实验报告

实验名称：**缓存模拟**
实验时间：2018 年 5 月 2 日
实验人员：刘悦（姓名）11510424（学号）大三（年级）
实验目的：掌握缓存机制的过程，深入理解缓存的设计理念和不同策略之间的优劣
实验环境：Linux
实验步骤：
1.了解不同缓存的细节与思想
2.编写模拟程序
实验陈述：
1、基础知识回顾：
1.direct mapped cache 的内容：主存的一个某块只能映像到 Cache 的一个准确确定的块中,但是缓存块可以有多个主存块与之对应。该方法成本虽低，但命中率低，效率较低。

2.set associate cache 的内容：可以将一个主存块存储到唯一的一个 Cache 组中任意一个行。结合了 direct mapped cache 和 fully associate cache 俩者的优势，cache 被分为了若干 set，每个 set 里面有若干个 set line，一个主存块映射到 set 中，在这个 set 中，可以任意放置。

3.fully associate cache 的内容：可以将一个主存块存储到任意一个 Cache 行。但最大的问题就是当寻找一个地址是否已经被 cache 时，需要遍历每一个 cache line 来寻找，这个代价很高。

4.TLB 原理以及和其他 cache 的不同之处：TLB 的全称是 Translation Lookaside Buffer，是一个虚拟寻址的 cache，存储了虚拟地址到物理地址的转换表。当处理 器要在主内存寻址时，不是直接在内存的物理地址里查找的，而是通过一组虚拟地址转换到主内存的物理地址，TLB 就是负责将虚拟内存地址翻译成实际的物理内 存地址，而 CPU 寻址时会优先在 TLB 中进行寻址。区别在于 TLB 靠操作系统进行维护，而其他的 cache 靠 CPU 维护，而且其他 cache 中存放的是数据，而 TLB 存放物理地址转换

2、缓存模拟
模拟程序使用的缓存策略是？direct mapped cache
请写出地址、index、byteselect 大小的计算过程 内存块大小是 256Mbytes(28 位),cache 大小是 1kbytes(10 位)，而 cache line 大小为 16 bytes(4 位)，所以一共有 64 个 cache line（6 位），所以 index 为 6 位，byteselect 为 4 位

采用的策略有哪些弊端？ 替换操作频繁，命中率比较低

为什么我们不把 cache 设计的大一些？ 因为我们在设计时必须考虑成本问题，而且 cache 增大查找速度也会降低

3、 遇到的问题与解决方法

问题 1：这次实验主要是对题意的理解花了我很多的时间,刚开始对 direct mapped cache 去寻址思路没有明白

解决方法 第一是看 lab 的说明文档，第二时看网上的教程一步一步看寻址的步骤

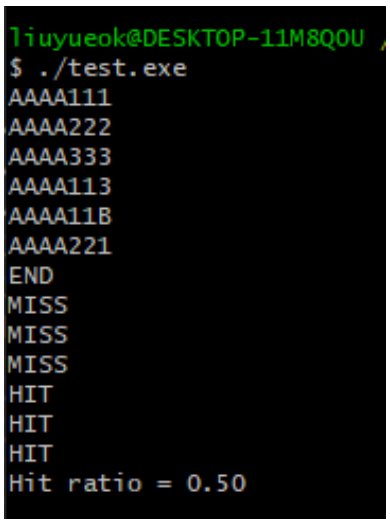
问题 2： 十六进制的获取，不知道怎么实现

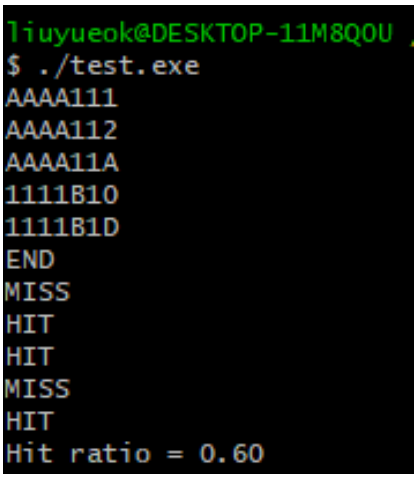
解决方法 解决这个问题，我首先的想法时 scanf（%x,）去读取十六进制数，但由于需要辨别“EDD”，而且在测试过程我发现输入“A23Z”也会识别，就没有采用这种方法，后来是先读取字符串，再将字符串转化为十六进制数

问题 3： 始终未正确输出 HIT 和 MISS

解决方法 发现判断条件写错了，这样写，if (t==1)，一直判断未 true

实验截图：





从实验截图中，我们可以看到结果是正确的。比如图一，AAAA111，AAAA222 和 AAAA333 是冷启动，都 miss。

实验总结：

实验真的不难，但我理论去理解花了我很多的时间，还有就是在 c 语言中对于字符串的操作也特别要注意 。