**操 作 系 统**

**实 验 报 告**

**实验名称：实验3 多线程程序实验**

**姓名： 陈衍斌**

**学号： 16340042**

实验名称：多线程程序实验

一、实验目的：

1. 加深对线程概念的理解。
2. 初步认识pthread线程库并进行运用。

二、实验要求：

1. 用线程生成Fibonacci数列：用pthread线程库，按照第四章习题4.11的要求生成并输出Fibonacci数列。

2. 多线程矩阵乘法：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，即



**要求**：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)将初始化矩阵A和B，并分配足够的内存给矩阵C，它将容纳矩阵A和B的积。这些矩阵将声明为全局数据，以使每个工作线程都能访问矩阵A、B和C。

三、实验过程：

**1．**在进行实验之前，需要先了解一些前置知识：

pthread\_create()：创建线程（实际上就是确定调用该线程函数的入口点），在线程创建以后，就开始运行相关的线程函数。成功返回0。

* 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。

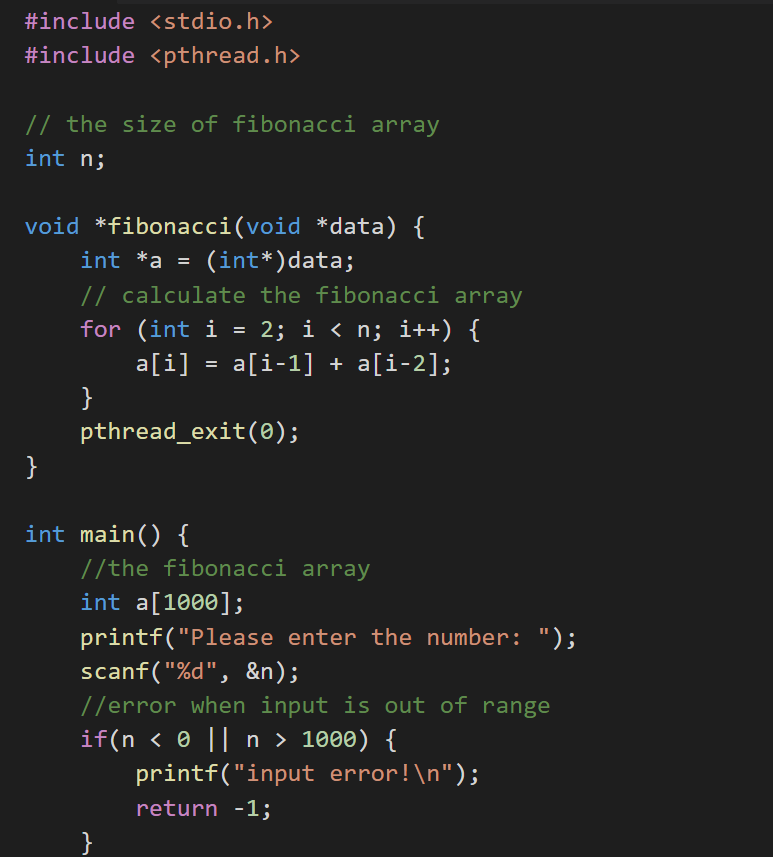
pthread\_join()：阻塞调用线程，直到threadid所指定的线程终止。

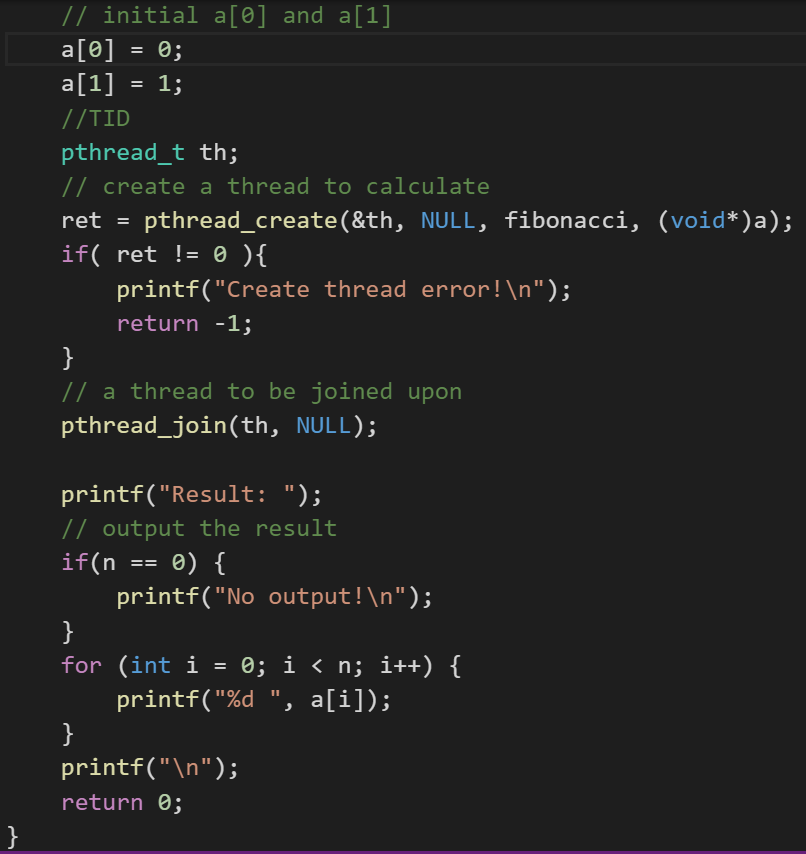
pthread\_exit()：终止调用线程，这是线程的主动行为。

使用pthread的程序的编译命令：

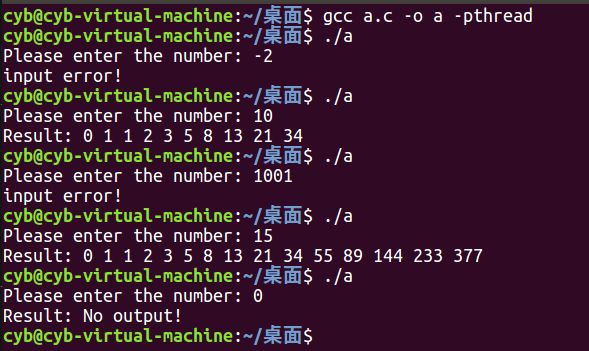


一些细节写在代码注释里了，就不多说，直接上代码：





运行的结果如下：



2. 前置知识：

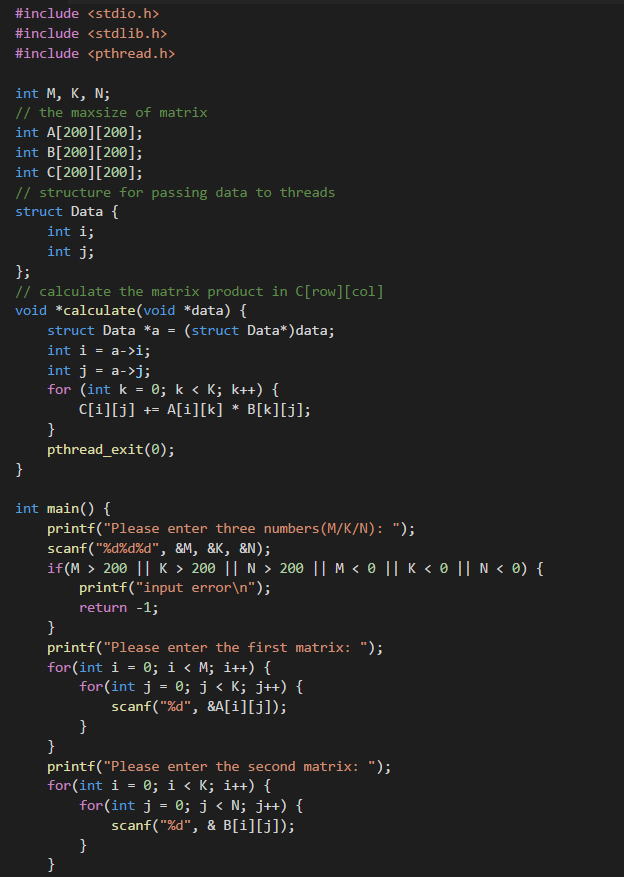
pthread\_attr\_init ()：初始化线程属性为默认属性

pthread\_attr\_getscope ()：获得线程竞争范围

pthread\_attr\_setscope ()：设置线程竞争范围

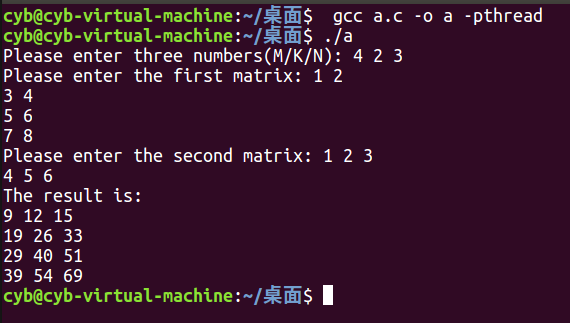
选择用struct传指针。基本原理都差不多，也没什么别的可讲的。

代码如下：





结果如下：



总体来说，这次实验其实并没有很大的难度。课本上对线程有了较详细的描述，ppt上也给了一些具体的例子，综合运用起来很快就能完成实验。通过这次实验，我对线程的了解进一步加深。这次体会到了多线程的好处：资源利用率更好、程序设计在某些情况下更简单，而且响应更快。