跨操作系统的异步驱动模块设计与实现

英文标题：

Design and Implementation of Asynchronous Driver Module Across Operating Systems

1. 选题依据
2. 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。

1. 国内外研究概况
2. 研究目标与内容
3. 研究目标和主要内容

本研究的主要目标是使用 Rust 语言针对几个特定的硬件（包括 QEMU 模拟器中的串口和网卡、Starfive2板子上的各种硬件驱动等），开发跨操作系统的异步驱动模块，并对该异步驱动模块的性能与传统的同步驱动模块的性能在一定条件下进行比较，最终得出异步驱动模块相较于同步驱动模块的优势。

本研究的主要内容包括以下几项：针对 QEMU 模拟器中提供的虚拟串口，为rCore-N，Alien，ArceOS以及Linux提供异步串口驱动；针对QEMU 模拟器中提供的虚拟网卡，为rCore-N，Alien，ArceOS以及Linux提供异步网卡驱动；针对Starfive2开发板上的各种硬件，给出相应的异步驱动实现；并设计对比实验，对以上所实现的硬件驱动的性能进行分析。

1. 关键问题

本项目的关键问题包括：对已有设备驱动的学习和总结、对 Rust有关协程和异步支持的学习、对异步驱动模块的设计和测试等。

1. 研究方案
2. 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。
* 实验研究法和定性分析法。在设计并开发异步驱动模块之后，设计对比实验，在相同测试用例，不同操作系统下比较采用本项研究设计的异步驱动和操作系统原有驱动之间的硬件操作性能；在这之后，还将采用定性分析的方式，根据异步驱动相对于同步驱动设计的不同，分析产生这种结果的原因。

1. 技术路线
2. 实验方案
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。在完成跨操作系统的异步串口驱动之后，异步网卡驱动的设计要更加复杂一些，但总体上的开发思路和开发流程与异步串口驱动的开发大致相同，预估在设计和开发过程中不会遇到太大的问题。

1. 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

2024年1月下旬 进行开题答辩工作

2024年1月下旬-2024年3月初 进行异步串口驱动的开发和测试

2024年3月初-2024年4月 进行异步网卡驱动的开发和测试

2024年4月 进行中期检查工作

2024年4月-2024年5月 进行Starfive2 板子上异步驱动的开发和测试

2024年5月 进行论文提交

2024年6月 进行毕业设计答辩

1. **创新点与预期研究成果**
2. 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

1. 预期研究成果

* 支持rCore-N、AlienOS、ArceOS以及Linux等不同操作系统的异步串口驱动模块
* 支持rCore-N、AlienOS、ArceOS以及Linux等不同操作系统的异步网卡驱动模块
* Starfive2 board 上的不同硬件资源的异步驱动模块
* 对上述异步驱动模块的性能分析结果
* 开题报告、中期报告、毕业论文

1. **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.