

北京理工大学

本科生毕业设计（论文）外文翻译

外文原文题目: Writing Network Drivers in Rust

中文翻译题目: 使用 Rust 语言编写网卡驱动

跨操作系统的异步音频驱动模块设计与实现

Design and Implementation of Cross-Platform Asynchronous Audio and Video Driver Module

学 院: 计算机学院

专 业: 计算机科学与技术

班 级: 07112005

学生姓名: 穆新宇

学 号: 1120202695

指导教师: 陆慧梅

使用 Rust 语言编写网卡驱动

摘 要

许多开发者认为编写网络驱动程序是一项不愉快的任务。主要有三个原因导致他们不愿意这么做：在内核空间工作的困难性，大多数驱动程序的复杂性以及对 C 语言开发的普遍不情愿。

然而，现在有其他选择。内核模块不必一定用 C 语言编写，现代网卡支持越来越多的硬件委托功能，可以大大简化驱动程序的复杂度。用户空间网络驱动程序的兴起也避免了编写内核代码的需要。

我们展示了一个用 Rust 编写的先进用户空间网络驱动程序，该驱动程序注重简单性、安全性和性能，从而证明了驱动程序开发既具有挑战性又具有回报性。这个驱动程序总行数 1306 行，不安全代码占比不到 10%，它注重核心的报文处理（packet processing）功能，但在单个 3.3GHz 的 CPU 核心上，转发（forwarding）性能高达每秒超过 2600 万包，超过内核及其他一些用户空间驱动程序。

我们讨论了我们的实现细节，从不同角度对这个驱动进行评估。从结果来看，我们得出结论——Rust 是否适合编写网络驱动程序。

关键词：北京理工大学；本科生；毕业设计（外文翻译）；Rust；网卡驱动；

目 录

摘 要	I
第 1 章 介绍	1
第 2 章 Linux 上的网络通信	2
2.1 内核空间	3
结 论	4
参考文献	5
附 录	6
附录 A L ^A T _E X 环境的安装	6
附录 B B _I Thesis 使用说明	6

第 1 章 介绍

近年来，越来越多的开发者将驱动程序从内核空间移植到用户空间。与网络驱动相关的这个趋势背后的一个主要原因，就是套接字 API 的性能瓶颈问题。通用内核栈对现代需求来说简直太慢了。过去，开发者为规避这个问题而自己编写内核驱动程序。但是，内核驱动开发是一个繁琐的过程，因为在这么低级的代码层面上出现任何编程错误都可能和将导致内核崩溃。此外，内核还对开发环境和内核中可用工具施加各种限制。

幸运的是，内核驱动开发有其他选择。在第二章中，我们通过展示 Linux 中低级网络通信的不同方法来说明这些选择。我们还谈到用户空间驱动的优点之一就是可以选择任何编程语言进行实现。但是，大多数用户空间驱动仍然用 C 语言编写，这是一种早已过时的语言，如果不小心处理的话很容易导致缓冲区溢出、堆栈溢出、段错误、内存泄漏等未定义行为的问题。由于可以选择任何编程语言，那么就产生了一个问题——哪种编程语言特别适合网络驱动开发？

为回答这个问题，我们在第三章中讨论了网络驱动编程语言应该具备的优良属性，并建议选择 Rust 这一先进编程语言来替代 C 进行网络编程。Rust 能够满足我们所有理想的特性。我们通过介绍 Rust 的核心特性和一些独特的 Rust 概念来详细讨论 Rust。

第 2 章 Linux 上的网络通信

虽然互联网如今无处不在，但网络通信、网卡和网络驱动程序仍然经常被开发人员和用户视为黑箱。对于用户来说，大多数时候，他们只需要将网线插入电脑就可以上网了。对于开发人员来说，操作系统提供的高级 API 让他们可以轻松地在其程序中包含网络通信功能，而无需处理低级网络编程的复杂性。

然而，如果还没有为某种网卡提供驱动程序，或者当报文处理性能成为一个重要考虑因素时，有必要对网络通信的基础知识，尤其是应用程序与操作系统和网卡交互的方式，有初步的了解。每次从应用程序发送数据时，数据都会经过几层处理，直到到达网卡。

7	应用层
6	表示层
5	会话层
4	传输层
3	网络层
2	数据链路层
1	物理层

图 2-1 OSI 参考模型中的通信层次

一个好的模型——也是事实上的标准——是 OSI 参考模型，OSI 模型描述了这些层次（layers），它的结构如图2-1所示。

模型中的每一层都为上层提供不同的服务，如数据包分割（第 4 层）、路由（第 3 层）或可靠传输（第 2 层）。

通常，应用程序负责应用层、表示层和会话层，而传输层和网络层则由操作系统通过通用内核网络堆栈进行处理。最低的两层，物理层和数据链路层，由网络接口卡（NIC）和网络驱动程序控制。

由于 Linux 操作系统的设计，传统上，网络应用程序和网络驱动程序的不同任务在 Linux 中被用户空间和内核空间分隔开。然而，Linux 中有不同的低级数据包处理方法：大多数应用程序使用内核的套接字 API，而一些应用程序运行自己的内核模块，一些在内核空间中处理所有内容，一些在用户空间中处理所有内容。

2.1 内核空间

Linux 是基于宏内核的操作系统，这意味着整个操作系统都在内核空间中运行，而其他所有服务都在用户空间中运行。内核为用户空间应用程序提供套接字 API，即各种函数和数据类型，用于在所谓的 *Berkeley* 或 *POSIX* 套接字上提供易于使用的接口。这些套接字是特殊文件，可以通过简单的读写操作与本地进程或远程主机进行通信，遵循 Unix 的“一切都是文件”的概念。希望建立通信通道的用户空间应用程序会调用 `socket()` 函数，该函数最终会在内核中创建一个套接字结构实例，并向应用程序返回该套接字的文件描述符。随后，`bind()` 和 `connect()` 或 `listen()` 和 `accept()` 将两个或多个进程之间的连接与套接字关联起来，`send()` 和 `recv()` 则用于向套接字发送和接收数据。

为了减小通用内核堆栈的问题，一些应用实现了它们自己的内核模块。对于这种方式，两个著名的例子是 Open vSwitch^{pfaff2015design} 和 Click Modular Router^{kohler2000click}。

结 论

本文结论……。^[1]

结论作为毕业设计（论文）正文的最后部分单独排写，但不加章号。结论是对整个论文主要结果的总结。在结论中应明确指出本研究的创新点，对其应用前景和社会、经济价值等加以预测和评价，并指出今后进一步在本研究方向进行研究工作的展望与设想。结论部分的撰写应简明扼要，突出创新性。阅后删除此段。

结论正文样式与文章正文相同：宋体、小四；行距：22 磅；间距段前段后均为 0 行。阅后删除此段。

参考文献

- [1] 李成智, 李小宁, 田大山. 飞行之梦: 航空航天发展史概论[M]. 北京: 北京航空航天大学, 2004.

附 录

附录相关内容...

附录 A L^AT_EX 环境的安装

L^AT_EX 环境的安装。

附录 B B^IThesis 使用说明

B^IThesis 使用说明。

附录是毕业设计（论文）主体的补充项目，为了体现整篇文章的完整性，写入正文又可能有损于论文的条理性、逻辑性和精炼性，这些材料可以写入附录段，但对于每一篇文章并不是必须的。附录依次用大写正体英文字母 A、B、C……编序号，如附录 A、附录 B。阅后删除此段。

附录正文样式与文章正文相同：宋体、小四；行距：22 磅；间距段前段后均为 0 行。阅后删除此段。