实验4：性能优化实验

**一、实验目的**

本实验是在CACHELAB基础上进行，利用高速缓冲存储器机制，对高级语言程序性能优化。通过本实验加深对程序局部性和Cache工作原理，掌握C语言编程和算法优化的技巧，以编写高效、正确的代码。

**二、实验内容**

通过修改trans.c，优化矩阵转置函数，使其内存访问生成的记录，由cache模拟器处理后，提高缓存命中率。

**三、实验步骤（同实验3，可省略）**

1. 请各位同学在ubuntu的浏览器输入<http://10.160.106.190/cachelab-handout.tar>，下载该压缩包。

2. 解压压缩包

linux>tar -xvf cachelab-handout.tar解压刚下载的压缩包。

压缩包中主要包含以下文件：

csim.c:实现缓存模拟器的文件

tran.c:实验矩阵转置的文件

csim-ref:标准的缓存模拟器，可以做运行结果对比

test-trans:测试你的矩阵转置优化的如何，并给出评分。

traces子目录包含参考跟踪文件的集合，将用于评估在编写的缓存模拟器的正确性。跟踪文件是由 valgrind的程序产生的。

跟踪文件中每一行代表一次对缓存的操作，格式为：操作-地址-大小三元组

操作共有以下四种：

I：加载指令

L：加载数据

S：存储数据

M：修改数据

其中地址为64位，数据大小以字节为单位。

3. 安装valgrin

linux>sudo apt-get install valgrind

4.安装python2

linux>sudo apt install python2

**四、实验过程**

(1)修改完善trans.c程序

该程序实现了一个矩阵转置，即计算N×M矩阵A的转置，并将结果存储在M×N矩阵B中，即。

程序中的转置函数trans虽然正确，但效率低下，许多缓存命中率低。要求编写程序中的transpose\_submit函数，对转置函数进行优化，例如可以采用分块转置、优化循环顺序等方法来提高命中率。  
 输入是三个不同的固定大小的矩阵，矩阵分别为：

32 x 32，满分要求：miss 次数小于 300。（必做）

64 x 64，满分要求：miss 次数小于 1300。（选作）

61 x 67，满分要求：miss 次数小于 2000。（选作）

其他限制：

最多使用 12 个 int 类型的本地变量；

不能修改 A 矩阵的内容，B 矩阵可随意修改；

不能用 malloc 类函数申请内存；

矩阵A和矩阵B的内存分布是连续的。

* （2）编译链接

linux> make clean

make

（3）测试验证

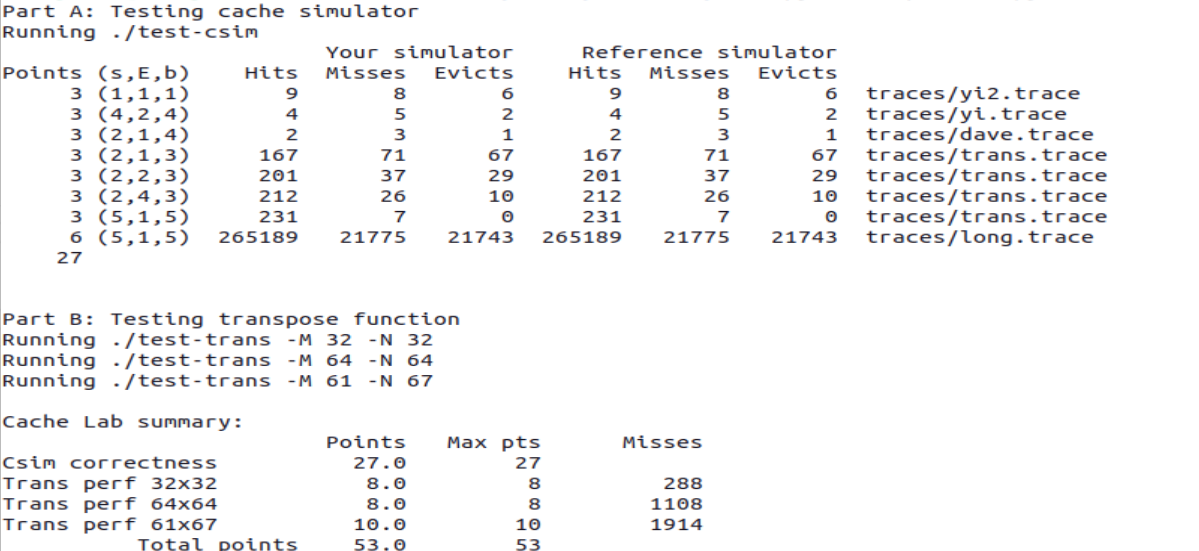
使用提供的测试工具test-trans验证转置函数效率。

linux> ./test-trans

例如：./test-trans -M 32 -N 32

（4）总测试

linux>python2 ./driver.py



**五、实验程序提交**

1.在个人目录下创建子目录lab8。

>>>>> 在服务器上 >>>>>

cd /home/username

mkdir lab8

4.将trans.c 和实验报告复制到子目录lab8中。

实验报告命名格式：学号姓名lab8.docx

注意: trans.c文件中一定要将你的姓名和学号用注释方式放在开始。

**六、通知**

1.截止日期：2024.11.27

2.当实验室提交日期到期时，我们将从子目录中取走符合命名格式的作业，延迟提交或不符合命名格式的作业将不会被取走。

3.这个运行于10.160.106.190的实验室系统在我们的校园内。如果你在我们的外面校园，请使用南邮VPN客户端程序ENWAgent。