

**计算机系统结构实验报告**

|  |  |
| --- | --- |
| 姓 名： | 聂鸿勇 |
| 学 院： | 计算机科学与技术 |
| 专 业： | 计算机科学与技术 |
| 班 级： | CS1707 |
| 学 号： | U201714785 |
|  |  |

|  |  |
| --- | --- |
| 分数 |  |
| 教师签名 |  |

2020 年. 5月. 1日

**目 录**

[1. Cache模拟器实验 3](#_Toc23963)

[1.1. 实验目的 3](#_Toc3705)

[1.2. 实验环境 3](#_Toc865)

[1.3. 实验思路 3](#_Toc25045)

[1.4. 实验结果和分析 3](#_Toc21474)

[2. 总结和体会 4](#_Toc18091)

[3. 对实验课程的建议 4](#_Toc32753)

# Cache模拟器实验

* 1. **实验目的**

（1）理解cache工作原理；

（2）如何实现一个高效的模拟器。

* 1. **实验环境**

操作系统：Ubuntu LTS 16.04.11

编译器：gcc 5.4.0

IDE：Notepad++

* 1. **实验思路**

(1)、内存地址采用组相联映射，实验过程中只是为了简单模拟LRU算法的实现，所以将其数据结构改为下图所示：valid标记当前行是否有效，tag代表当前行所映射的内存地址，lru用来辅助比较帮助替换块时做出抉择。

深度截图_选择区域_20190426005533

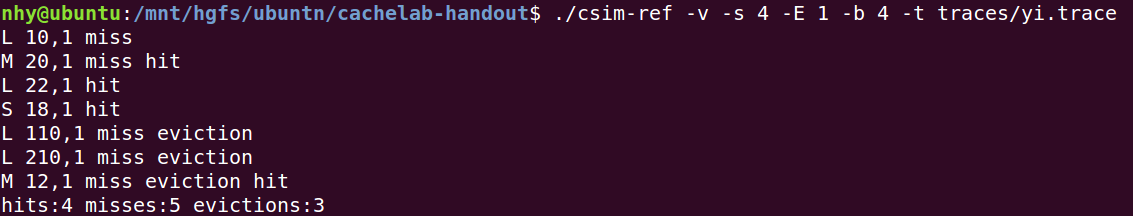
图1-1

(2)、Cache替换采用LRU算法，LRU即最近最不经常使用(least recently used)。检查cache某一组内所有cacheline的tag是否与输入的tag相等，如果相等且cacheline的valid位为1则表示cache访问命中，此时LRU计数器清零，若不想相等，将所有的的cacheline，LRU计数器加1；如果cache不命中，则需要把主存块调入cache中，先去找cache空闲即valid = 0的那些cacheline，找到后，直接将当前访问的地址tag存储进cache，并将当前cacheline的valid置为1；如果没有空闲的cacheline，则此时需要对cache进行替换，替换策略是遍历所有cacheline将lru计数器最大的cacheline替换掉。

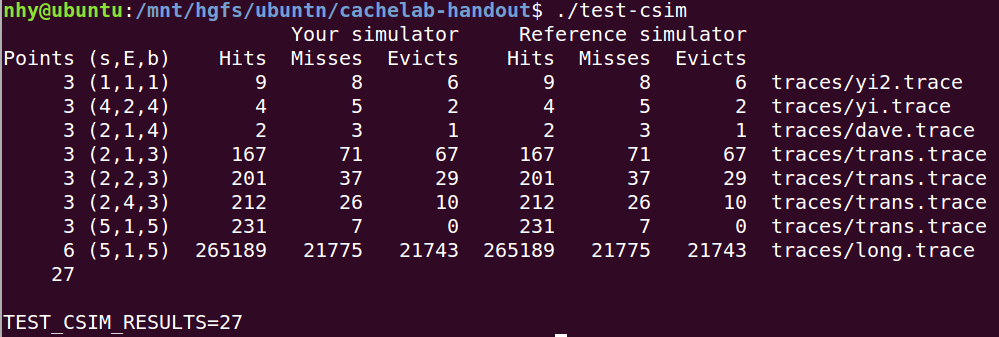
(3)、本实验中还需要编写代码对输入的指令进行相应的处理，对于Load/Store指令，只需要访问主存一次，因此只需要调用一次accessData；忽略所有指令cache访问；Modify指令需要访问两次主存，因此需要调用两次accessData。

* 1. **实验结果和分析**

（1）单例./csim(-ref) -v -s 4 -E 1 -b 4 -t traces/yi.trace运行结果如下：



（2）全部样例./test-csim运行结果如下：



# 总结和体会

在之前的组成原理课程当中，我们以及对cache有了初步的了解，也对cache的各种替换算法进行了学习，在组成原理的实验当中也使用logisim实现过cache的电路图，因此，在本实验中，只需要使用代码实现相应的功能便可以完成相应的任务要求，整体来说比较容易实现。

# 对实验课程的建议

（1）、计算机系统结构的课程内容与组成原理的课程内容有较大部分的重合，因此，建议可以在本课程的讲解当中更加深入、详细。

（2）、实验选自CSAPP(深入理解计算机系统)第三次cache实验，但截取的部分比较简单，可以加大一点难度。