**操作系统实验报告**

**实验名称： 实验三 多线程程序实验**

**姓名： 刘硕**

**学号： 16340154**

一、实验目的

1.进一步理解线程。

2.学会使用pthread线程库。

二、实验要求

1.用线程生成fibonacci数列。

2. 多线程矩阵乘法。

三、实验过程

1. **用线程生成fibonacci数列**

**pthread库函数**

多线程编程设计的主要部分是线程创建，参数传递，数据同步，结果返回以及线程销毁。在pthread库里提供了一些方法来实现这种编程。

**pthread\_create：创建线程**。函数原型：int pthread\_create(pthread\_t \*restrict tid, const pthread\_attr\_t \*restrict attr, void \*(\*start\_rtn)(void), void \*restrict arg)。第一个参数为指向线程标识符的指针，用于输出线程标识符；第二个参数用来设置线程属性；第三个参数是线程运行函数的起始地址；最后一个参数是运行函数的参数。

**pthread\_join：阻塞调用线程。**阻塞调用线程，直到threadid所指定的线程终止。每个线程只能join一次，若多次就会发生逻辑错误。函数原型：int pthread\_join(pthread\_t tid, void \*\*status)。第一个参数为指向线程标识符的指针，用于输出线程标识符；第二个参数用来设置线程属性。

**pthread\_exit：终止调用线程。**函数原型：void pthread\_exit()。

**pthread\_attr\_init：对线程属性初始化。**函数原型：int pthread\_attr\_init(pthread\_attr\_t\* attr)。参数是线程属性变量。

**pthread\_attr\_getscope：获得线程竞争范围。**函数原型：int pthread\_attr\_getscope (pthread\_attr\_t\* attr, int scope)。第一个参数是线程属性变量；另一个是线程作用域。指定了作用域就说明了进程与谁竞争资源。

**pthread\_attr\_setscope：获得线程竞争范围。**函数原型：int pthread\_attr\_setscope (pthread\_attr\_t\* attr, int scope)。第一个参数是线程属性变量；另一个是线程作用域。指定了作用域就说明了进程与谁竞争资源。

**斐波拉契数列**

在主线程中创建一个无属性的线程tid，该线程执行fibonacci方法的功能。并在计算好fibonacci数列相应的元素之后阻塞这个线程。之后利用主线程输出结果。效果如图。

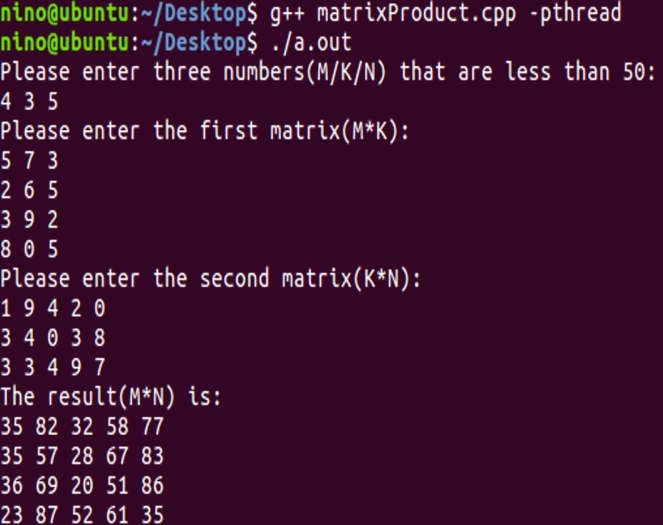
1. **按要求实现一个多线程矩阵乘法**

**矩阵乘法运算**

给定两个矩阵A和B，其中A是具有M行、K列的矩阵， B为K行、N列的矩阵， A和B的矩阵积为矩阵C， C为M行、N列。矩阵C中第i行、第j列的元素Cij就是矩阵A第i行每个元素和矩阵B第j列每个元素乘积的和。



**用多线程的方法计算矩阵的积**

每个Ci j的计算用一个独立的工作线程，因此它将会涉及生成M\*N个工作线程。主线程(或称为父线程)将初始化全局矩阵A和B，并分配足够的内存给全局矩阵C，它将容纳矩阵A和B的积。创建完毕之后开辟一个有属性的线程，用于计算线程矩阵，结果储存在C矩阵当中。C中所有项都计算完成之后，用主线程输出。结果如图。

四、实验心得

fibonacci实验较为简单，但是在多线程完成矩阵乘法的实验过程中遇到了很多困难。首先，我一开始没有考虑线程属性。这之后我的算法出现了问题，导致没有运行出正确的结果，后来经过断点调试这个问题终于解决。通过这次实验，我进一步理解了线程，并且了解了pthread库中的有关函数。