

cachelab 实验说明 2022

实验内容

本实验分为 A/B 两部分, Part B 依赖于 Part A.

Part A. 用 C 语言实现一个 cache simulator

Part B. 优化矩阵转置算法减少其 cache miss, 并利用 Part A 中实现的算法计算该算法的 cache miss 次数.

完成后请提交 `<SID>-handin.tar` 文件和你的 PDF 格式实验报告两个文件. 实验报告中请描述 Part A 的实现架构设计与 Part B 的优化思路.

实验步骤

获取实验文档

OBE平台上的压缩包中有 3 个文档, 分别是本文档/ `cachelab.pdf` / `cachelab-ppt.pdf`.

`cachelab.pdf` 是本次实验的**第一手参考资料**, 其中包含框架代码与测试接口的详细描述以及自行测试的文档, 以及编写代码时必须遵守的规则. **该文档包含你需要知道的关于本实验的所有信息.**

务必仔细阅读 `cachelab.pdf`

`cachelab-ppt.pdf` 为布置作业时使用的课件, 包含一些对 cache 的简介和实现实验简要思路.

获取实验框架

实验框架位于 `ics.ruc.rvalue.moe` 服务器的 `/mnt/ics1-2020/cachelab-handout.tar`. 请自行到该路径读取并解包使用. 方法与先前实验一致在此不再赘述.

Part A

Part A 部分需要你在 `csim.c` 文件中补全 cache simulator 的实现.

具体 I/O 格式以及测试数据的生成见 `cachelab.pdf`.

务必仔细阅读 `cachelab.pdf`

补全后在实验框架目录下执行 `make` 命令即可编译生成可执行程序 `test-csim`. 执行命令 `./test-csim` 即可测试 Part A 的正确性.

Part B

Part B 部分需要你优化 `trans.c` 文件中的矩阵转置算法.

优化之后你可以执行 `make` 命令编译生成可执行程序 `test-trans`, 执行命令 `./test-trans -M <m> -N <n>` 即可测试 `trans.c` 中各个矩阵转置实现在 m 行 n 列矩阵的情况下的性能.

最终测试会使用 `./test-trans -M 32 -N 32` / `./test-trans -M 64 -N 64` / `./test-trans -M 61 -N 67` 三种情况进行测试.

务必仔细阅读 `cachelab.pdf`

最终总评

`make` 之后执行 `./driver.py` 即可测试你在该实验的总体得分. 该测试脚本满分为 53 分, 不代表你在本实验的最终得分.

`make` 之后会在你的实验框架目录下生成一个 `<SID>-handin.tar` 文件. 请将其与你的实验报告一同提交至OBE.