# 哈爾濱工業大學

# 实验报告

## 实 验（五）

题　　目 Cachelab

高速缓冲器模拟

专　　业 计算机类

学　　号 1161800218

班　　级 1636101

学　　生 陈翔

指 导 教 师 史先俊

实 验 地 点 G709

实 验 日 期 2017.12.8

## 计算机科学与技术学院

# 目 录

# 第 1 章 实验基本信息

## 1.1 实验目的

理解现代计算机系统存储器层级结构
掌握 Cache 的功能结构与访问控制策略
培养 Linux 下的性能测试方法与技巧
深入理解 Cache 组成结构对 C 程序性能的影响

## 1.2 实验环境与工具

### 1.2.1 硬件环境

X64 CPU；2GHz；2G RAM；256GHD Disk 以上

### 1.2.2 软件环境

Windows7 64 位以上；VirtualBox/Vmware 11 以上；Ubuntu 16.04 LTS 64 位/优麒麟 64 位；

### 1.2.3 开发工具

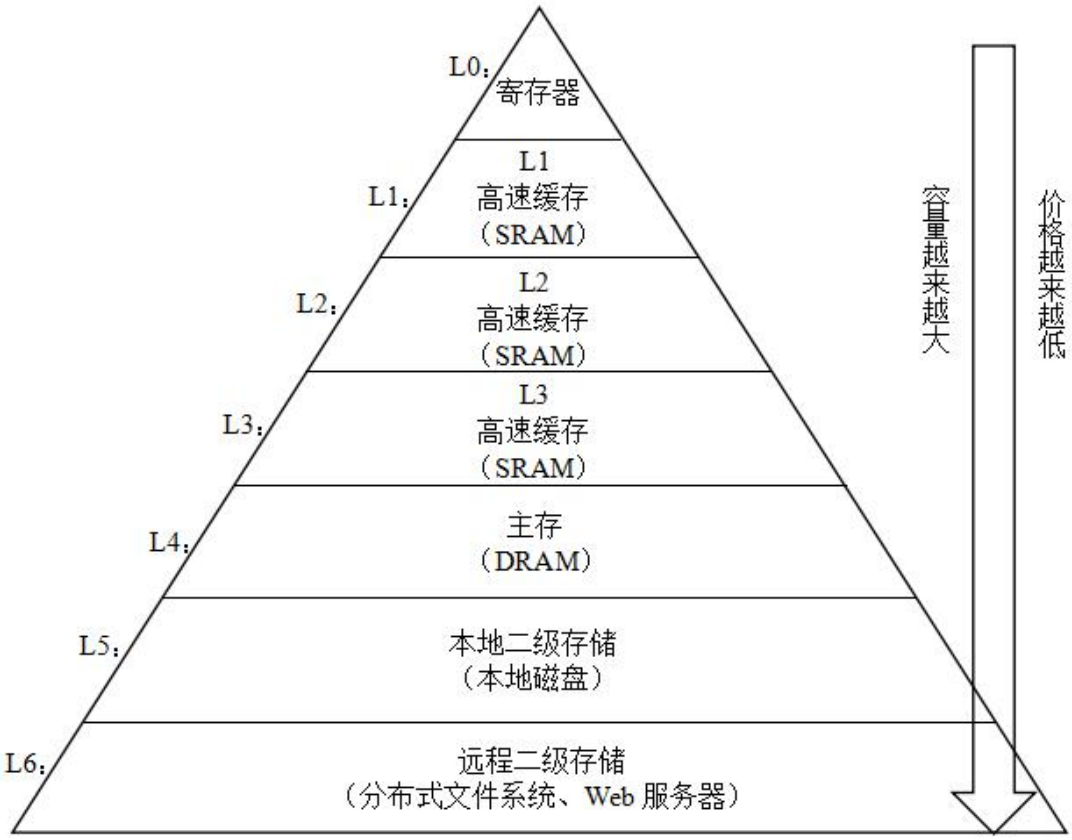
Visual Studio 2010 64 位以上；TestStudio；Gprof;Valgrind 等

## 1.3 实验预习

了解实验的目的、实验环境与软硬件工具、实验操作步骤，复习与实验有关的理论知识。

# 第 2 章 实验预习

2.1 画出存储器层级结构，标识容量价格速度等指标变化（5 分）




2.2 用 CPUZ 等查看你的计算机 Cache 各参数，写出各级 Cache 的 C S E B s e b（5 分）


L1　C=16K　　S=64 s=2 B=64 b=6　E=4 e=1
L2　C=2048K　　S=16384 s=14 B=64 b=6　E=2 e=1

## 2.3 写出各类 Cache 的读策略与写策略（5 分）

1.直接映射
1）组选择 2）行匹配 3）字抽取
替换：用新取出的行替换当前的行
2.组相连
1）组选择 2）行匹配 3）字选择
随机选择要替换的行；最不尝试用；最近最少使用
3.全相连
1）组选择 2）行匹配 3）字选择
并行搜索

直写：立即将 w 的高速缓存块写回到紧接着低一层中；
写回：推迟更新，当替换算法要驱逐更新过的块，才把它写回到紧接着的低一层

中

写分配；非写分配

## 2.4 写出用 gprof 进行性能分析的方法（5 分）

- -b 不再输出统计图表中每个字段的详细描述。
- -p 只输出函数的调用图（Call graph 的那部分信息）。
- -q 只输出函数的时间消耗列表。
- -e Name 不再输出函数 Name 及其子函数的调用图（除非它们有未被限制的其它父函数）。可以给定多个 -e 标志。一个 -e 标志只能指定一个函数。
- -E Name 不再输出函数 Name 及其子函数的调用图，此标志类似于 -e 标志，但它在总时间和百分比时间的计算中排除了由函数 Name 及其子函数所用的时间。
- -f Name 输出函数 Name 及其子函数的调用图。可以指定多个 -f 标志。一个 -f 标志只能指定一个函数。
- -F Name 输出函数 Name 及其子函数的调用图，它类似于 -f 标志，但它在总时间和百分比时间计算中仅使用所打印的例程的时间。可以指定多个 -F 标志。一个 -F 标志只能指定一个函数。-F 标志覆盖 -E 标志。
- -z 显示使用次数为零的例程（按照调用计数和累积时间计算）。

一般用法： gprof －b 二进制程序 gmon.out >report.txt

## 2.5 写出用 Valgrind 进行性能分析的方法（（5 分）

用法: valgrind [options] prog-and-args [options]: 常用选项，适用于所有 Valgrind 工具

-tool=<name> 最常用的选项。运行 valgrind 中名为 toolname 的工具。默认 memcheck。

h –help 显示帮助信息。

-version 显示 valgrind 内核的版本，每个工具都有各自的版本。

q –quiet 安静地运行，只打印错误信息。

v –verbose 更详细的信息, 增加错误数统计。

-trace-children=no|yes 跟踪子线程? [no]

-track-fds=no|yes 跟踪打开的文件描述？ [no]

-time-stamp=no|yes 增加时间戳到 LOG 信息? [no]

-log-fd=<number> 输出 LOG 到描述符文件 [2=stderr]

-log-file=<file> 将输出的信息写入到 filename.PID 的文件里，PID 是运行程序的进行 ID

-log-file-exactly=<file> 输出 LOG 信息到 file

-log-file-qualifier=<VAR> 取得环境变量的值来做为输出信息的文件名。 [none]

-log-socket=ipaddr:port 输出 LOG 到 socket ，ipaddr:port

LOG 信息输出

-xml=yes 将信息以 xml 格式输出，只有 memcheck 可用

-num-callers=<number> show <number> callers in stack traces [12]

-error-limit=no|yes 如果太多错误，则停止显示新错误? [yes]

-error-exitcode=<number> 如果发现错误则返回错误代码 [0=disable]

-db-attach=no|yes 当出现错误，valgrind 会自动启动调试器 gdb。[no]

-db-command=<command> 启动调试器的命令行选项[gdb -nw %f %p]

适用于 Memcheck 工具的相关选项：

1.-leak-check=no|summary|full 要求对 leak 给出详细信息? [summary]

2.-leak-resolution=low|med|high how much bt merging in leak check [low]

3.-show-reachable=no|yes show reachable blocks in leak check? [no]

# 第 3 章 Cache 模拟与测试

## 3.1 Cache 模拟器设计

提交 csim.c

程序设计思想：

LRU 算法的算法实现是基于 LruNumber，是一个简易版本的实现。模拟内存对象初始化时所有 LruNumber 均为 0。当某组中的某行 hit 或者加载缓存成功时赋值 LruNumber 为 MAGIC_LRU_NUMBER，而该组中的其他行 LruNumber 全部减一。当 eviction 时，选择最小 LruNumber 的行进行 evict。

L 对应 loadData，这个操作会可能引发如下几种情况：hit OR miss OR miss eviction。 S 对应 storeData，这个操作可能引发的情况与 L 相同，所以实现中直接在 storeData 函数中调用了 loadData 函数 M 对应 modifyData，这个操作实质上是先调用 load data then store data，所以可能会出现 2 次 hit or miss hit or miss evition hit 三种情况。

测试用例 1 的输出截图（5 分）：

```
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./csim -s 1 -E 1 -b 1 -t traces/yi2.trace
hits:9 misses:8 evictions:6
```

测试用例 2 的输出截图（5 分）：

```
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./csim -s 4 -E 2 -b 4 -t traces/yi.trace
hits:4 misses:5 evictions:2
```

测试用例 3 的输出截图（5 分）：

```
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./csim -s 2 -E 1 -b 4 -t traces/dave.trace
hits:2 misses:3 evictions:1
```

测试用例 4 的输出截图（5 分）：

```
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./csim -s 2 -E 1 -b 3 -t traces/trans.trace
hits:167 misses:71 evictions:67
```

测试用例 5 的输出截图（5 分）：

```
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./csim -s 2 -E 2 -b 3 -t traces/trans.trace
hits:201 misses:37 evictions:29
```

测试用例 6 的输出截图（5 分）：

```
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./csim -s 2 -E 4 -b 3 -t traces/trans.trac
hits:212 misses:26 evictions:10
```

测试用例 7 的输出截图（5 分）：

```
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./csim -s 5 -E 1 -b 5 -t traces/trans.trac
hits:231 misses:7 evictions:0
```

测试用例 8 的输出截图（10 分）：

```
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./csim -s 5 -E 1 -b 5 -t traces/long.trace
hits:265189 misses:21775 evictions:21743
```

**注：每个用例的每一指标 5 分（最后一个用例 10）**——与参考 csim-ref 模拟器输出指标相同则判为正确

```
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./test-csim
                        Your simulator     Reference simulator
Points (s,E,b)    Hits  Misses  Evicts    Hits  Misses  Evicts
     3 (1,1,1)       9       8       6       9       8       6  traces/yi2.trace
     3 (4,2,4)       4       5       2       4       5       2  traces/yi.trace
     3 (2,1,4)       2       3       1       2       3       1  traces/dave.trace
     3 (2,1,3)     167      71      67     167      71      67  traces/trans.trace
     3 (2,2,3)     201      37      29     201      37      29  traces/trans.trace
     3 (2,4,3)     212      26      10     212      26      10  traces/trans.trace
     3 (5,1,5)     231       7       0     231       7       0  traces/trans.trace
     6 (5,1,5)  265189   21775   21743  265189   21775   21743  traces/long.trace
    27

TEST_CSIM_RESULTS=27
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./csim -s 1 -E 1 -b 1 -t traces/yi2.trace
hits:9 misses:8 evictions:6
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./csim -s 4 -E 2 -b 4 -t traces/yi.trace
hits:4 misses:5 evictions:2
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./csim -s 2 -E 1 -b 4 -t traces/dave.trace
hits:2 misses:3 evictions:1
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./csim -s 2 -E 1 -b 3 -t traces/trans.trace
hits:167 misses:71 evictions:67
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./csim -s 2 -E 2 -b 3 -t traces/trans.trace
hits:201 misses:37 evictions:29
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./csim -s 2 -E 4 -b 3 -t traces/trans.trace
hits:212 misses:26 evictions:10
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./csim -s 5 -E 1 -b 5 -t traces/trans.trace
hits:231 misses:7 evictions:0
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./csim -s 5 -E 1 -b 5 -t traces/long.trace
hits:265189 misses:21775 evictions:21743
```

```
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./test-csim
                        Your simulator     Reference simulator
Points (s,E,b)    Hits  Misses  Evicts    Hits  Misses  Evicts
     3 (1,1,1)       9       8       6       9       8       6  traces/yi2.trace
     3 (4,2,4)       4       5       2       4       5       2  traces/yi.trace
     3 (2,1,4)       2       3       1       2       3       1  traces/dave.trace
     3 (2,1,3)     167      71      67     167      71      67  traces/trans.trace
     3 (2,2,3)     201      37      29     201      37      29  traces/trans.trace
     3 (2,4,3)     212      26      10     212      26      10  traces/trans.trace
     3 (5,1,5)     231       7       0     231       7       0  traces/trans.trace
     6 (5,1,5)  265189   21775   21743  265189   21775   21743  traces/long.trace
    27

TEST_CSIM_RESULTS=27
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$
```

```
linux> ./test-csim
                        Your simulator     Reference simulator
Points (s,E,b)    Hits  Misses  Evicts    Hits  Misses  Evicts
3       (1,1,1)   9       8       6       9       8       6  traces/yi2.trace
3       (4,2,4)   4       5       2       4       5       2  traces/yi.trace
3       (2,1,4)   2       3       1       2       3       1  traces/dave.trace
3       (2,1,3)   167     71      67      167     71      67  traces/trans.trace
3       (2,2,3)   201     37      29      201     37      29  traces/trans.trace
3       (2,4,3)   212     26      10      212     26      10  traces/trans.trace
3       (5,1,5)   231     7       0       231     7       0  traces/trans.trace
6       (5,1,5)   265189  21775   21743   265189  21775   21743  traces/long.trace
27
```

## 3.2 矩阵转置设计

提交 trans.c

程序设计思想：

利用局部性原理

一共有 32 个 set 每个 set1line 每个 block 有 32 个 byte

32 时： 一个 int 元素 4byte 也就是 cache 里面存放这些元素的时候 一个 set 只能放 8 个

64 时：矩阵大了 所以不能完全放在 cache 里面；/相当于把 64*64 的矩阵分割成了 8*8，/同样是先取八个出来，前面四个正常放置，后面四个先并列放好

61 时：所以先划成小一点的形状，到了最后一大块的时候再继续分

**32×32（10 分）：运行结果截图**

```
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./test-trans -M 32 -N 32

Function 0 (2 total)
Step 1: Validating and generating memory traces
Step 2: Evaluating performance (s=5, E=1, b=5)
func 0 (Transpose submission): hits:1766, misses:287, evictions:255

Function 1 (2 total)
Step 1: Validating and generating memory traces
Step 2: Evaluating performance (s=5, E=1, b=5)
func 1 (Simple row-wise scan transpose): hits:870, misses:1183, evictions:1151

Summary for official submission (func 0): correctness=1 misses=287

TEST_TRANS_RESULTS=1:287
```

**64×64（10 分）：运行结果截图**

```
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./test-trans -M 32 -N 32

Function 0 (2 total)
Step 1: Validating and generating memory traces
Step 2: Evaluating performance (s=5, E=1, b=5)
func 0 (Transpose submission): hits:1766, misses:287, evictions:255

Function 1 (2 total)
Step 1: Validating and generating memory traces
Step 2: Evaluating performance (s=5, E=1, b=5)
func 1 (Simple row-wise scan transpose): hits:870, misses:1183, evictions:1151

Summary for official submission (func 0): correctness=1 misses=287

TEST_TRANS_RESULTS=1:287
```

**61×67（20 分）：运行结果截图**

```
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./test-trans -M 61 -N 67

Function 0 (2 total)
Step 1: Validating and generating memory traces
Step 2: Evaluating performance (s=5, E=1, b=5)
func 0 (Transpose submission): hits:8113, misses:1986, evictions:1954

Function 1 (2 total)
Step 1: Validating and generating memory traces
Step 2: Evaluating performance (s=5, E=1, b=5)
func 1 (Simple row-wise scan transpose): hits:3756, misses:4423, evictions:4391

Summary for official submission (func 0): correctness=1 misses=1986

TEST_TRANS_RESULTS=1:1986
```

```
xiangxiang@xiangxiang-Lenovo-G50-75m:~/桌面/实验五/cachelab-handout$ ./driver.pyPart A:
Running ./test-csim
                        Your simulator     Reference simulator
Points (s,E,b)    Hits  Misses  Evicts    Hits  Misses  Evicts
     3 (1,1,1)       9       8       6       9       8       6  traces/yi2.trace
     3 (4,2,4)       4       5       2       4       5       2  traces/yi.trace
     3 (2,1,4)       2       3       1       2       3       1  traces/dave.trace
     3 (2,1,3)     167      71      67     167      71      67  traces/trans.trace
     3 (2,2,3)     201      37      29     201      37      29  traces/trans.trace
     3 (2,4,3)     212      26      10     212      26      10  traces/trans.trace
     3 (5,1,5)     231       7       0     231       7       0  traces/trans.trace
     6 (5,1,5)  265189   21775   21743  265189   21775   21743  traces/long.trace
    27


Part B: Testing transpose function
Running ./test-trans -M 32 -N 32
Running ./test-trans -M 64 -N 64
Running ./test-trans -M 61 -N 67

Cache Lab summary:
                        Points   Max pts      Misses
Csim correctness          27.0        27
Trans perf 32x32           8.0         8         287
Trans perf 64x64           8.0         8        1219
Trans perf 61x67          10.0        10        1986
          Total points    53.0        53
```

# 第 4 章 总结

## 4.1 请总结本次实验的收获

对 cache 有了更加深刻的认识
对 cache 的结构，读写规则有了更加深刻的认识
对 LRU 算法的实现也有了具体的认识
对编写 cache 友好的代码有了深刻的理解
对局部性原理有了较好的认识

## 4.2 请给出对本次实验内容的建议

注：本章为酌情加分项。

# 参考文献

**为完成本次实验你翻阅的书籍与网站等**

[1] 林来兴. 空间控制技术[M]. 北京：中国宇航出版社，1992：25-42.

[2] 辛希孟. 信息技术与信息服务国际研讨会论文集：A 集[C]. 北京：中国科学出版社，1999.

[3] 赵耀东. 新时代的工业工程师[M/OL]. 台北：天下文化出版社，1998 [1998-09-26]. http://www.ie.nthu.edu.tw/info/ie.newie.htm（Big5）.

[4] 谌颖. 空间交会控制理论与方法研究[D]. 哈尔滨：哈尔滨工业大学，1992：8-13.

[5] KANAMORI H. Shaking Without Quaking[J]. Science，1998，279（5359）：2063-2064.

[6] CHRISTINE M. Plant Physiology: Plant Biology in the Genome Era[J/OL]. Science，1998，281：331-332[1998-09-23]. http://www.sciencemag.org/cgi/collection/anatmorp.