



北京交通大学  
BEIJING JIAOTONG UNIVERSITY

# 《计算机体系结构》 实验报告

实验名称:	实验 6_cache 模拟实验报告
学 号:	
姓 名:	
学 院:	计算机与信息技术学院
日 期:	2022 年 12 月 06 日

## 目录

1. 运行 cache128.exe 程序.....	3
2. cache 运行情况跟踪.....	3

# 1. 运行 cache128.exe 程序

在 C++ 环境下（比如，可采用这里提供的 Dev C++）运行 cache 系统模拟程序 cache2018.c，也可在命令行环境下直接运行可执行文件 cache2018.exe。

介绍：cache2018.c 是我编的一个小程序，用来展示 4-路组相连 cache 的运行结果。可参见书 4-35 页上的图 4.19。为便于实验，采用了与书图上不同的数字：

Tag 位数为 3，而不是 22。

Index 位数采用了 3，而不是 8。因此组数为  $2^3=8$ 。

字节偏移位数为 2。一个块的大小是一个字，即 4 个字节。

总的地址空间的位数为 8，而不是书图中的 32 位。因此，存储器中总的块数为  $2^{(8-2)}=64$ 。Cache 中的总的块数为：组数 $\times$ 相联数 $=8\times4=32$ ，是存储器中总的块数的一半。

运行后，程序随机产生出 40 次访问的存储器块号（在 0-63 之间），并计算命中次数、不中次数、以及替换次数。程序采用的是最近最少使用(LRU)策略。

直接命令行运行程序：

```
D:\BJTU\subject\Junior First\Computer Architecture\lab6\实验6>cache2018.cpp
D:\BJTU\subject\Junior First\Computer Architecture\lab6\实验6>cache2018.exe

The number of address bits is 8.
We assume 1-word block, so the number of block bits 8-2=6.
So there are 64 blocks in memory.

We assume there are 32 blocks in cache, half of that in memory.
We adopt 4-way associative cache, so the number of sets is 8.
We create 40 blocks numbers to access memory, and calculate the number of hits, misses and replacements.
Theses accesses are:

    1   19   13   58   23    8   11   47   39    3
   21   29   56    0   10   57   57    5   40   20
    1   22   53   51   27   41   29   27    2   25
   28   43   41   30   30   47   63   61   37   49

The result is as following ...
The number of accesses is 40, in which
  the number of Hits is 7,
  the number of Misses is 33,
  the number of replacements is 6.

Now it is your task to demonstrate how these results are obtained.
Good luck!

D:\BJTU\subject\Junior First\Computer Architecture\lab6\实验6>
```

图表 1-1 cache128.exe 程序运行结果

稍后会在标题 2 中进行分析。

## 2. cache 运行情况跟踪

请根据软件随机产生的 40 个连续访问的块号，一步一步地跟踪 cache 系统的执行过程，

记录各个地址来了以后，cache 的变化情况（对于每个 miss，要额外标注是哪种 miss（compulsory, conflict, capacity））。关于三种 miss 的定义，可参看第 4 章课件的第 156 页。

最后，核实一下你的最终结果是否与程序给出的结果相同。

根据课件上的分类，所有的缺失分为三种，分别是强制缺失 compulsory miss、容量缺失 capacity miss、冲突缺失 conflict miss。

下边开始对产生的 40 个连续访问块号进行分析。而分析结果由表格形式给出。

首先是一开始的强制缺失，见表格 2-1。

组号	块号0	块号1	块号2	块号3
0	8	56	0	
1	1	57		
2	58	10		
3	19	11	3	29
4				
5	13	21		
6				
7	23	47	39	

表格 2-1 程序一开始的强制缺失

这里一共 16 个强制缺失。

接下来下一个是访问 57，命中。

组号	块号0	块号1	块号2	块号3
0	8	56	0	40
1	1	57		
2	58	10		
3	19	11	3	29
4	20			
5	13	21	5	
6				
7	23	47	39	

表格 2-2

下边有 3 个强制缺失，分别是访问了 5、40、20。之后的访问 1 命中。对于之前命中的我们标记为红色。

组号	块号0	块号1	块号2	块号3
0	8	56	0	40
1	1	57		
2	58	10		
3	19	11	3	29
4	20			
5	13	21	5	53
6	22			
7	23	47	39	

表格 2-3

后边有两个强制缺失，分别是访问 22 和 53。

接下来访问 51，冲突 miss 同时也是强制 miss。根据 LRU 换出 19。

此时统计数据为

**Hit:2, miss:22, replacement: 1**

下边访问 27，强制 miss 和冲突 miss，置换出 11；访问 41，强制 miss；访问 29，命中。

组号	块号0	块号1	块号2	块号3
0	8	56	0	40
1	1	57	41	
2	58	10		
3	51	27	3	29
4	20			
5	13	21	5	53
6	22			
7	23	47	39	

表格 2-4

统计数据为：

**Hit 3, miss 24, replacement 2**

我们把置换后的结果标记为绿色。

下边访问 27，命中；2，强制 miss；25，强制 miss；28，强制 miss；43，冲突 miss 和强制 miss 兼有，换出 3。

组号	块号0	块号1	块号2	块号3
0	8	56	0	40
1	1	57	41	25
2	58	10	2	
3	51	27	43	29
4	20	28		
5	13	21	5	53
6	22			
7	23	47	39	

表格 2-5

Hit 4, miss 28, replacement 3。

访问 41, 命中; 30, 强制 miss; 30, 命中; 47 命中; 63, 强制 miss; 61, 强制 miss 和冲突 miss, 换出 13。

组号	块号0	块号1	块号2	块号3
0	8	56	0	40
1	1	57	41	25
2	58	10	2	
3	51	27	43	29
4	20	28		
5	61	21	5	53
6	22	30		
7	23	47	39	63

表格 2-6

Hit 7, miss 31, replacement 4

访问 37, 强制 miss 和冲突 miss, 置换出 21; 49, 强制 miss 和冲突 miss, 换出 25。

组号	块号0	块号1	块号2	块号3
0	8	56	0	40
1	1	57	41	47
2	58	10	2	
3	51	27	43	29
4	20	28		
5	61	37	5	53
6	22	30		
7	23	47	39	63

表格 2-7

Hit 7, miss 33, replacement 6。

分析结束，和程序结果对比，完全一致。

其中强制 miss33，冲突 miss6，无容量 miss。特别地，在询问指导老师后发现，三种 miss 并不是绝对互斥的，本程序中强制 miss 和冲突 miss 是具有一定的关联性的。