

日志任务（2.4 和 2.5）—2021 春

step1:

【10%】实验 valgrind 安装。

【5%】学会 PartA 部分参考引用 csim-ref 模拟器使用命令行参数及使用方法。

【20%】认识 yi.trace 中各项(I,L,S,M)操作含义，并运行在缓存模型（参数-v -s 4 -E 1 -b 4）上，能逐条分析 trace 文件操作且对比 LRU（最近最少使用的）替换策略，能分析 hit、miss、eviction 结果产生过程。

step2:

【10%】检验主函数代码编写和结果分析。在 csim.c 基础上，进一步编写主函数：能解析输入命令参数且输出自定义缓存模型结构（如直接映射型）；put_Sets()函数显示各组信息；能在错误输入命令行参数时输出帮忙信息。

注意：本步骤代码标注注释/*step2 用：*/

step3:

【15%】检验 LRU 主函数代码编写与结果分析。在之前 csim.c 基础上，进一步编写主函数：假设 cache 模型（如 s1E2b2）；判定不命中时调用更新 cache 的 Lru 运行脚本函数（参数为缓存指针、组号、标记位）；编写 put_Sets()函数逐行显示包含标记位 tag 的各组信息；编写一段 Lru 运行脚本能通过输出信息检验出正常 LRU 算法行为。

注意：本步骤代码标注注释/*step3 用：*/

step4:

【10%】编写加载数据的 L 命令处理函数代码。注意 getopt 详细模式(-v 选项)时，需要打印出相应结果 miss 或 hit 或 eviction。

【5%】编写存储数据的 S 命令处理函数和修改数据的 M 命令处理函数。

【10%】编写获取 trace 脚本操作地址中的组索引与标记位的函数。

【15%】检验 LRU 主函数代码编写与结果分析。在之前 csim.c 基础上，进一步编写主函数：用 fscanff()函数读取 trace 文件并分支处理 LSM 命令；最后要调用 printSummary()输出 hit、miss 和 eviction 各项总数以及释放内存。make 后比较 csim 与 csim-ref 在缓存 s4E1b4 模型上运行 yi.trace 脚本结果。

注意：本步骤代码标注注释/*step4 用：*/