2024年12月16日

15:36

# 第一次实验：

## 一、 实验目的

在单处理器环境下，实现多任务的核心是并发程序设计，进程的并发执行提高了CPU的利用率，使得CPU与设备并行、设备与设备并行成为可能。但并发执行也存在许多单任务中所没有的问题，其中之一是互斥和同步的控制。

## 二、 实验要求

在BACI环境下，对程序并发执行的实验：

（1）没有控制时正确的程序执行的结果不正确；

（2）BACI中PV操作的并发控制的实现。

要求：

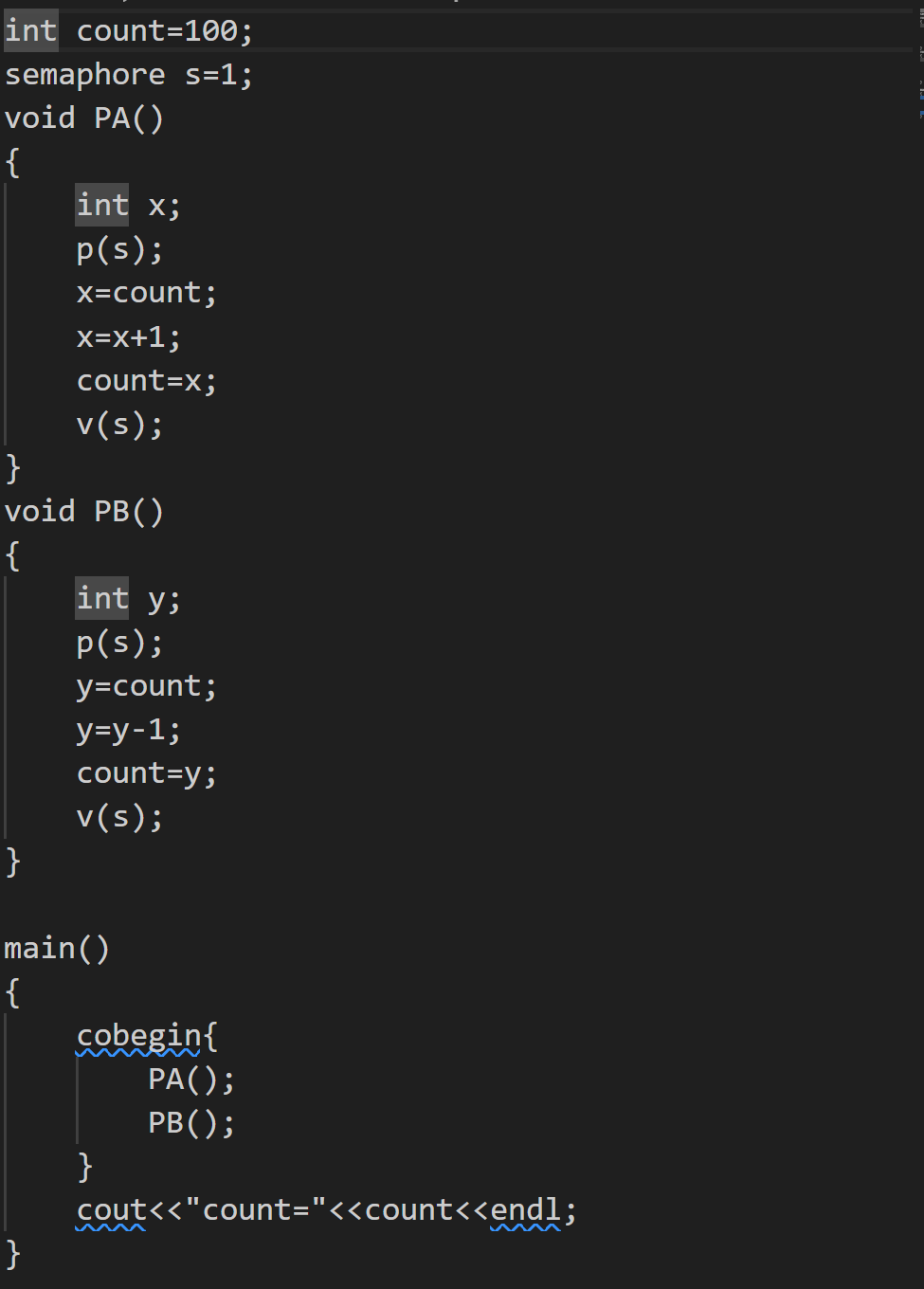
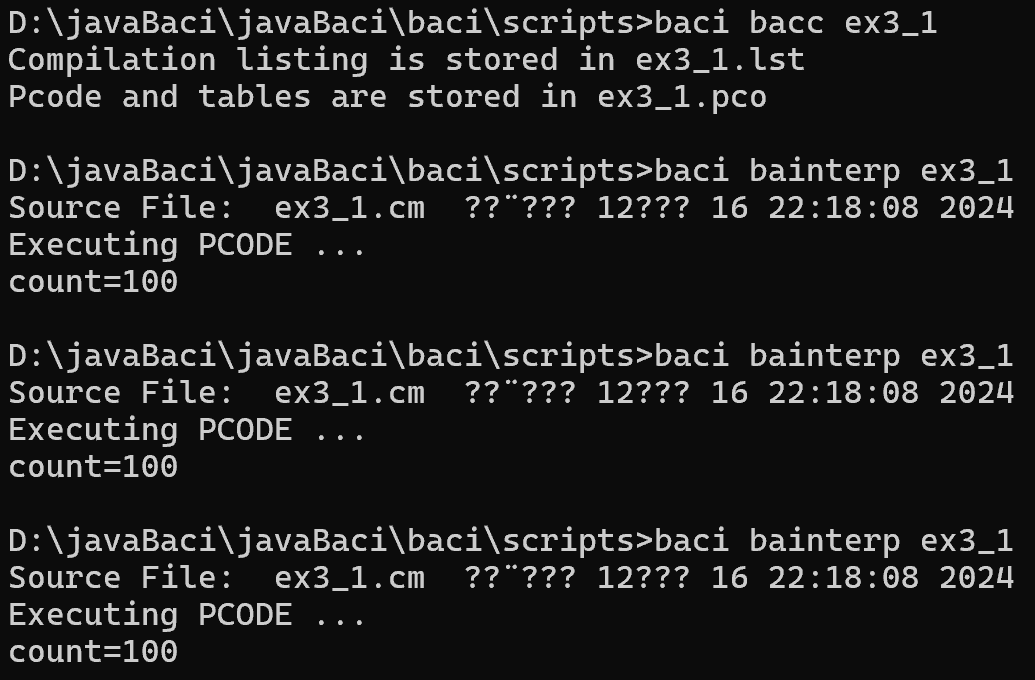
（1）熟悉教材中有关进程并发执行的内容。

（2）设计并实现：没有控制时正确的程序执行的结果不正确。

（3）BACI中PV操作的并发控制的实现。

（4）将课堂和习题中的同步、互斥的并发程序设计的习题的设计与实现

## 三、实验任务

1. 编写两道程序PA和PB，实现对如下问题的模拟：它们对同一个变量count进行操作，PA程序每次运行时对变量count进行加1操作，而PB程序每次运行时对变量count进行减1操作。count的初始值为100。请用PV操作描述他们的同步程序。  
    
2. 两道系统程序A、B共享一个整形变量count，其代码如下

A(){

    count = count +1;

    printf("count= %d",count)

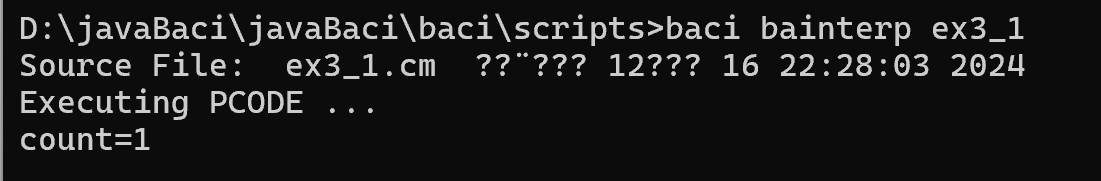
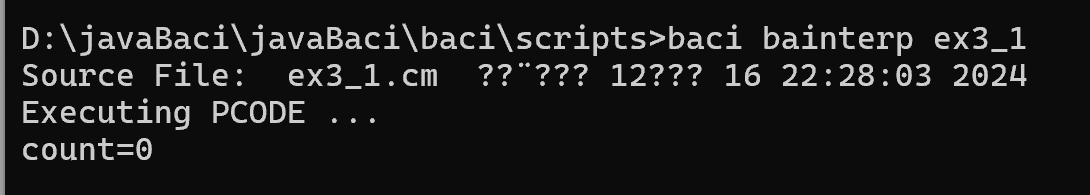
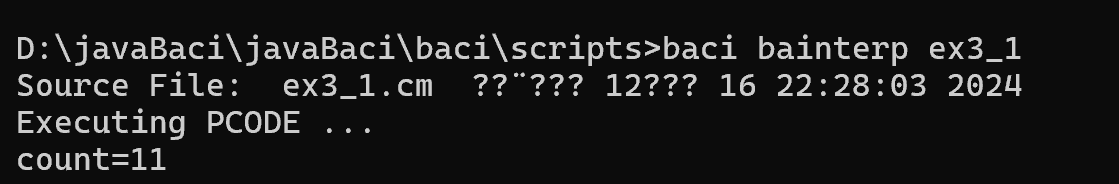
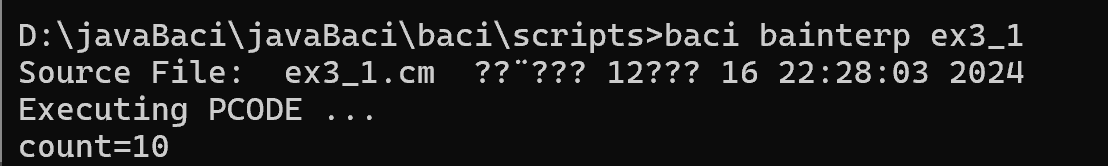
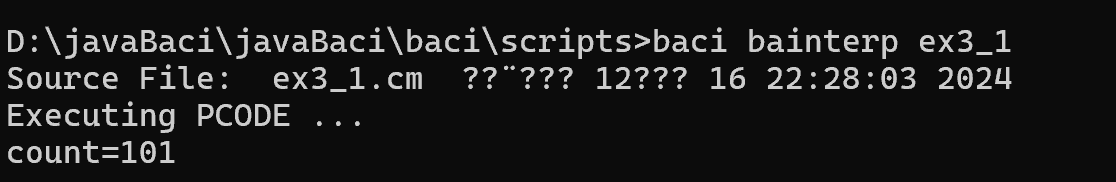
}

B(){

    count = 0;

    count = count + 10;

}

假定count的初值都为100，那么在多道程序设计环境下，A、B各执行一次，请给出printf()所有可能的输出结果  
    

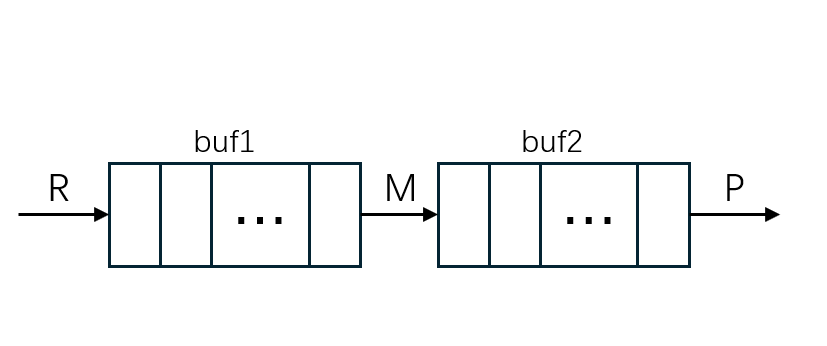
1. 今有3个并发进程R、M、P，如下图所示。它们的任务如下：

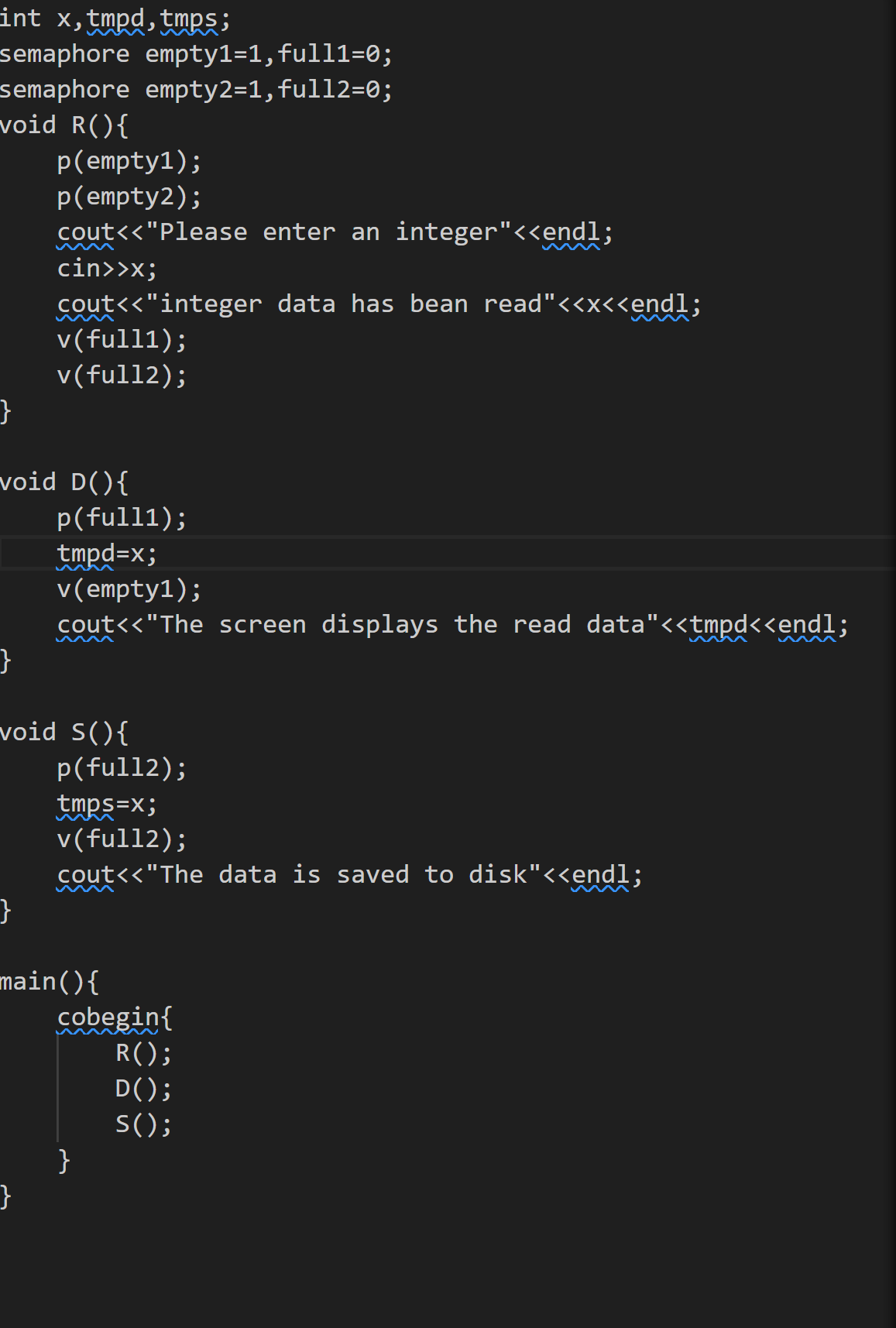
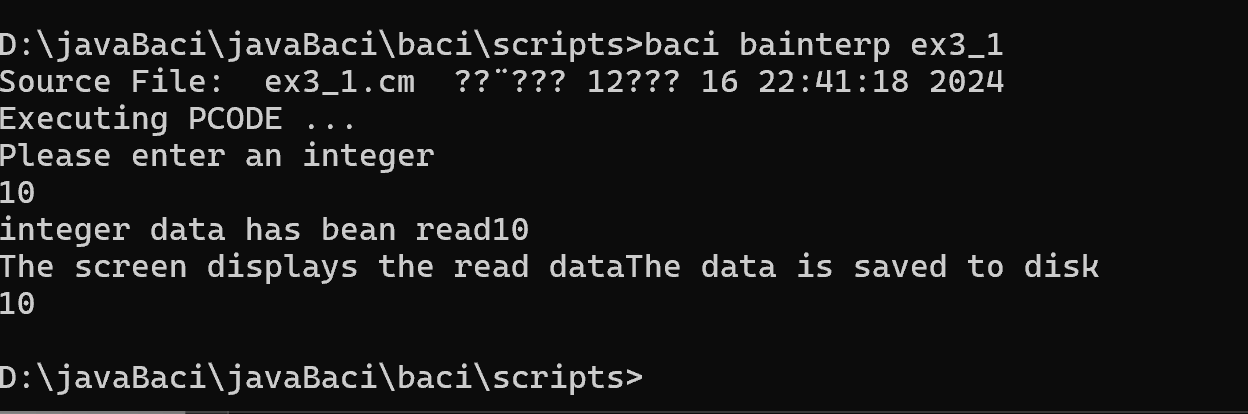
R负责从输入设备读取记录信息，每读一个记录后，把他存在缓冲区队列buf1；

M从缓冲队列buf1中读取记录后，读出后加工记录并把结果存入缓冲区队列buf2中；

P从缓冲区队列buf中读取记录打印输出

假定缓冲区队列buf1有m个缓冲区，缓冲区队列buf2中有n个缓冲区，且有 m > 1、n > 1,

一个记录存放在一个缓冲区。请用信号量机制实现R、M和P的并发执行

# 第二次实验：

## 一、 实验目的

熟悉Linux常用命令，熟悉Vim文本编辑器代码的使用，熟悉Linux消息队列机制，熟悉多任务并行机制的运作，熟悉进程的各种状态变化，熟悉消息队列的通信过程

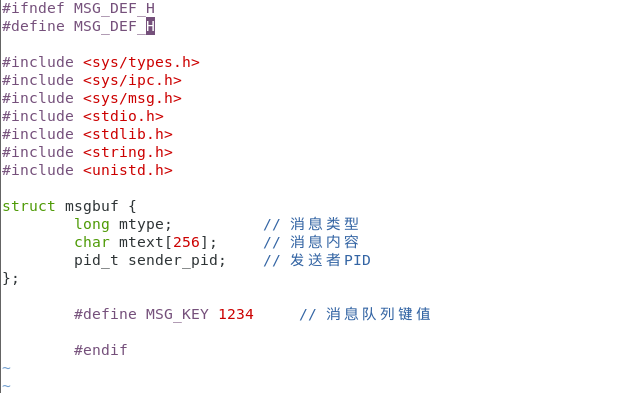
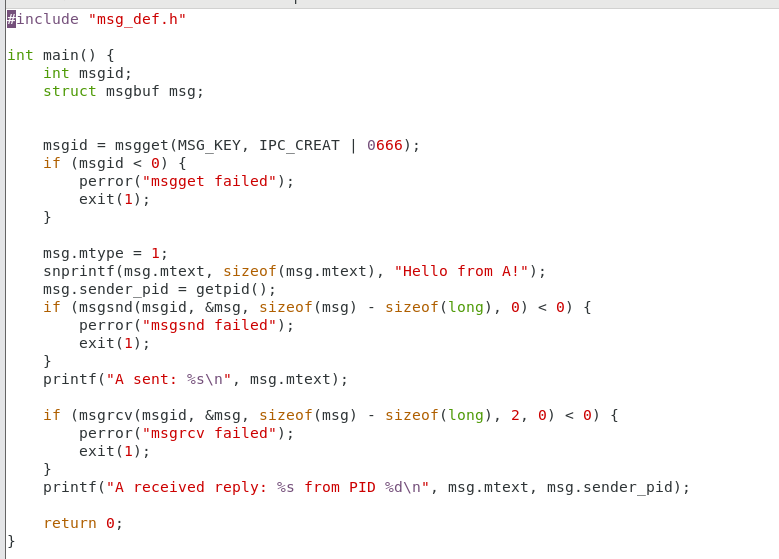
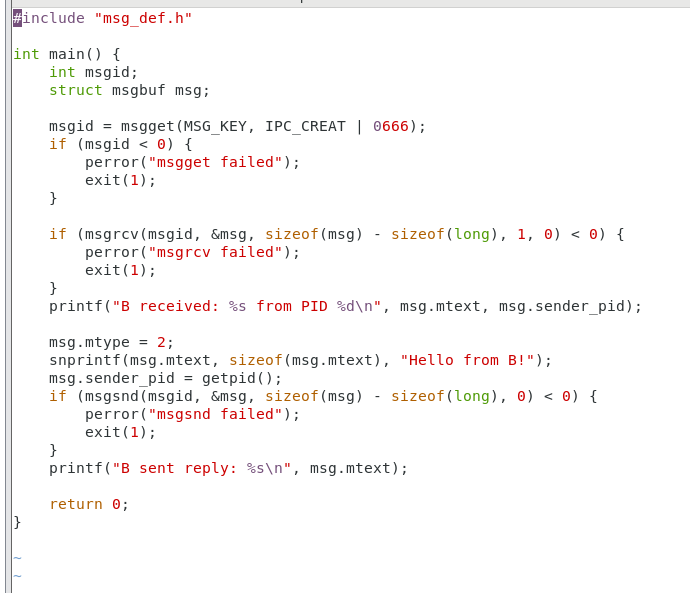
## 二、 实验要求

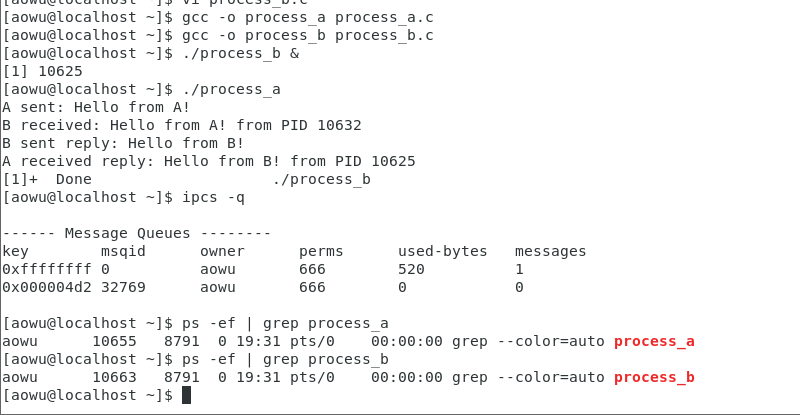
Linux下的进程相关命令，进程状态的查看，使用Vim文本编辑器设计程序完成基本的消息队列通信，并且观察进程运行结果

## 三、 实验任务

1. 按下述要求完成程序的设计：

编写两道程序，实现A、B进程之间的双向通信，通过消息队列实现以下核心功能：

1. 进程间发送与接收消息：
   1. A 向 B 发送消息。
   2. B 接收到消息后回复给 A。
   3. 动态确认通信双方的身份：
2. 每条消息都包含发送者的进程 ID，接收者根据消息类型动态处理通信。
3. 使用Linux的相关指令，对第一题的代码进行编译和运行，查看代码的运行结果和当前相关进程的进程信息
4. 创建头文件msg\_def.h定义消息队列通信  
   
5. 程序A的代码  
   
6. 程序B的代码  
   
7. 运行结果以及相关进程信息



# 第三次实验：

## 一、 实验目的

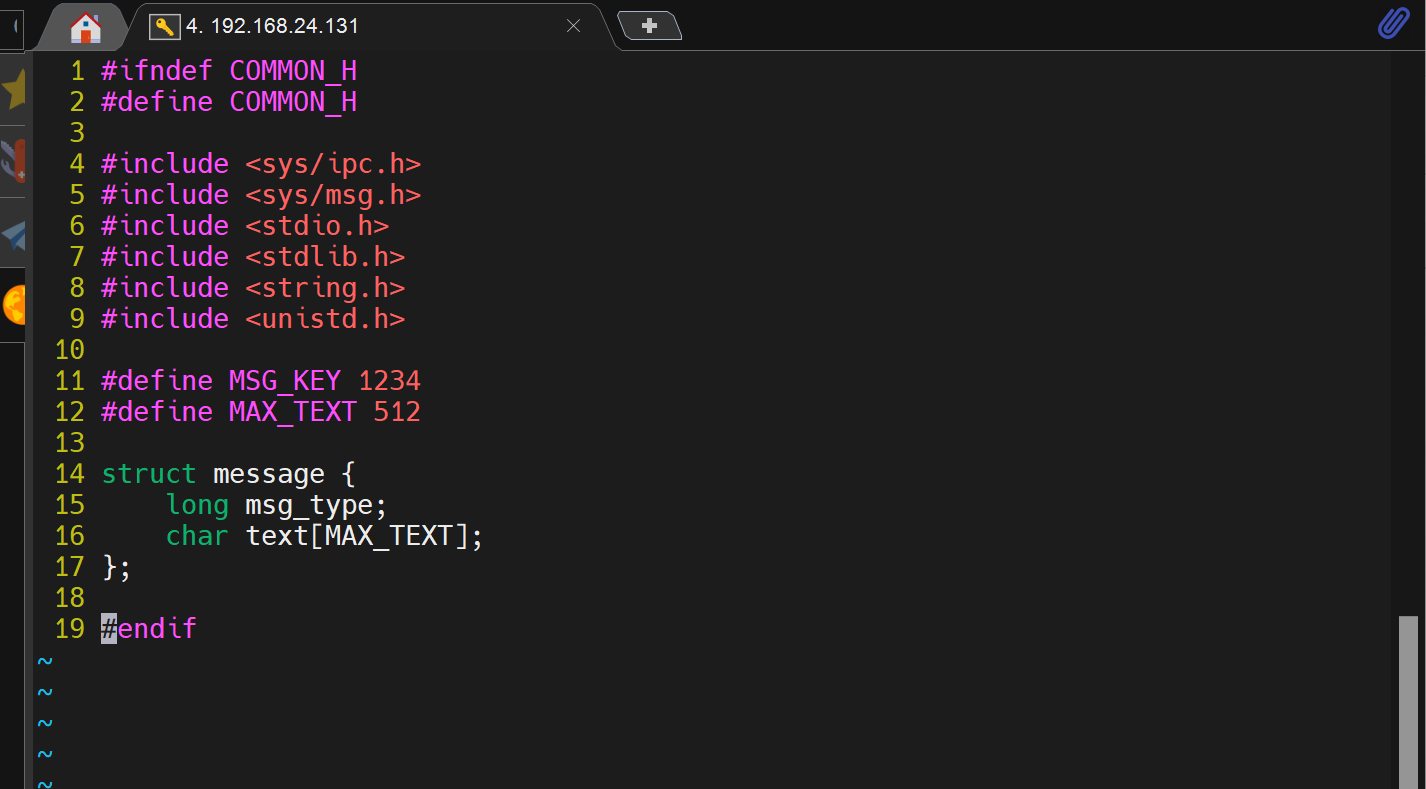
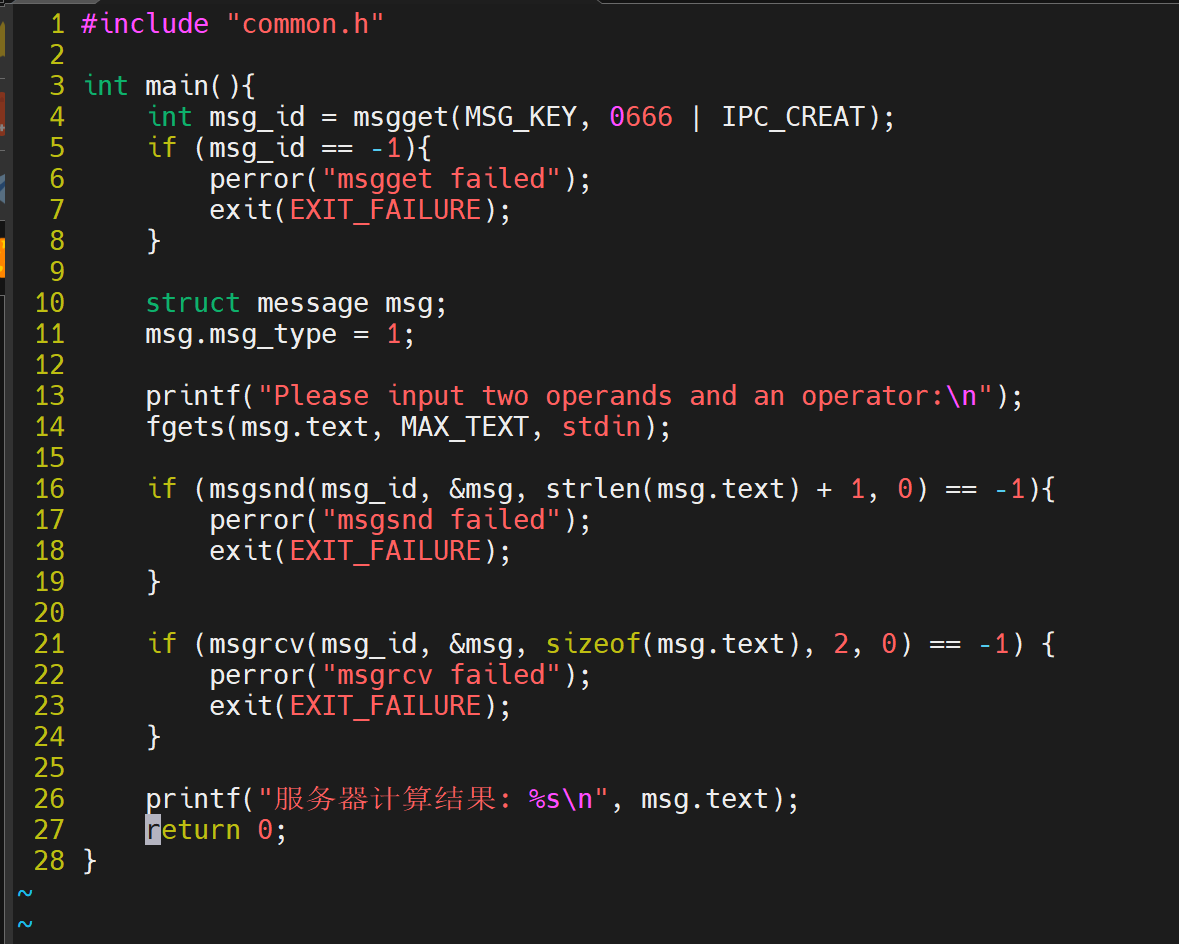
熟悉Linux消息队列机制，熟悉多任务并行机制的运作，了解不同进程利用通信来实现共同协作的方法

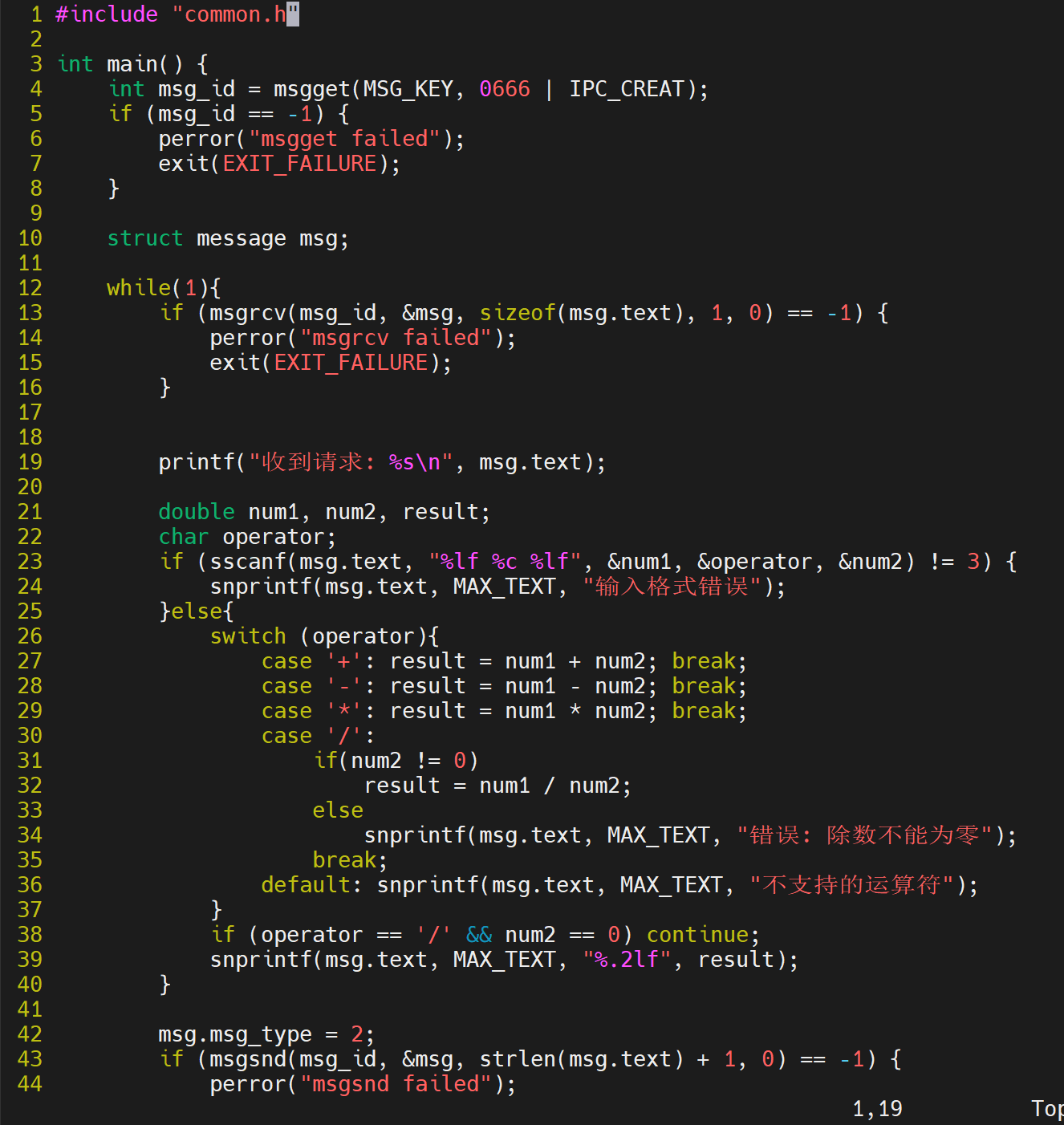
## 二、 实验要求

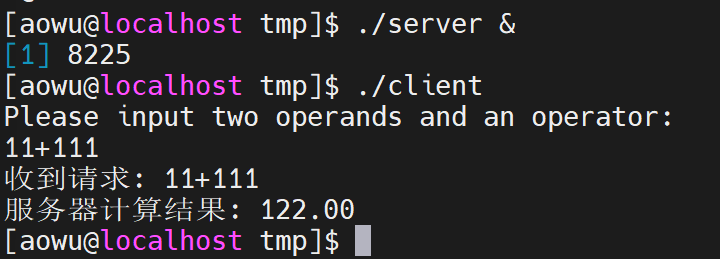
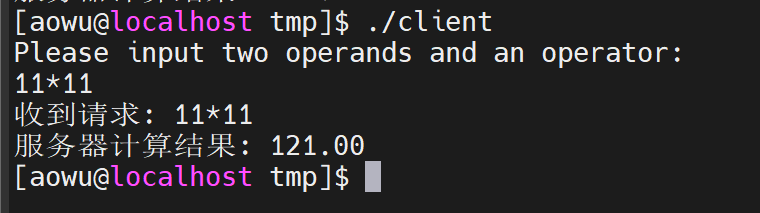
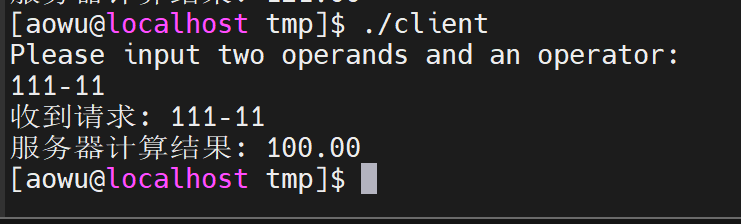
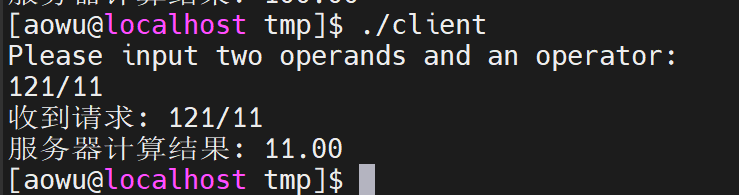
在Linux环境下设计、编译和运行代码，利用Linux的消息通信来分析和解决简单问题

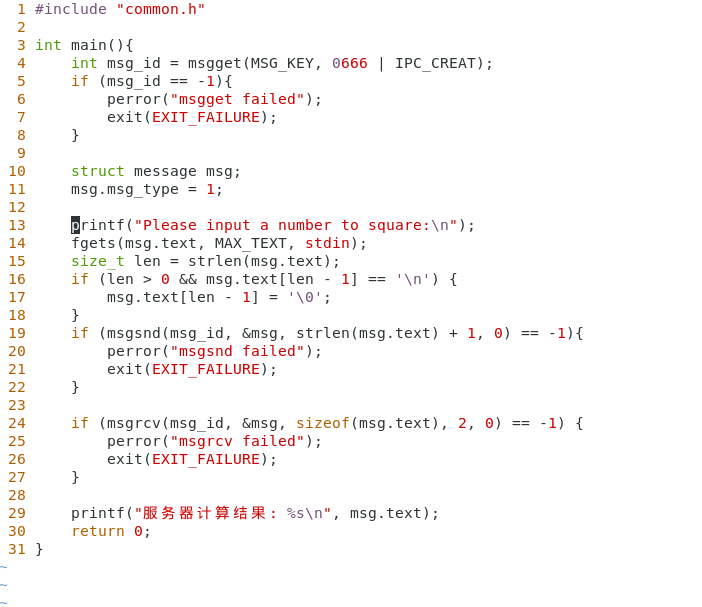
## 三、 实验任务

1. 两个进程利用通信协作机制实现简单的四则运算：

在这个任务中，一个四则运算的任务被分解成两个子任务：客户端（Client）和服务器（Server）。客户端程序实现人-机交互部分的代码，完成请求的提出和结果的处理，具体任务是从键盘上接收3个参数：2个操作数和1个运算操作符。然后，把这3个参数发送给服务器进程；服务器程序负责实现具体的数据处理部分的代码，具体任务接收客户进程的请求，并根据提供的运算操作符，对两个操作数进行运算，服务器端把运算的结果返回给客户端  
1、公共头文件common.h  
2、客户端程序  


3、服务器程序：  


4、运行截图：   

1. 自行设计两道程序，使两个进程利用通信机制实现任意的数学运算  
   实现输入数字，计算数字的平方
   1. 客户端程序：
   2. 服务端程序：
   3. 运行截图：