

**本科生毕业设计(论文)**

# 开题报告

**跨操作系统的异步串口驱动模块设计与实现**

**Design and Implementation of Asynchronous Serial Port**

**Driver Module Across Operating Systems**

|  |  |
| --- | --- |
| 学 院： | 计算机学院 |
| 专 业： | 计算机科学与技术 |
| 班 级： | 07112005 |
| 学生姓名： | 林晨 |
| 学 号： | 1120202738 |
| 指导教师： | 陆慧梅 |

|  |
| --- |
| **一、选题依据**  （简述该选题的研究意义和背景，国内外研究概况和发展趋势等）  1. 研究背景和研究意义  在操作系统发展中，外设起到了非常重要的作用。为了满足不同需求而设计的外设，帮助操作系统实现了丰富的功能。从操作系统的架构上看，驱动程序是与各种外设进行直接交互的软件组件。由于外设种类多样，不同厂家对于相同外设的设计也不尽相同，这就导致了操作系统开发人员需要花大量的时间在对硬件手册的阅读以及对外设驱动程序的实现和调试上。但几乎所有的硬件外设，独立于上层的操作系统实现，因此针对于一种特定的硬件，设计并实现一个对上层操作系统独立的硬件驱动模块，供操作系统开发人员直接调用，能够帮助在其开发的操作系统中快速实现对硬件的控制。  从操作系统用户进程的角度来讲，用户进程需要的是对外设进行读写，读写的方式是通过系统调用方式。在读写系统调用执行模型的设计上，主要包括阻塞和非阻塞、同步和异步等执行模型。如果一个系统调用采用同步执行模型时，那么调用该系统调用时，在完成该系统调用的全部任务前，该系统调用都不会返回；而当一个系统调用采用异步执行模型时，那么调用该系统调用时，会立即返回，尽管该系统调用所规定的读写任务还未达成。在计算机中有些IO处理比较耗时。在调用这种同步代码时，如果进程在此处长时间等待，会严重影响程序的性能，因此在IO密集型的应用场景中，采用异步编程，能够极大程度的提升程序的运行性能。另外，采用无栈协程实现的异步，相比于传统的多线程并发设计，更是省去了操作系统对于上下文，堆栈等进行开辟和切换的开销。  Rust语言是一门适合系统编程的新兴编程语言，其拥有高于C语言的内存安全性、更现代的语法特性和等同C语言的性能。RISC-V指令集则是一套开源、简洁、模块化的指令集。因此，基于Rust语言和RISC-V指令集的操作系统是操作系统开发的新兴方向之一。这一方向上已经出现了一些教学和科研目的的操作系统，例如rCore教学操作系统[1]。 Rust 所使用的包管理工具 Cargo 为使用 Rust 语言开发的项目提供了方便的依赖包管理，能够使操作系统开发人员很容易的使用本项目开发的异步驱动模块。另外，Rust 语言也对异步和协程的相关特性有所支持，能够使我们更加自然地实现一个异步的驱动模块。  2. 国内外研究概况  异步I/O是指在进行事务处理时并发执行输入、输出以及计算操作的能力，可以将逻辑独立的I/O操作从指定的动作中分离出来单独执行[3]。简单来说，相较于阻塞的同步I/O，异步I/O是非阻塞的，无需等到I/O操作返回相应的I/O结果，就可继续执行接下来的操作。  实现异步I/O的方式有很多，包括使用操作系统提供的多线程接口，编程语言提供的函数回调、Future等语言特性编写程序。方兴等人[4]提出了一种基于WIN32的多线程异步I/O模型，用于解决复杂的多路并发I/O问题，并说明了该模型的运行机制及其优越性，这是使用操作系统支持的多线程实现同步。段楠的研究[5]使用Java提供的“消息队列”方式进行了异步的网络通讯开发，在语言层面使用语言特性对异步进行了支持。Harris T的研究[6]中更是将介绍了一组在C/C++等原生语言中用于可组合异步IO的语言结构AC，为开发人员提供了更方便的异步程序开发体验。Rust在语言层面提供了对协程和异步的支持Rust Future，使得开发人员能够方便的使用Future特性开发异步模块，实现异步外设驱动。  使用异步I/O接口，在大量I/O的应用场景下，能够极大的增加整个系统的吞吐量，使用无栈协程所实现的异步I/O相较于用多线程实现的异步I/O，更是能够避免分配大量的堆栈等内存空间，节省系统资源。沙泉 [7]在嵌入式Linux的串行通信中实现了异步事件驱动模型，在一定的硬件和实例测试的环境下说明了异步串口通信程序的设计思路和实现。Zhu L[8]、Kwon G[9]等人的研究分别在自己所设计的监测系统和实时数据归档系统中使用了异步驱动模块，在I/O量比较大的应用场景下，提升了整个系统的吞吐率和安全性。  目前，使用Rust语言编写操作系统组件逐渐成为趋势。Linux已经对其众多的硬件驱动模块进行Rust语言改造[10]，在一定程度上说明使用Rust语言所编写的更安全的操作系统模块正在逐渐被该领域的开发者所认可。在Rust语言社区中，基于Rust语言对各种硬件设备的驱动开发也正在如火如荼地进行着。Rust语言以其内存安全和并发处理能力著称，使其成为驱动开发的理想选择。越来越多的开发者加入到这一领域，编写用于嵌入式系统、物联网设备以及各种硬件组件的高效、安全的驱动程序。社区内充满了活力与合作，开发者们共享他们的经验和知识，共同推动着Rust在硬件驱动开发中的应用。通过不断的创新和实践，Rust正在成为驱动开发领域不可忽视的重要力量。  ArceOS是一个使用Rust开发的组件化操作系统[11]，它利用Rust语言的内存安全特性来确保系统的稳定性和安全性，并使用Rust语言成熟的包管理保证每个模块的高度可复用性；Rust社区也已经有许多硬件接口的驱动，包括用于与I2C和SPI总线进行通信的embedded-hal和linux-embedded-hal模块[12]，驱动HD44780字符液晶显示器的HD44780 LCD模块[13]，用于与SD卡通信的驱动程序模块[14]等。 |
| **二、研究目标和内容**  （研究目标、主要内容及关键问题等）  1. 研究目标和主要内容  本论文的主要目标是使用Rust语言中对于异步的支持，参照Embassy中异步运行时的实现，针对QEMU虚拟机模拟的串口设备，设计并开发跨操作系统的异步串口驱动模块。  毕业设计的主要工作包括以下几项：（1） 学习Rust在异步和模块化方面的相关支持；（2）学习并总结Embassy的异步运行时；（3）设计并实现异步串口驱动模块；（4）在QEMU虚拟环境下对Alien完成异步串口驱动模块的适配。  2. 关键问题  本论文工作研究的关键问题包括：在充分学习Embassy异步运行时的前提下，借鉴其异步运行时的具体设计，设计并实现异步串口驱动模块的异步运行时；将Rust异步运行时与串口的收发机制结合起来，完成异步串口驱动模块的设计与实现；在Alien中使用异步串口驱动模块等。 |
| **三、研究方案**  （拟采用的研究方法、技术路线、实验方案及可行性分析等）  1. 研究方法  本研究采用的研究方法包括：   * 文献研究法。在研究前期，充分调研相关的研究进展，了解并学习当前已有的各种硬件驱动的设计和实现、Rust语言对于异步和协程的支持、目前相关研究人员和开发人员已进行的一些尝试和工作，从中得出在使用Rust编写跨操作系统的各种异步驱动模块的设计思路。 * 定性分析法。在设计并开发异步驱动模块之后，在相关的操作系统中完成相应的正确性验证，并在理论上分析异步串口驱动模块相较于同步串口的优势之处。   2. 技术路线  本项研究的开发计划和技术路线如下：   * 首先，针对Embassy中异步运行时的设计进行学习。 * 针对 QEMU 模拟器中提供的虚拟串口开发异步驱动程序。 * 最后在Alien中验证异步串口驱动程序的正确性。   3. 可行性分析  在前期调研阶段，已经进行了初步的学习工作，包括：完成了rCore-os这一用Rust语言实现操作系统的课程实验，学习Rust对于异步和协程的支持，学习已有的同步串口驱动、异步串口驱动、异步网卡驱动等。目前对跨操作系统的异步串口驱动已有初步构思，未遇到特别大的障碍。 |
| **四、研究计划及进度安排**  2023年12月 确定研究选题  2023年12月-2024年1月下旬 调研、学习相关知识；设计异步运行时  2024年1月下旬 进行开题答辩工作  2024年2月-2024年4月 进行异步串口驱动模块的开发  2024年4月 进行中期检查工作  2024年5月 在Alien中对异步串口进行适配  2024年6月 进行毕业设计答辩 |
| **五、创新点及预期研究成果**  1. 创新点  第一，将异步机制引入外设驱动模块，在IO密集型的任务下，降低程序运行过程中的中断消耗，提升CPU的吞吐量。  第二，将对外设的操作抽象成一些与操作系统无关的接口，为操作系统开发人员提供硬件初始化、读写等接口，方便其进行操作系统开发。  第三，本研究所开发的异步驱动模块使用 Rust 语言进行开发，面向 RISC-V 指令集架构，符合时代潮流，并在一定程度上丰富该平台上的软硬件生态。  2. 预期研究成果   * QEMU中的异步串口驱动模块 * 外文翻译 * 开题报告、中期报告、毕业论文 |
| **六、参考文献**  [1] 孙卫真,刘雪松,朱威浦,等. 基于RISC-V的计算机系统综合实验设计[J]. 计算机工程与设计,2021,42(4):1159-1165. DOI:10.16208/j.issn1000-7024.2021.04.037.  [2] rCore-OS. rCore[EB/OL]. [2024-05-16]. https://github.com/rcore-os/rCore  [3] 卫一芃,杨晓宁. 嵌入式实时操作系统异步I/O技术的研究[J]. 信息通信,2017(01):141-142.  [4] 方兴,秦琦,刘维国. 多线程异步I/O模型[J]. 舰船电子对抗,2005(04):61-64.DOI:10.16426/j.cnki.jcdzdk.2005.04.014.  [5] 段楠. 异步非阻塞网络通讯技术研究[J]. 现代计算机,2019(17):79-82.  [6] Harris T. Special Topic: AC – Composable Asynchronous IO For Native Languages[C]//Conference on Object-Oriented Programming Systems, Languages, and Applications. ACM, 2011.DOI:10.1145/2048066.2048134.  [7] 沙泉. 异步事件驱动模型在嵌入式系统中的应用[J]. 微计算机信息, 2007(29):33-34+73.  [8] Zhu L , Huang L , Fu P ,et al. The upgrade to the EAST poloidal field power supply monitoring system[J]. Fusion Engineering and Design, 2021, 172(10):112757.DOI:10.1016/j.fusengdes.2021.112757.  [9] Kwon G , Lee W , Lee T ,et al. Development of a real-time data archive system for a KSTAR real-time network[J]. Fusion Engineering and Design, 2018, 127(feb.):202-206.DOI:10.1016/j.fusengdes.2018.01.019.  [10] Rust for Linux Project Developers. Rust for Linux[EB/OL]. [2024-05-16]. https://github.com/Rust-for-Linux  [11] rCore-OS. ArceOS[EB/OL]. [2024-05-16]. https://github.com/rcore-os/arceos  [12] Rust Embedded Project Developers. Embedded-hal[EB/OL]. [2024-05-16]. https://github.com/rust-embedded/embedded-hal  [13] duinoWitchery. HD44780[EB/OL]. [2024-05-16]. https://github.com/duinoWitchery/hd44780  [14] Rust Embedded Community. Embedded-sdmmc-rs[EB/OL]. [2024-05-16]. https://github.com/rust-embedded-community/embedded-sdmmc-rs  [15] Rust Project Developers. Cargo Book[EB/OL]. [2024-05-16]. https://rustwiki.org/en/cargo/index.html  [16] Bai S. Futures Explained in 200 Lines of Rust[EB/OL]. [2024-05-16]. https://stevenbai.top/rust/futures\_explained\_in\_200\_lines\_of\_rust/  [17] Rust Project Developers. Futures Crate Documentation[EB/OL]. [2024-05-16]. https://docs.rs/futures/latest/futures/  [18] Embassy Project Developers. Embassy Documentation[EB/OL]. [2024-05-16]. https://embassy.dev/dev/index.html  [19] Godones. Alien[EB/OL]. [2024-05-16]. https://github.com/Godones/Alien  [20] Jan Axelson. Serial Port Complete: COM Ports, USB Virtual COM Ports, and Ports for Embedded Systems[M]. Madison: Lakeview Research, 2000.  [21] Anonymous. UART 16550 Documentation[EB/OL]. [2024-05-16]. https://uart16550.readthedocs.io/\_/downloads/en/latest/pdf/ |
| **七、指导教师意见**  通过个人努力可实现目标，同意开题！  **签字：**  2024年2月29日 |
| **成绩：无权查看，占比：无权查看** |
| **八、开题审核负责人意见**  同意开题。  签字：  2024年3月1日 |