sv6 for RV64 SMP - 课程设计中期报告

计53 谭闻德 尹宇峰

选题概述

近年来，人工智能及大数据的众多应用为计算能力提出了新的需求，而处理器核心频率的提高已经遇到困难，现如今，多核心处理器、并行化已成为提高计算系统性能的主要手段。事实上，在2013年，Austin T. Clements等人[1, 2]就提出过一种新的手段，用于优化并行系统软件的性能。此项工作提出了一个名为commuter的工具，借助符号化执行技术[10, 11]以及SMT求解器[12]，用于挖掘系统规范中潜在的并行化优化点；同时，他们用此工具辅助设计实现了一个并行度很好的精简的操作系统，名为sv6。sv6在x86-64指令集架构上实现，考虑到目前新兴的RISC-V指令集架构[3, 4]更开放、灵活性更好、硬件设计更简洁，为此，本课程设计计划基于上述工作，将上述工作提出的sv6移植到RISC-V 64位多核平台，以期能够推动RISC-V并行架构的发展。最后，本课程设计还计划利用commuter工具进一步优化sv6的并行性。

由此，本课程设计的主体工作计划分成以下两个部分：

1. **移植** 将sv6操作系统从x86-64移植到RISC-V 64位（以下简称RV64）对称多处理器（以下简称SMP）环境。
2. **优化** 利用commuter工具进一步优化sv6操作系统的一部分，例如网络子模块。

预期收获

- 熟悉RV64指令集
- 熟悉一个操作系统的移植过程
- 学习符号化执行方法
- 学习Z3 SMT求解器的使用方法
- 了解sv6以及commuter
- 深入理解socket

相关工作

这一小节简要介绍本课程设计会用到的前人的一些已有工作。RISC-V [3, 4]是一个新的指令集架构，它更开放、灵活性更好、硬件设计更简洁。sv6操作系统[1]则是一款基于xv6，并行度很好的精简的类POSIX的研究性质的操作系统。通过符号化执行的方法[10, 11]，可以将一个函数的输入参数指定为符号值，函数的每一个路径可以产生一个先决条件逻辑表达式，通过Z3等SMT求解器[12]进行求解，就可以知道这个路径是否可达，是否会产生特定的情况（大多数情况是断言失败）。Austin T. Clements等人在2013年提出的Commuter [2]能够利用上面提到的符号化执行技术以及Z3求解器，完成对系统调用接口规范模型的分析，为设计良好并行化的接口给出建设性意见。

设计方案

本课程设计计划分为以下9个步骤分阶段进行：

准备开发环境 — 已完成

把RV64 SMP的工具链，包括编译器GCC、模拟器QEMU等环境准备好。

这是基础性的工作，为进行RV64的开发，必不可少。

学习RV64 — 已完成

学习RV64用户态指令集架构和特权指令集架构。

这也是基础性的工作，为进行RV64的开发，必不可少。当然，若之后遇到问题，可以进一步查阅文档。

动手实践RV64 SMP — 已完成

基于bbl编写几个RV64 SMP的小例子，注重SMP的启动以及管理。这个环节旨在熟悉RV64 SMP的一些关键性问题，例如谁负责启动AP（除了启动处理器之外的处理器核心称为AP），启动AP后的状态如何设置等。

开始移植 — 正在进行中， 预计第10周周末完成

开始将sv6移植到RV64单核环境。

这个阶段的重点以及难点都在于将原有的x86-64相关的部分在RV64架构下重新实现，包括虚拟内存管理、中断和异常（包括系统调用）、原子操作指令、寄存器上下文、内核栈设置、外设中断配置、设备驱动程序等。

多核移植 — 预计第10周周末完成

进一步将sv6移植到RV64多核系统。

注：根据实际情况，单核和多核的移植可以分开或者同时进行。

完善驱动程序 — 预计第10周周末完成

由于RV64与x86-64架构有所不同，外设差异也相当大，驱动程序无法兼容。为此，需要针对RV64的情况，自行实现或者从已有的RV64操作系统（如Linux）移植驱动程序，特别是磁盘驱动程序以及网卡驱动程序。磁盘起到信息持久化存储的作用，而网络起到与外界交换数据的作用。如果使用QEMU进行模拟，则需要实现VirtIO驱动程序；若要运行在真实硬件上，则需要根据硬件说明文档编写或移植相应的驱动程序。

进行性能测试 — 预计第10周周末完成

根据sv6论文提出的方法，尝试用sv6与Linux进行性能测试对比。为保证结果真实性，需要使用硬件RV64。HiFive Unleashed是一个全新的RV64 SMP开发板，配置四核RV64处理器，最高主频1.5GHz，配有1Gbps网卡和其他多种实用外设，文献[14]指出其特权指令集架构版本为1.10。此开发板售价为999美元，现已隆重发售。

学习并实践符号化执行方法、Z3求解器 — 预计第10周周末完成

学习符号化执行方法以及Z3求解器的使用。

commuter工具使用到了符号化执行方法以及SMT求解器，为了更好地完成sv6的优化，更好地写出接口规范，需要学习这两项内容。

优化sv6 — 预计第12周周末完成

利用commuter工具进一步优化sv6操作系统的一部分，例如网络子模块。

进一步可以分为接口规范的编写，实现代码的调优等步骤。

小组分工

sv6移植RV64 SMP

谭闻德主导，尹宇峰辅助。

commuter优化sv6系统子模块

尹宇峰主导，谭闻德辅助。

已完成工作

1. 构建完成了RV64的工具链。
2. 在学习RV64指令集的同时，帮助修复ucore labs RV64移植的2个bug。
3. 编写了RV64 SMP操作系统的例子，包括VirtIO驱动的简单例子。
4. 修复了x86-64平台上原始的sv6，使其可以正常编译及运行。目前不确定现在的修复方案是否有潜在风险。
5. 开始了sv6的移植，目前遇到问题（请见下文）。
6. 阅读符号化执行相关文献，初步了解符号化执行工作原理。
7. 阅读commuter相关文献，初步了解commuter工作原理。
8. 搭建并重现commuter project，生成了测试用例。
9. 配置完sv6运行环境，使用sv6成功运行commuter生成的测试用例。

潜在问题分析

sv6移植RV64 SMP

1. （已遇到）sv6大部分代码是C++语言写的，而C++的运行时环境要求比C语言苛刻。目前在移植sv6时，遇到了缺少libgcc_eh.a库而无法链接的情况。
2. RV64没有TSS，需要思考和调研如何保存内核栈基址。
3. Supervisor态无法访问mhartid，这个寄存器存放了处理器核的编号，而为了支持多核，需要让操作系统可以访问当前处理器核的编号，从而能够访问处理器特定的数据结构，为此需要将处理器编号存入别处。

commuter优化sv6系统子模块

1. 如果我们最终选择sv6的网络子模块优化，我们需要明确并分析socket一类系统调用的规范。
2. commuter相关文献中的实验部分，使用的服务器是80核处理器，而我们并没有如此发达的计算资源，可能需要教学团队给予一定帮助。

3. 配置commuter运行环境成功后，因为安装的库不兼容的原因，导致了我们的虚拟机图形界面崩溃了。

参考文献

1. sv6 https://github.com/aclements/sv6
2. Commuter https://pdos.csail.mit.edu/archive/commuter/ https://github.com/aclements/commuter
3. The RISC-V Instruction Set Manual Volume I: User-Level ISA

   https://riscv.org/specifications/
4. The RISC-V Instruction Set Manual Volume II: Privileged Architecture

   https://riscv.org/specifications/privileged-isa/
5. bbl-ucore https://ring00.github.io/bbl-ucore/
6. ucore labs RISC-V 64移植 https://gitee.com/shzhxh/ucore_os_lab/tree/riscv64-priv-1.10
7. Booting a RISC-V Linux Kernel https://www.sifive.com/blog/2017/10/09/all-aboard-part-6-booting-a-risc-v-linux-kernel/
8. Entering and Exiting the Linux Kernel on RISC-V https://www.sifive.com/blog/2017/10/23/all-aboard-part-7-entering-and-exiting-the-linux-kernel-on-risc-v/
9. Paging and the MMU in the RISC-V Linux Kernel https://www.sifive.com/blog/2017/12/11/all-aboard-part-9-paging-and-mmu-in-risc-v-linux-kernel/
10. 符号执行入门 https://zhuanlan.zhihu.com/p/26927127
11. mini-mc https://github.com/xiw/mini-mc
12. Z3 https://github.com/Z3Prover/z3
13. HiFive Unleashed https://www.sifive.com/products/hifive-unleashed/
14. Freedom U540-C000 Manual https://www.sifive.com/documentation/chips/freedom-u540-c000-manual/