

**本科生毕业设计(论文)**

# 开题报告

**北京理工大学本科生毕业设计（论文）题目**

**The Subject of Undergraduate Graduation Project (Thesis) of Beijing Institute of Technology**

|  |  |
| --- | --- |
| 学 院： | 计算机学院 |
| 专 业： | 计算机科学与技术 |
| 班 级： | 07112101 |
| 学生姓名： | 戴骏翔 |
| 学 号： | 1120210610 |
| 指导教师： | 陆慧梅 |
| 校外指导教师： | 向勇 |

|  |
| --- |
| 1. **选题依据**   随着嵌入式系统和安全关键系统对可靠性和安全性的需求不断提升，微内核架构逐渐成为操作系统设计中的重要方向。在这一领域，sel4 微内核因其卓越的安全性和高效性获得了广泛关注。选题中的rel4是使用rust改写的sel4，并针对ipc和系统调用实现了异步化，现在正处于开发阶段，接下来将从sel4和rust两个方面介绍选题背景。  **1.seL4：**  seL4 是全球首个通过形式化验证的操作系统内核，其自 2004 年开始开发，并于 2014 年实现开源。它采用微内核架构，其许多特性保证了其作为嵌入式系统内核的广泛应用场景。  首先是安全性：sel4 通过比传统 linux 更为严格的能力机制实现安全高效的访问控制，从而提供了细粒度的资源管理能力。此机制使得任务能够严格遵循最小权限原则，显著降低了恶意操作和意外错误的风险。  其次是性能优化：尽管微内核设计的初衷并非追求处理速度，但sel4为了应对现实使用的需要，实现了成为处理速度最快的微内核。  最后是形式化验证：sel4 是首个通过数学形式化方法验证正确性的操作系统内核。这一验证保证了理论上内核没有bug，确保其具备严格的可靠性。其形式化验证是安全性的重要保障  以上这些特点，使其广泛应用于军事、航空航天、铁路信号和医疗设备等拥有高安全需求的领域。  **2.Rust 语言**  rust 语言因内存安全成为近年来操作系统编程的可靠工具，其无需依赖垃圾回收等机制和方便的编译器环境适合作为系统编程语言。而其安全性与sel4匹配，适合开发对安全性和性能要求苛刻的系统组件。 |
| **二、研究目标和内容**  研究的目标是基于 Rel4 微内核，结合异步系统调用和异步IPC的机制，设计并实现一个高性能的异步网络协议栈，使其具备 Socket API 兼容性，适应嵌入式系统的需求，保证高性能。  主要内容：已经确定了使用smoltcp作为参考的开源网络协议栈进行设计，但是其为了适应rel4的环境并不能直接使用，需要做的就是结合rel4的ipc异步化去实现一个异步的网络协议栈，其中smoltcp作为一个轻量化的tcp模型，比较适合rel4嵌入式的特征，但是因为其并没有实现很多的组件，所以例如socket通信等需要进行补充，且经过老师指点，smoltcp作为一个异步网络框架，其异步化并不是很彻底，需要进行修改。 |
| **三、研究方案**  1.配置实验环境，首先配置Linux 双系统以保证开发与测试的效率，在此基础上使用docker配置rel4环境，使在开发过程中与设计rel4异步化的学长保持一致，避免环境差异带来的问题。  2.学习sel4的具体知识，并阅读rel4代码，因为rel4 的异步通信机制将在网络协议栈的设计中其很大作用，所以要分析学习异步化后 Rel4 微内核中的ipc机制，明白如何利用 Rel4 的异步 IPC 来实现协议栈的任务调度与数据传递，    3. 学习Rust异步编程，在前面rust知识的基础上，主要学习 Rust 异步框架的设计和实现细节，如Rust 提供的 async/await 等语法，并在此过程中复习rust在操作系统开发中的代码组织，确保代码的简洁可读性，同时保持代码的安全和稳定性。    4.学习与分析 smoltcp 协议栈，这一部分是主要内容，首先理解其实现，即smoltcp 作为适用于嵌入式系统的轻量级 TCP/IP 协议栈，通过分析其代码，理解其如何在嵌入式设备上实现高效的网络通信。之后重点关注 smoltcp 中的异步 I/O 的机制及其内存管理策略，结合论文以及网上资料学习异步编程的知识，将知识点与smoltcp中的实际代码结合，分析出其实现异步化的代码部分，以及那些可以进行修改，供后面拓展。  5.完成smoltcp本身的了解后，在其基础上，完成适配 Rel4 微内核的异步架构。其中分为smoltcp本身的异步化增强和如socker通信等机制的补充，包括对于rel4的适配，即把 Rel4 的异步 IPC 机制引入到协议栈的数据传输与连接管理中，实现基于消息传递的异步网络协议栈。  6.研究首先在qemu上进行，可以初步先跑通项目，再进一步增强性能，通过实际的发包测试与之前的结果比较得到具体的数据，便于进行后续的结果展示以及确认异步化的具体效果 |
| **四、研究计划及进度安排**  1．学习并掌握Rust异步编程基础（1-2周）。 .  2．学习ReL4和seL4的相关论⽂、背景知识和熟悉代码。（3-4周）。 .  3．设计⼀套满⾜设计要求的异步⽹络协议栈（5-6周）。  4．编码和调试（7-9周） .  5．进⾏性能测试并分析数据。（11-13周）  6．完成本科⽣毕业设计（论⽂）外⽂翻译。（第 1 周-第 7 周） .  7．完成毕业论⽂，提交软件及相关⽂档。（第 13 周-第 14 周） .  8．完成本科⽣毕业设计（论⽂）答辩。（第 15 周） |
| **五、创新点及预期研究成果**  **Rel4 的开发愿景**  Rel4 项目结合了 seL4 的微内核特性和 Rust 语言的安全性，旨在构建一个更可靠的微内核系统。其开发目标包括：   1. **提升内核代码的可维护性**：Rust 语言的丰富类型系统和内建的错误处理机制有助于减少常见的编程错误。 2. **增强内核的安全保障**：通过 Rust 的内存和并发安全特性，进一步降低低级内存管理中的安全隐患。 3. **推动微内核生态的发展**：Rel4 的成功开发将为未来更多的微内核项目提供参考，实现更安全和高效的操作系统内核。   综上所述，Rel4 项目的发展将 Rust 语言的现代编程优势与 seL4 微内核的形式化验证能力相结合，为嵌入式系统和安全关键领域的操作系统设计树立了新的标杆。 |
| **六、参考文献**  [1] Klein G, Elphinstone K, Heiser G, et al. seL4: Formal verification of an OS kerne[C]//Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles. 2009: 207-220.  [2] Heiser G, Elphinstone K. L4 microkernels: The lessons from 20 years of research and deployment[J]. ACM Transactions on Computer Systems (TOCS), 2016, 34(1): 1-29.  [3] Klimiankou Y. Micro-CLK: returning to the asynchronicity with communication-less microkernel[C]//Proceedings of the 12th ACM SIGOPS Asia-Pacific Workshop on Systems. 2021: 106-114.  [4] Heiser G. The seL4 Microkernel–An Introduction[J]. The seL4 Foundation, 2020, 1 |
| **七、指导教师意见**  **签字：**  **年 月 日** |
| **成绩： ，占比： %** |
| **八、开题审核负责人意见**  签字：  年 月 日 |



**本科生毕业设计(论文)**