****

**计算机组成原理实验报告**

**题目：**cache 和程序访问的局部性（一）

**姓 名： 段欣然**

**专 业： 计算机科学与技术**

**年 级： 2020级**

**学 号： 202011081033**

**任课教师： 王志春**

**完成日期： 2021年5月20日**

**人工智能学院**

1. 实验要求

通过实际程序的执行结果，了解程序访问的局部性对带有 cache 的计算机系统性能的影响。

正文中文字体统一为宋体，英文及数字字体为Times New Roman，字号小四，行间距最小值20磅，段落首行缩进2字符

1. 实验结果与分析

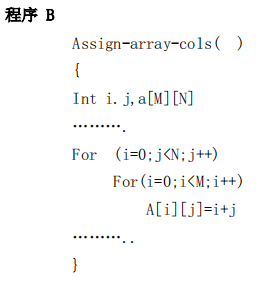
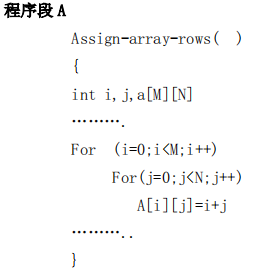


figure 程序代码示例

在如figure 1所示程序中，修改或添加必要的语句（如计时函数等），以计算和打印主体程序段的执行时间。分别以、、，执行程序 A 和程序 B，以比较两个程序执行时间的长短。下为实验代码及运行结果示例。

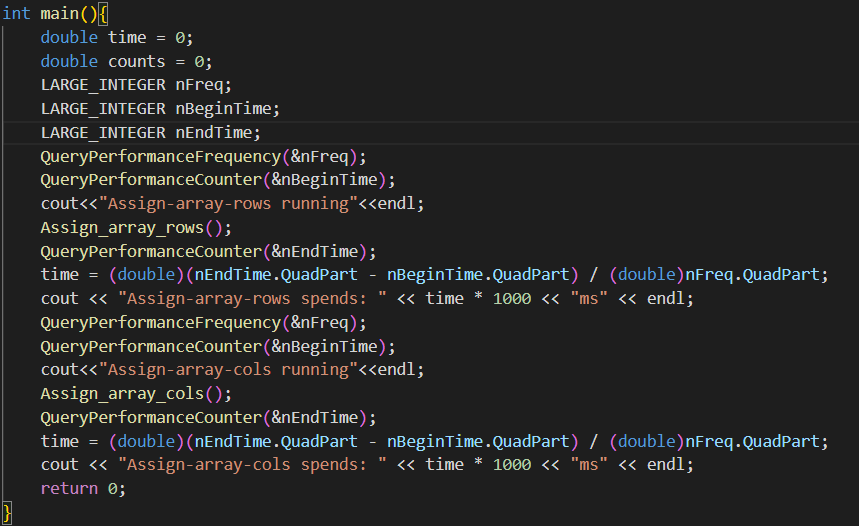


figure 2实验主要代码示例

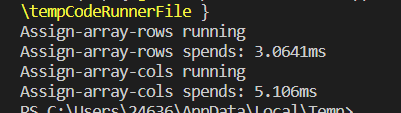


figure 3程序运行结果示例

## 实验结果

3次实验结果下表所示。

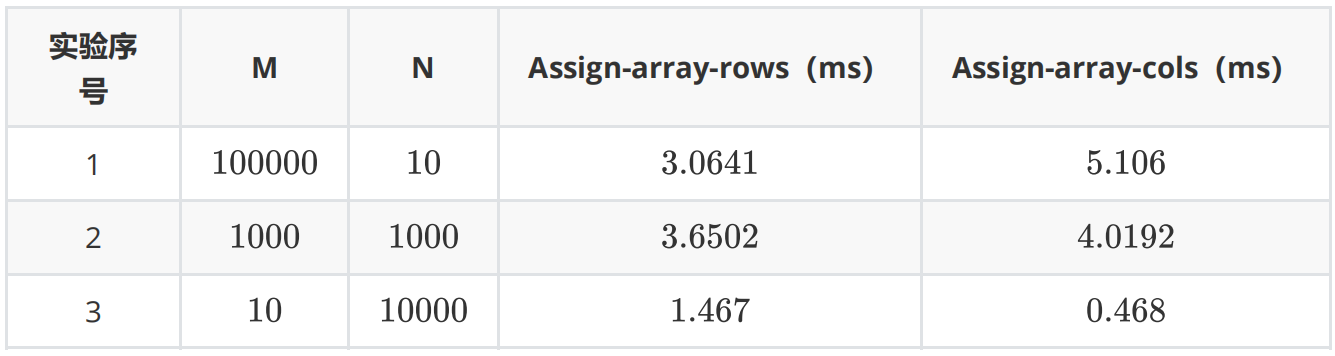


table 实验结果

## 结果分析

在实验1中，由于Assign-array-rows按先行后列的顺序访问数组，每次都访问离上次访问位置最近的元素，故从缓存中调用数据次数较多，从主存中调用数组次数较少。而Assign-array-cols按先列后行的顺序访问数组，每次访问有固定跨度且跨度较大（），故基本都需要从主存中读取数据。因此实验1结果呈现出Assign-array-rows耗时少于Assign-array-col。

在实验2中，数组行列大小一致，但Assign-array-rows仍然每次访问离上次访问位置最近的元素，故读取缓存中数据次数较多，读取主存中数据次数较少；而Assign-array-cols同样每次具有固定跨度地访问数组，因此Assign-array-rows运行时间少于Assign-array-cols。

在实验3中，数组每行数据较多，当cache大小较小时，可能发生多次替换，命中可能性减小。所以结果呈现出Assign-array-rows运行时间多于Assign-array-cols。

1. 实验小结

我们编写程序时可以充分利用局部性原理以加快运行速率，但需要注意缓存中每行的块大小不能过大，否则可能类似实验3的结果，先列后行反而比先行后列的顺序访问更快。