使用原始 RPC 进行低层程序测试	(181)
7.6 高级 RPC 编程技巧	(183)
7.6.1 服务方侧的 select ()	(183)
7.6.2 广播 RPC	(184)
7.6.3 批处理	(185)
7.6.3.1 批处理性能	(188)
7.6.4 认证	(189)
7.6.4.1 AUTH-NONE: 委托方侧	(189)
7.6.4.2 AUTH-NONE: 服务方侧	(189)
7.6.4.3 ANTH-SYS 认证	(190)
7.6.4.4 ANTH-DES 认证	(193)
7.6.5 使用端口监控程序	(195)
7.6.5.1 使用 inetd	(196)
7.6.5.2 使用 listener	(196)
7.7 高级例子	(197)
7.7.1 版本	(198)
7.7.2 面向连接的传输	(199)
7.7.3 回调过程	(202)
7.7.4 用 XDR 分配内存	(206)

第八章 外部数据表示标准: 协议规格

(209)

8.1 XDR 引言	(209)
8.1.1 基本的块大小	(209)
8.1.2 技术信息的组织	(210)
8.2 XDR 数据类型声明	(210)
8.2.1 整数	(210)
8.2.1.1 描述	(210)
8.2.1.2 声明	(210)
8.2.1.3 编码	(210)
8.2.2 无符号整数	(211)
8.2.2.1 描述	(211)
8.2.2.2 声明	(211)
8.2.2.3 编码	(211)
8.2.3 枚举	(211)

8.2.3.1 描述	(211)
8.2.3.2 声明	(211)
8.2.3.3 编码	(211)
8.2.4 布尔量	(211)
8.2.4.1 描述	(211)
8.2.4.2 声明	(212)
8.2.4.3 编码	(212)
8.2.5 双整数和无符号双整数	(212)
8.2.5.1 描述	(212)
8.2.5.2 声明	(212)
8.2.5.3 编码	(212)
8.2.6 浮点数	(212)
8.2.6.1 描述	(212)
8.2.6.2 声明	(213)
8.2.6.3 编码	(213)
8.2.7 双精度浮点数	(213)
8.2.7.1 描述	(213)
8.2.7.2 声明	(213)
8.2.7.3 编码	(213)
8.2.8 定长不透明数据	(214)
8.2.8.1 描述	(214)
8.2.8.2 声明	(214)
8.2.8.3 编码	(214)
8.2.9 可变长不透明数据	(214)
8.2.9.1 描述	(214)
8.2.9.2 声明	(215)
8.2.9.3 编码	(215)
8.2.10 串	(215)
8.2.10.1 描述	(215)
8.2.10.2 声明	(215)
8.2.10.3 编码	(216)
8.2.11 定长数组	(216)
8.2.11.1 描述	(216)
8.2.11.2 声明	(216)
8.2.11.3 编码	(216)
8.2.12 可变长数组	(216)

8.2.12.1 描述	(216)
8.2.12.2 声明	(217)
8.2.12.3 编码	(217)
8.2.13 结构	(217)
8.2.13.1 描述	(217)
8.2.13.2 声明	(217)
8.2.13.3 编码	(217)
8.2.14 加以鉴别的联合	(218)
8.2.14.1 描述	(218)
8.2.14.2 声明	(218)
8.2.14.3 编码	(218)
8.2.15 空	(218)
8.2.15.1 描述	(218)
8.2.15.2 声明	(219)
8.2.15.3 编码	(219)
8.3 其它 XDR 声明	(219)
8.3.1 常量	(219)
8.3.2 typedef	(219)
8.3.3 可选的数据	(220)
8.4 XDR 语言规格	(221)
8.4.1 记号约定	(221)
8.4.2 词法注释	(221)
8.4.3 语法信息	(222)
8.4.3.1 语法注释	(224)
8.5 XDR 数据描述的例子	(224)
8.6 参考文献	(225)

第九章 远程过程呼叫：协议规格 (227)

9.1 引言	(227)
9.1.1 术语	(227)
9.1.2 协议的一般属性	(227)
9.1.2.1 RPC 模式	(227)
9.1.2.2 传输和语义	(228)
9.1.2.3 联编和约定独立性	(228)
9.1.2.4 认证	(228)

9.1.3 技术信息的组织	(229)
9.2 RPC 协议需求	(229)
9.2.1 程序和过程	(229)
9.2.2 认证	(230)
9.2.3 程序号赋值	(230)
9.2.4 RPC 协议的其它用途	(231)
9.2.4.1 成批处理	(231)
9.2.4.2 广播 RPC	(232)
9.3 RPC 消息协议	(232)
9.4 认证协议	(235)
9.4.1 AUTH-NONE 认证	(235)
9.4.2 AUTH-SYS 认证	(235)
9.4.2.1 AUTH-SHORT 验证器	(236)
9.4.3 AUTH-DES 认证	(236)
9.4.3.1 命名	(236)
9.4.3.2 AUTH-DES 认证验证器	(237)
9.4.3.3 别名和时钟同步	(237)
9.4.3.4 DES 认证协议(用 XDR 语言)	(238)
9.4.3.5 Diffie-Hellman 密码	(239)
9.5 记录标记标准	(240)
9.6 RPC 语言	(241)
9.6.1 用 RPC 语言描述的一个服务实例	(241)
9.6.2 RPC 语言规格	(242)
9.6.3 语法注释	(242)
9.7 rpcbind 协议	(243)
9.7.1 rpcbind 协议规格(用 RPC 语言)	(243)
9.7.2 rpcbind 操作	(244)
9.7.2.1 RPCBPROC-NULL 过程	(244)
9.7.2.2 RPCBPROC-SET 过程	(245)
9.7.2.3 RPCBPROC-UNSET 过程	(245)
9.7.2.4 RPCBPROC-GETADDR 过程	(245)
9.7.2.5 RPCBPROC-DUMP 过程	(245)
9.7.2.6 RPCBPROC-CALLIT 过程	(245)
9.7.2.7 RPCBPROC-GETTIME 过程	(246)
9.7.2.8 RPCBPROC-UADDR2TADDR 过程	(246)
9.7.2.9 RPCBPROC-TADDR2UADDR 过程	(246)

9.8 参考文献	(246)
第十章 RPC 管理	(247)
10.1 引言	(247)
10.2 RPC 管理文件	(247)
10.2.1 名字到地址映射	(247)
10.2.2 系统 RC 文件/etc/rc2.d/s75rpc	(247)
10.2.3 /etc/publickey 文件	(250)
10.2.4 /etc/master.d/kernel 文件	(250)
10.3 安全 RPC 概述	(250)
10.3.1 RPC 域	(251)
10.4 安全 RPC 管理	(252)
10.4.1 建立安全 RPC 域	(252)
10.4.2 主/etc/publickey 文件	(253)
10.4.2.1 用 newkey 命令添加 RPC 用户	(253)
10.4.2.2 网络口令和 chkey 命令	(254)
10.4.3 故障查寻注释	(254)
第十一章 YP 服务	(255)
11.1 YP 服务引言	(255)
11.1.1 什么是 YP	(255)
11.1.2 YP 元素	(255)
11.1.3 YP 环境	(256)
11.1.3.1 YP 域	(256)
11.1.3.2 YP 机器类型	(256)
11.1.3.3 YP 映射	(258)
11.2 实现 YP 服务	(259)
11.2.1 建立域	(259)
11.2.2 准备映射	(260)
11.2.2.1 publickey 映射	(260)
11.2.2.2 其它映射	(262)
11.2.3 构造映射	(263)
11.2.3.1 默认的 makefile	(263)
11.2.3.2 更改 makefile	(264)

11.2.4	设置主服务方	(265)
11.2.5	启动主服务方上的精灵进程	(266)
11.2.6	设置从服务方	(267)
11.2.7	启动从服务方上的精灵进程	(269)
11.2.8	装备 YP 委托方	(269)
11.3	管理 YP 映射	(270)
11.3.1	更新已有的映射	(270)
11.3.1.1	更改标准映射	(270)
11.3.1.2	创建和更改非标准映射	(271)
11.3.2	传播 YP 映射	(273)
11.3.2.1	ypxfr 使用 crontab	(273)
11.3.2.2	ypxfr 使用 shell 脚本	(274)
11.3.2.3	直接调用 ypxfr	(275)
11.3.2.4	记录 ypxfr 的活动	(275)
11.3.3	在 makefile 中加入新的 YP 映射	(275)
11.4	在原先集合中加入新的 YP 服务方	(276)
11.4.1	改变映射的主服务方	(277)
11.5	与 YP 相关的命令小结	(278)
11.6	排除 YP 问题	(279)
11.6.1	调试 YP 委托方	(279)
11.6.1.1	在委托方上挂起命令	(279)
11.6.1.2	YP 服务不可用	(282)
11.6.1.3	ypbind 瘫痪	(282)
11.6.1.4	ypwhich 显示不一致	(283)
11.6.2	调试 YP 服务方	(283)
11.6.2.1	服务方有不同的 YP 映射版本	(284)
11.6.2.2	ypserv 瘫痪	(284)
11.7	关闭 YP 服务	(286)

第三篇 网络选择和名字到地址映射

第十二章	网络选择和名字到地址映射	(287)
12.1	网络选择	(287)
12.1.1	网络选择怎样工作	(287)
12.1.2	netconfig 文件	(288)

12.1.3	NETPATH 环境变量	(289)
12.1.4	通过 NETPATH 访问 netconfig 的例程	(290)
12.1.5	直接访问 netconfig 例程	(291)
12.1.6	代码实例	(293)
12.1.6.1	循环通过所有“可见的”netconfig 项	(294)
12.1.6.2	循环通过用户定义的 netconfig 项	(294)
12.1.6.3	循环通过所有的 netconfig 项	(294)
12.1.6.4	规定单个传输供者	(295)
12.2	名字到地址映射	(295)
12.2.1	名字到地址映射的库	(296)
12.2.2	使用名字到地址映射例程	(297)
12.2.2.1	netdir-getbyname	(298)
12.2.2.2	netdir-getbyaddr	(298)
12.2.2.3	netdir-free	(298)
12.2.2.4	taddr2uaddr	(299)
12.2.2.5	uaddr2taddr	(299)
12.2.2.6	netdir-options	(299)
12.2.2.7	netdir-perorr	(300)
12.2.2.8	netdir-sperror	(300)

第四篇 为服务访问设施编写端口监控程序

第十三章	为服务访问设施编写端口监控程序	(303)
13.1	引言	(303)
13.2	服务访问设施概述	(303)
13.2.1	服务访问控制程序	(304)
13.3	基本的端口监控程序功能	(304)
13.3.1	端口管理	(305)
13.3.2	活动监视	(305)
13.3.3	其它的端口监控功能	(305)
13.3.3.1	限制对系统的访问: 启用和禁用端口监控程序和端口	(305)
13.3.3.2	创建 utmp 登记项	(306)
13.3.3.3	端口监控程序进程 ID 和锁定文件	(306)
13.3.3.4	修改服务环境: 运行 doconfig()	(306)
13.3.3.5	文件: 端口监控程序管理文件	(306)

13.3.3.6	文件: 每种服务配置文件	(306)
13.3.3.7	私用端口监控程序文件	(306)
13.3.3.8	终止端口监控程序	(307)
13.4	服务访问控制程序/端口监控程序界面	(307)
13.4.1	消息格式	(307)
13.4.1.1	sac 消息	(307)
13.4.1.2	端口监控程序消息	(308)
13.4.2	消息类	(309)
13.5	端口监控程序管理界面	(309)
13.5.1	SAC 管理文件_sactab	(309)
13.5.2	端口监控程序管理文件_pimtab	(310)
13.5.3	SAC 管理命令 sacadm	(311)
13.5.4	端口监控程序管理命令 pmadm	(312)
13.5.5	端口监控程序的“针对具体端口监控程序的”管理命令	(312)
13.5.6	端口监控程序/服务界面	(312)
13.5.6.1	新的服务引用	(313)
13.5.6.2	常设的服务引用	(313)
13.6	端口监控程序需求	(313)
13.6.1	初始环境	(313)
13.6.2	重要的文件	(313)
13.6.3	责任	(314)
13.6.3.1	各种杂项任务	(314)
13.6.3.2	与任务相关的服务	(314)
13.7	配置文件和配置语言	(314)
13.7.1	配置文件	(314)
13.7.1.1	每个系统配置文件	(314)
13.7.1.2	每个端口监控程序配置文件	(315)
13.7.1.3	每种服务配置文件	(315)
13.7.2	配置语言	(315)
13.7.3	显示、安装和替换配置脚本	(316)
13.7.3.1	每个系统配置脚本	(316)
13.7.3.2	每个端口监控程序配置脚本	(316)
13.7.3.3	每种服务配置脚本	(317)
13.7.4	解释配置脚本: doconfig()	(317)
13.7.5	样本配置脚本	(318)
13.7.5.1	每个系统样本配置脚本	(318)

13.7.5.2 每个端口监控程序样本配置脚本	(318)
13.7.5.3 每种服务样本配置脚本	(318)
13.8 端口监控程序代码样本	(319)
13.9 前导文件 sac.h	(322)
13.10 服务访问设施逻辑图和目录结构	(324)
附录 A 手册页	(327)

图和表

图2-1: OSI参考模型	(3)
图2-2: 传输界面	(5)
图2-3: 用户和提供者间的通道	(7)
图2-4: 传输连接	(8)
图2-5: 监听和回答传输端点	(21)
图3-1: 初始化一个Internet域的流连接	(69)
图3-2: 接受一个Internet域的流连接	(70)
图3-3: 读Internet域数据报	(72)
图3-4: 发送Internet域数据报	(73)
图3-5: 用select()检查挂起的连接	(76)
图3-6: 远程注册委托方代码	(81)
图3-7: 远程注册服务方	(83)
图3-8: runtime程序的输出	(87)
图3-9: rwho服务程序	(87)
图3-10: 当接受不同波段数据时刷新终端I/O	(90)
图3-11: I/O请求异步通知的使用	(92)
图3-12: SIGCHLD信号的使用	(93)
图4-1: 面向流的应用程序的委托方侧	(104)
图4-2: TLI委托方代码	(105)
图4-3: Socket服务方代码	(107)
图4-4: TLI服务方代码	(108)
图4-5: 发送Internet域数据报	(110)
图4-6: TLI数据报代码	(112)
图5-1: 用远程过程呼叫的网络通信	(123)
图5-2: 委托方侧的RPC较低层次	(129)

图5-3: 服务方侧的RPC较低层次	(130)
图12-1: netconfig结构	(289)
图12-2: NETPATH环境变量的样本值	(290)
图12-3: 使用setnetpath()、getnetpath()和endnetpath()的样本代码	(291)
图12-4: 使用setnetconfig()、getnetconfig()和endnetconfig()的样本代码	(292)
图12-5: 使用getnetconfig()和freenetconfig()的样本代码	(293)
图12-6: 使用getnetconfig()和freenetconfig()的样本代码	(293)
图12-7: 代码例子,使用网络选择和名字到地址映射	(301)
图13-1: 服务访问设施逻辑图	(325)
图13-2: 服务访问设施目录结构	(326)
表2-1: “传输界面”的本地管理例程	(7)
表2-2: 用于建立传输连接的例程	(9)
表2-3: 连接方式数据传传例程	(9)
表2-4: 连接释放例程	(10)
表2-5: 非连接方式数据传传例程	(10)
表2-6: 描述传输界面状态转换的状态	(41)
表2-7: 发出事件	(42)
表2-8: 发来事件	(43)
表2-9: 公共的本地管理状态表	(44)
表2-10: 非连接方式状态表	(44)
表2-11: 连接方式状态表	(45)
表3-1: 运行库例程	(80)
表4-1: TLI/Socket对等项表	(115)
表4-2: Sock实现的不同之处	(117)

第一章 TLI 和 socket 编程引论

UNIX System V R 3 引入了 AT&T“传输层界面(TLI)”作为标准的与传输无关的编程界面中。在 R 4.0 中添加了“网络选择和名字到地址的映射”设施,为传输应用提供保证与媒体和协议无关的手段。“网络选择和名字到地址映射”允许程序员在与传输无关方式下获取网络应用针对具体传输的信息。

Socket 界面和 DARPA 协议(TCP/IP Internet 软件包)的支持程序,已经在 System V R 4.0 中添加了作为 UNIX System V 与伯克利 UNIX 统一的部分。。

TLI 和 Socket 提供了与传输层的编程界面。在 System V R 4.0 中,它们都在 STREAMS 框架内实现。它们的不同之处为:

- TLI与媒体和协议无关。它允许应用在任何支持TLI界面的传输协议上运行。
- Socket界面受到传统的Internet协议包TCP/IP和UDP/IP的束缚。

新的应用可以利用 TLI 协议独立性,而 Socket 界面将主要用于扩充和维护已有的基于 Socket 的应用。

1.1 本文档的组织

本文档包括本章“引论”和三个主要章节。

第二章,“传输界面编程”描述 UNIX 系统传输层界面(TLI)。

第三章,“Socket 界面”描述了基于 Socket 界面的传输层。

第四章,“Socket 迁移和 Socket 到 TLI 的转换”,描述 TLI 和 Socket 界面之间的不同,展示如何使 BSD Socket 应用适用于 System V R 4.0,并且展示无论是基于伯克利还是基于 System V 的 Socket 应用如何更改后即可在 TLI 上运行。这一章中还包括具有同等功能的 Socket 和 TLI 并行的程序示例及表格。

1.2 网络选择和名字到地址的映射

如果 TLI 应用是作为与媒体和协议无关的应用运行,那么就要求理解本版本提供的“网络选择和名字到地址的映射”设施。网络选择提供了在当前各种环境下可用的网络的标准界面。名字到地址的映射允许应用转换针对具体传输的地址。以下材料可供参考:

- 本书第十二章,“网络选择和名字到地址映射”。该章提供了这些设施的详细说明。
- 《系统管理员指南》第十章,“网络服务”。该章对“网络选择和名字到地址映射”设施的说明是针对管理员的,但没有包含库例程的完整说明。
- 本书以下手册页放在“网络选择和名字到地址映射”一章的后面: