AT&T

20307

UNIX系统V第4版
程序员指南：STREAMS

UNIX® SYSTEM V
RELEASE 4

Programmer's Guide: STREAMS

UNIX Software Operation

電子工業出版社

目　录

图和表

第一章　引论

1·1　本指南介绍

本指南向开发者提供在用户层和内核层使用 STREAMS 机制的信息。

在 UNIX System V R3 中加入了 STREAMS，这是为了扩展字符输入/输出(I/O)机制以及支持通信服务的开发。

STREAMS 向开发者提供一整套函数，一组公用程序例程以及加速软件设计和实现的设施。

1·1·1　读者

本指南主要面向网络和系统程序员，他们在 UNIX 系统通信服务的用户层和内核层使用 STREAMS 机制。

希望本指南的读者已具备有关 UNIX 系统、编程、网络和数据通信方面的知识。

1·1·2　组织

本指南分成若干章，每一章讨论一个独立的专题。第二、三和四章是介绍性的，已经熟悉 STREAMS 概念和功能的读者可以跳过这几章。

- 第一章，“引论”，描述本指南的组织和宗旨。它也规定了本指南所面向的读者以及希望他们应该具备的背景知识。
- 第二章，“STREAMS概述”，描述了STREAMS的概貌和长处。
- 第三章，“STREAMS机制”，描述了构造、使用和拆除流的基本操作。这些操作用 `open(2)`、`close(2)`、`read(2)`、`write(2)`和 `ioctl(2)`完成。
- 第四章，“STREAMS处理例程”，提供了STREAMS put和`service`例程的概貌。
- 第五章，“消息”，讨论了STREAMS消息，它们的结构、链接、排队以及与其它STREAMS组成部分的界面。
- 第六章，“轮询和信号”，描述了STREAMS怎样允许用户进程控制、管理和轮询流，以便有效地利用系统资源。
- 第七章，“模块和驱动程序概述”，描述了STREAMS模块和驱动程序环境、`ioctl`、例程、声明、刷清处理、驱动程序-内核界面，也提供了模块和驱动程序的一般性的设计准则。
- 第八章，“模块”，提供了有关模块构造和功能方面的信息。
- 第九章，“驱动程序”，讨论了STREAMS驱动程序、驱动程序流量控制、刷清处理以及增殖处理。

- 第十章，“多路复用”，描述STREAMS多路复用设施。
- 第十一章，“基于STREAMS的管道和FIFO”，提供有关基于STREAMS的管道和FIFO的创建、写入、读取、关闭以及唯一的连接方面的信息。
- 第十二章，“基于STREAMS的终端子系统”，讨论了基于STREAMS的终端和伪终端子系统。
- 附录A，“STREAMS数据结构”，汇集了STREAMS模块和驱动程序共同使用的数据结构。
- 附录B，“消息类型”，描述了STREAMS消息及其使用。
- 附录C，“STREAMS公用程序”，描述STREAMS公用程序例程和它们的用法。
- 附录D，“调试”，向开发者提供调试手段。
- 附录E，“配置”，描述如何把模块和驱动程序配置到UNIX系统中，如何调整参数以及自动压入设施。
- 附录F，“手册页”，提供了与STREAMS有关的手册页。
- 附录G，“硬件示例”，提供了用于STREAMS环境下，适合于某些硬件类型，比如AT&T 3B2的信息。
- “词汇表”定义了STREAMS独有的术语。

1.1.3 所用的约定

在本指南中，词“STREAMS”表示机制，而词“流”(stream)表示用户应用程序和驱动程序之间的通路。在基于STREAMS的管道部分，“流”表示在内核中，内核与一个或多个用户进程间的数据传输通路。

给出了一些示例，其目的是为了强调STREAMS中最重要和公共的能力。这些示例并不很完备，并且为了简单起见，引用了虚构的驱动程序和模块。

在本指南中，系统调用、STREAMS公用程序例程、前导文件以及数据结构都以等线体字给出。

变量名、指针和参数以斜体字给出。当提及示例中所独有的例程、字段和结构名时也以斜体字给出。

说明和短小的例子以等线体给出。

```
Screens are used to simulate what a user will see on a video display screen or to
show program source code.

Data structure formats are also shown in screens.
```

小心：本符号用于说明系统、应用程序、进程以及硬件的某个部分等，可能存在的损害和危险。

注意：本符号用于强调要点、提供附加的信息以及列举参考文档和命令。

1·1·4 其它文档

虽然本指南是一种有助于开发 STREAMS 应用的基本工具，但是鼓励读者从其它文档中获得更多的信息，这些信息包括 STREAMS 使用的系统调用(手册页第 2 节)和 STREAMS 公用程序(手册页 1M 节)。STREAMS 特定的输入-输出控制(`ioctl`)调用在 `streamio(7)`中提供。STREAMS 模块和驱动程序在手册页第 7 节中提供。《System V 界面定义》第三版的某些扩充条目中也对 STREAMS 作了描述。

有关 AT&T UNIX System V R4·O 详细书目的清单，请参见有关 R4·0 版的《产品概览和总索引》。

第二章　STREAMS 概述

2·1　什么是 STREAMS?

"STREAMS"是一种通用、灵活的设施和用于开发 UNIX 系统通信服务的一组工具。它在很大范围内支持多种服务的实现，从完整的网络协议包到单独一个个设备驱动程序。STREAMS 定义用于内核中的字符输入/输出，以及内核和 UNIX 系统其余部分之间的标准界面。相关的机制是简单而且开放的。它由一组系统调用、内核资源和内核例程组成。

这种标准界面和机制使得高性能的网络服务和它们的组成成分能够实现模块化，可以移植开发和容易集成。STREAMS 并不局限于任何特殊的网络体系结构。STREAMS 用户界面与 `open`、`close`、`read`、`write` 和 `ioctl` 等用户层字符 I/O 函数向上兼容。本章后面将详细地讨论 STREAMS 的长处。

"流"是在内核空间中的 STREAMS 驱动程序与用户空间中的进程之间的一种全双工处理和数据传输通路(参见图 2-1)。在内核中，流通过连接流首、驱动程序以及它们之间的零个或多个模块而构成。"流首"是流最靠近用户进程的那一端。由流上用户层进程发出的所有系统调用都由流首处理。

管道也是基于 STREAMS 的。基于 STREAMS 的管道(参见图 2-2)是内核中一种全双工(双向)的数据传输通路。它实现了内核与一个或几个用户进程间的连接，也具有基于 STREAMS 设备的优点。

"STREAMS 驱动程序"可以是提供外部 I/O 设备服务的一种设备驱动程序；也可以是一种软件驱动程序，通常这种驱动程序称为伪设备驱动程序。这种驱动程序主要处理内核与设备间的数据传输。除了 STREAMS 机制使用的数据结构与该设备理解的数据结构间的转换之外，它很少或根本不处理别的数据。

"STREAMS 模块"提供在流的数据流程上实施的处理功能。模块定义成一组用于处理数据、状态以及控制信息的内核层例程和数据结构。数据处理可能涉及修改数据表示的方式，在数据上添加或删除前导或后续信息以及/或者打包/拆包数据。状态和控制信息包括信号和输入/输出控制信息。每一个模块是自我封闭的，而且除了它的两个相邻的成分外，在功能上它与该流中所有其它成分相隔绝。模块通过传递消息与它的相邻成分通信。模块不是 STREAMS 中的必要成分，但除了基于 STREAMS 的管道(这里只需要流首)外，驱动程序是一定需要的。

在流首和驱动程序之间可以插入一个或多个模块，以便在流首和驱动程序间传递消息时对其进行中间处理。STREAMS 模块由用户进程在流中动态地互连。创建这种连接不需要内核编程、汇编或连接编辑。