

初赛阶段性报告(第二版)

引言

项目描述：

对南京大学的PA实验系统进行扩展，添加LoongArch架构支持。实验分成4个阶段：PA1-PA4

项目目标：

1. 通过PA实验构建一个简单完整的计算机系统, 来深入理解程序如何在计算机上运行。
2. 通过该项目练习如何进行团队协同开发与项目管理。
3. 撰写详细的实验手册，填补PA实验讲义对于loongarch架构相关的介绍空白，让往后的同学做pa实验选择LA32r作为目标架构时,上手难度能与 MIPS、RISC-V 持平。

实现部分

个人工作内容

PA1

1. 简易调试器sdb功能: 内存扫描
2. 简易调试器sdb功能: 监视点

PA2

1. LA32r反汇编器的完善(基础架构是师兄编写，我做后续补充)
2. 添加基础指令集
3. dtrace
4. 外设支持(时钟，键盘，VGA)
5. 简易标注库实现

PA3

1. Loader实现
2. 系统调用支持(yeild, exit)

挑战与解决方案

1. 在项目的初期阶段PA1，主要任务集中在构建sdb简易调试器的内存扫描与监视点功能。面临的主要挑战包括如何安全访问客户计算机内存数据、如何解析字符串以提取关键信息，以及如何进行字符的进制转换。通过深入研究源代码、查阅在线资源，这些问题逐一得到解决。此外，还成功掌握了Git命令，实现了本地文件与远程仓库的同步协作，进一步提升了团队的开发效率。
2. 进入PA2阶段，工作重心转向LA32r反汇编器的完善与基础指令集的添加，同时涉及dtrace工具的运用、外设支持以及简易标注库的实现。在此期间，面临的难点在于理解反汇编器的指令获取与翻译机制，以及如何区分具有相同操作码的不同指令。通过细致研读龙芯架构手册和团队的深入讨论，这些问题迎刃而解。针对外设支持，特别是VGA显示的优化，通过查阅PA文档及相关资料，成功实现了流畅的渐变图像输出，深化了对外设初始化及配置流程的认知。
3. PA3阶段，承担了Loader实现与系统调用支持的任务，特别是yeild与exit指令的实现。此阶段的关键挑战在于确定正确的内存加载位置与系统调用号的识别及处理。通过对ELF文件格式的学习和团队的密切讨论，成功解决了Loader的实施问题。至于系统调用，通过深入交流想法与文档研究，清晰梳理了系统调用的处理逻辑。

学习总结

1. 从项目的初始阶段PA1到PA3，我的参与和成长经历了一条从迷惘到明悟的路径。PA1阶段，面对的是基础设施的初步搭建，我在编码上的随意与态度的不严谨，导致了代码逻辑混乱且缺乏认真对待的态度。队长的批评如同当头棒喝，让我意识到项目不应仅是应付了事，而需倾注心力去理解与构建。因此，我开始痛定思痛，决心改变自己的学生思维模式，沉下心来，不仅学习如何独立解决问题，还期望能在后续中承担更多的主线开发任务，以此锻炼自己。
2. 进入PA2阶段，我承担了更多的开发责任，这一时期，我不仅对LoongArch架构有了深入的了解，还在实践中逐渐熟悉了nemu的内部运作。编写模块文档的过程，让我有机会深入剖析底层代码，虽然发现自己的代码缺乏面向工程的特性，但这也成为了我后续改进的方向。我期待着能够逐步揭开nemu的神秘面纱，使其从一个“黑盒”变成一个我可以熟练驾驭的工具，并且努力让我的代码更加模块化，增强其解耦性。
3. 抵达PA3阶段，我对nemu、AM运行时环境及nanos-lite简易操作系统的相互关联有了深刻的认识，同时也掌握了Makefile的基本用法。我对于计算机体系结构的理解日益加深，对于编写高效、可读性强的代码有了更明确的目标。尽管在系统调用方面仍感困惑，但我已明确这是接下来需要攻克的重点，希望通过学习和实践，最终实现剩余系统调用的支持。

未来的展望

总结而言，从PA1的自我觉醒，到PA2的技术深化，再到PA3的体系认知提升，我在这段旅程中不仅技术能力得到了显著提升，更重要的是，我学会了如何以更成熟的态度面对挑战，如何在团队合作中发挥自己的作用，以及如何持续地自我驱动，追求技术的精进与项目的成功。展望未来，我将持续探索项目代码的每一个角落，深化对整个执行流程的理解，同时致力于提升代码的工程性和解耦性，为项目的最终成功贡献自己的力量。

