

# **《操作系统》课程作业**

|  |  |  |
| --- | --- | --- |
| **学 号：** | **16281006** | |
| **姓 名：** | **贾飞阳** | |
| **专 业：** | **计算机科学与技术** | |
| **学 院：** | **计算机与信息技术学院** | |
|  |  |  |
|  |  |  |
| **提交日期：** | **2019年03月11日** | |

# **实验一：操作系统初步**

## **作业题目：**

1.1 （系统调用实验）了解系统调用不同的封装形式。

（1）参考下列网址中的程序。阅读分别运行用API接口函数getpid()直接调用和汇编中断调用两种方式调用Linux操作系统的同一个系统调用getpid的程序(请问getpid的系统调用号是多少？linux系统调用的中断向量号是多少？)。

（2）上机完成习题1.13。

（3）阅读pintos操作系统源代码，画出系统调用实现的流程图。

1.2 （并发实验）根据以下代码完成下面的实验。

（1）编译运行该程序（cpu.c），观察输出结果，说明程序功能。

(编译命令： gcc -o cpu cpu.c –Wall)（执行命令：./cpu）

（2）再次按下面的运行并观察结果：执行命令：./cpu A & ; ./cpu B & ; ./cpu C & ; ./cpu D &程序cpu运行了几次？他们运行的顺序有何特点和规律？请结合操作系统的特征进行解释。

1.3 （内存分配实验）根据以下代码完成实验。

（1）阅读并编译运行该程序(mem.c)，观察输出结果，说明程序功能。(命令： gcc -o mem mem.c –Wall)

（2）再次按下面的命令运行并观察结果。两个分别运行的程序分配的内存地址是否相同？是否共享同一块物理内存区域？为什么？命令：./mem &; ./mem &

1.4 （共享的问题）根据以下代码完成实验。

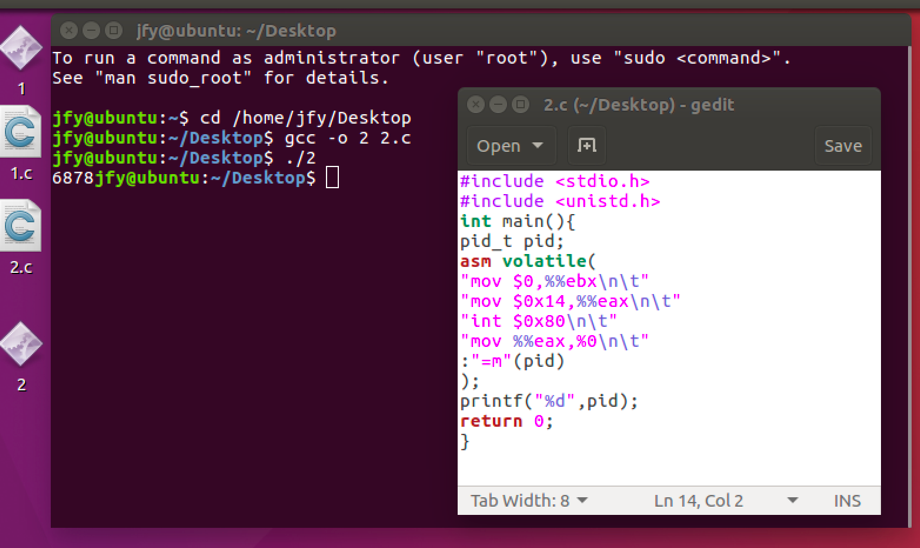
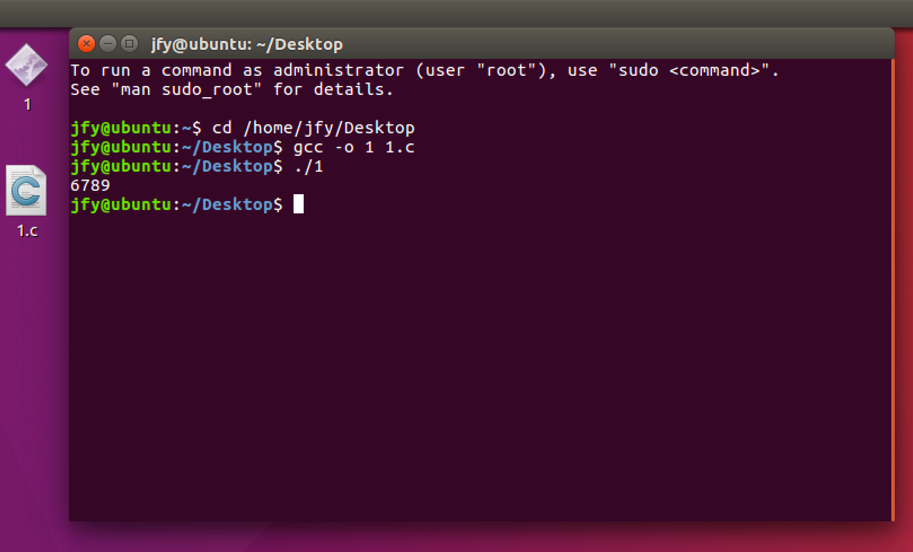
（1）阅读并编译运行该程序，观察输出结果，说明程序功能。（编译命令：gcc -o thread thread.c -Wall）（执行命令1：./thread 1000）

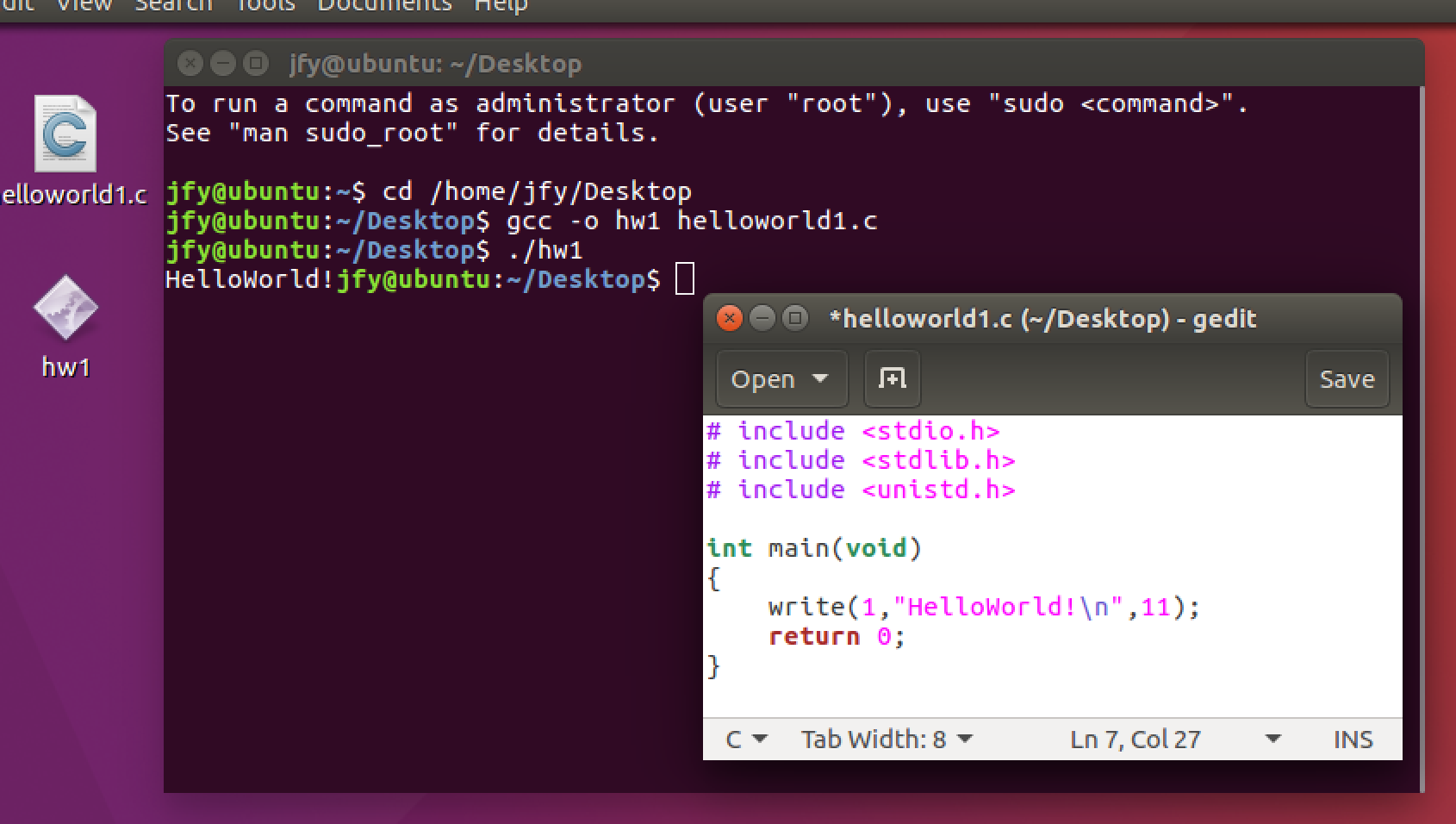
（2）尝试其他输入参数并执行，并总结执行结果的有何规律？你能尝试解释它吗？（例如执行命令2：./thread 100000）（或者其他参数。）

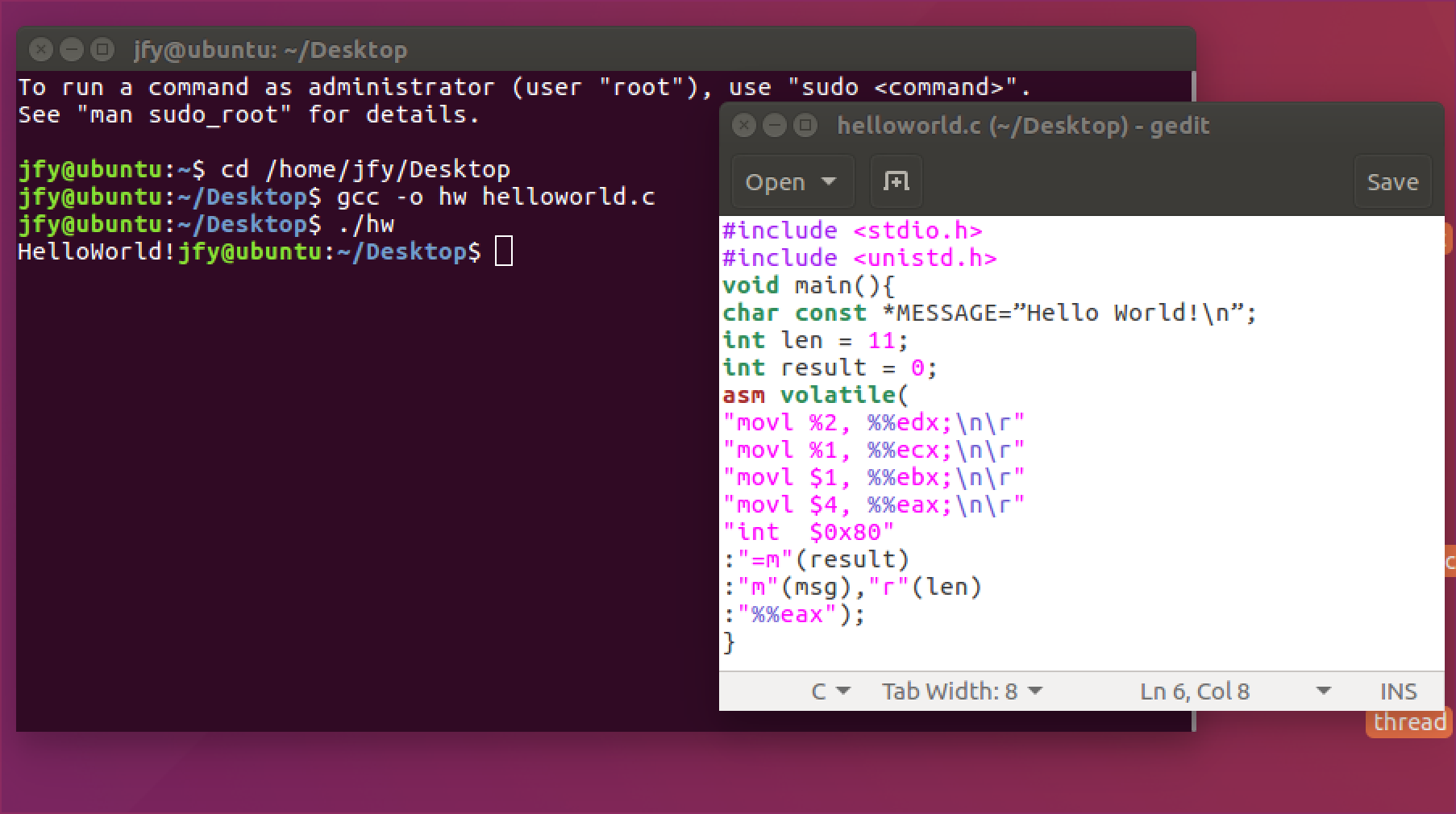
（3）提示：哪些变量是各个线程共享的，线程并发执行时访问共享变量会不会导致意想不到的问题。

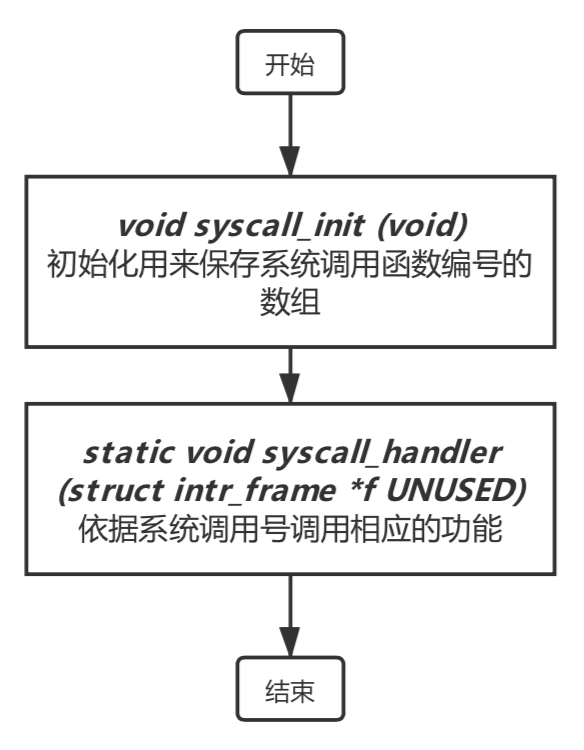
## **作业解答：**

### **第1.1题解答：**

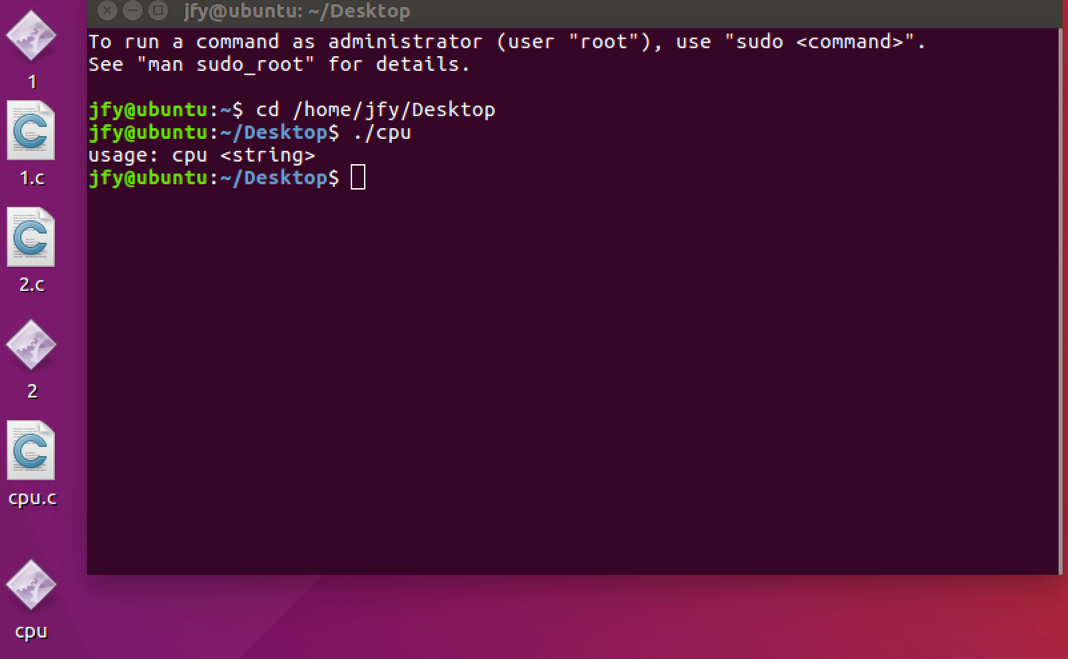
（1）从所给代码中可以看出getpid的系统调用号是0x14，linux系统调用的中断向量号是0x80，两种调用方式如下所示：

（2）两种调用方式如下。其中，linux的系统调用使用write函数。此函数有三个参数，第一个参数为1时是控制台输出。



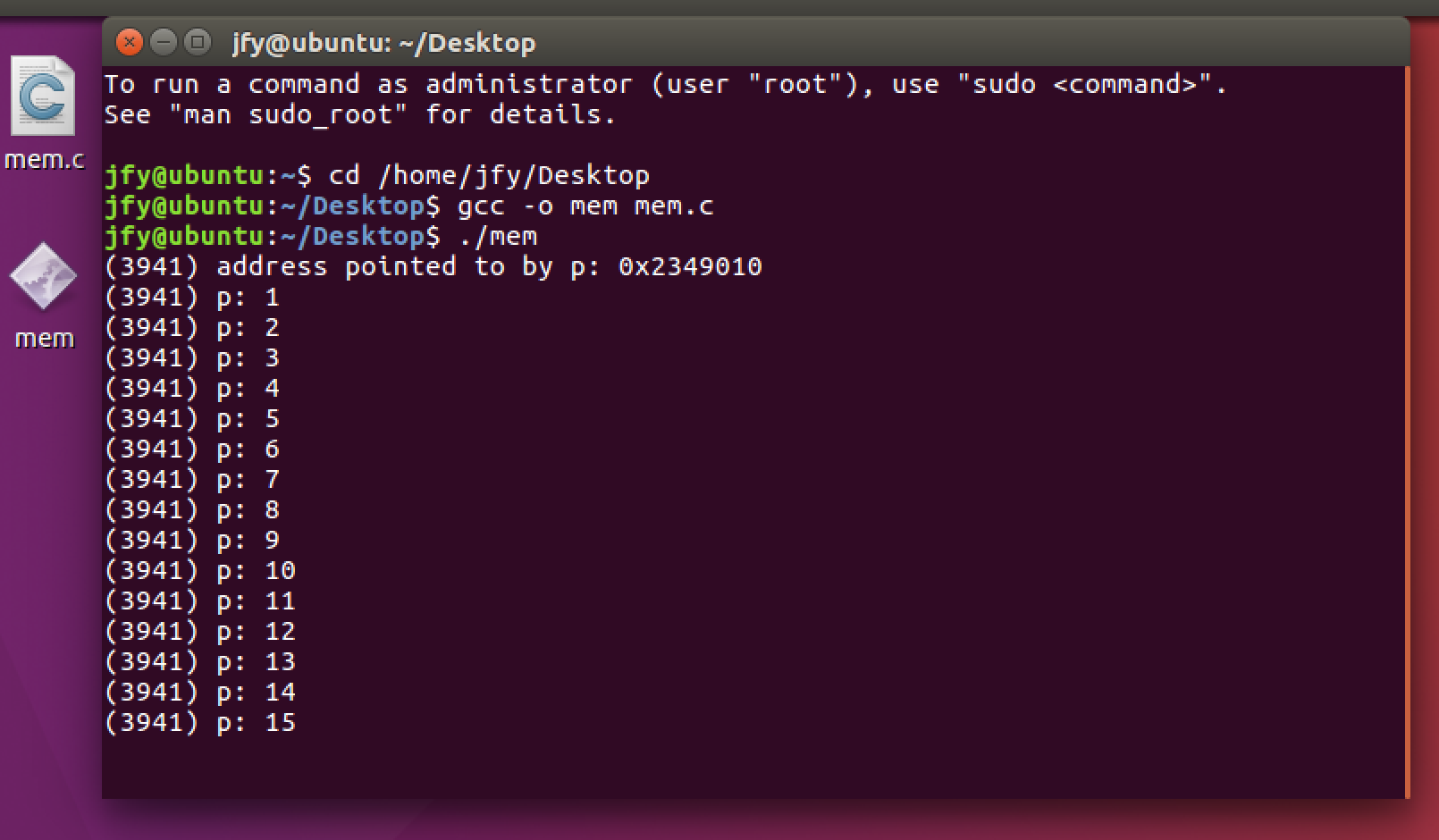
（3）通过阅读../userprog/syscall.c文件，可知pintos操作系统的系统调用的实现如下：

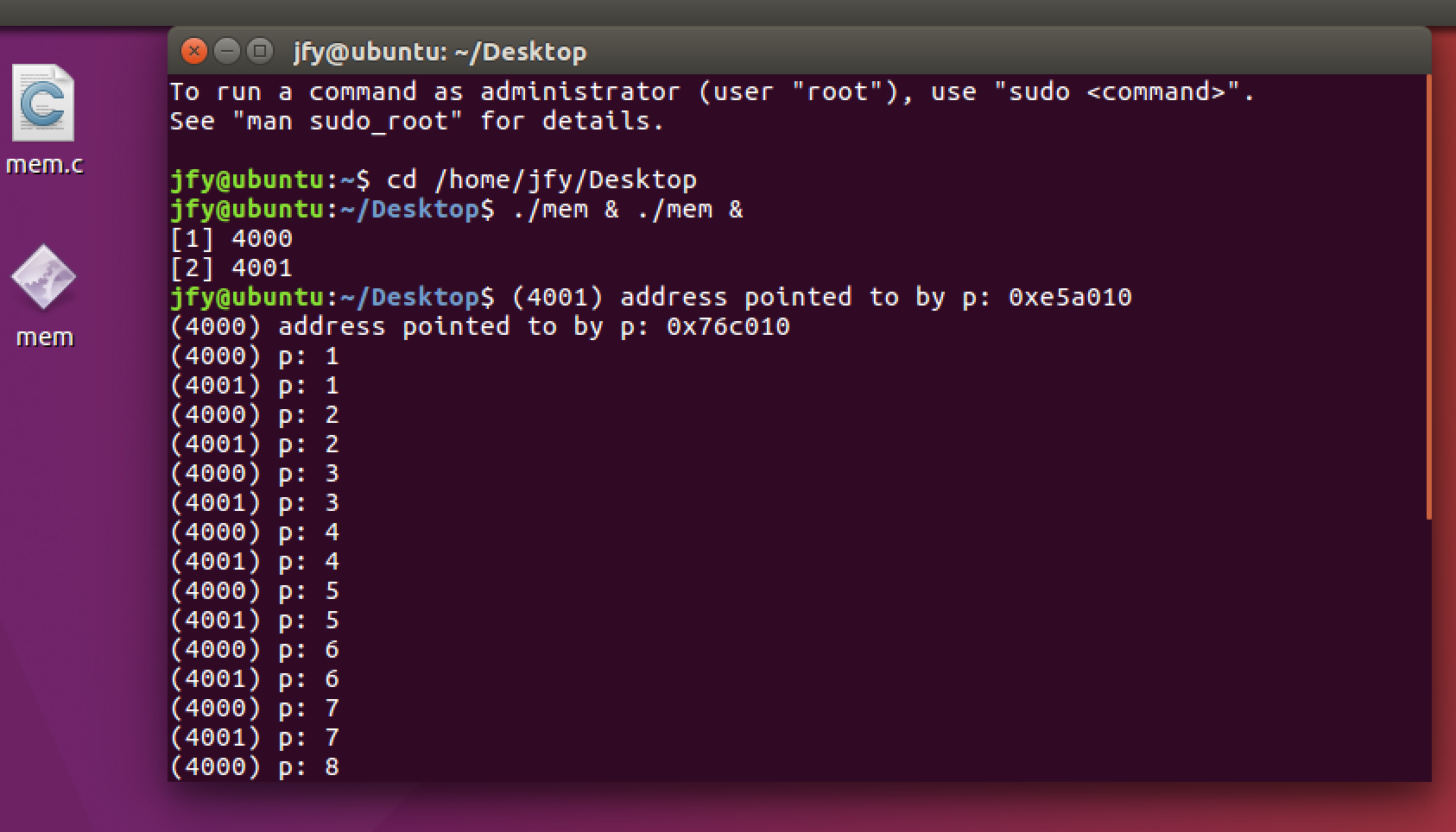
### **第1.2题解答：**

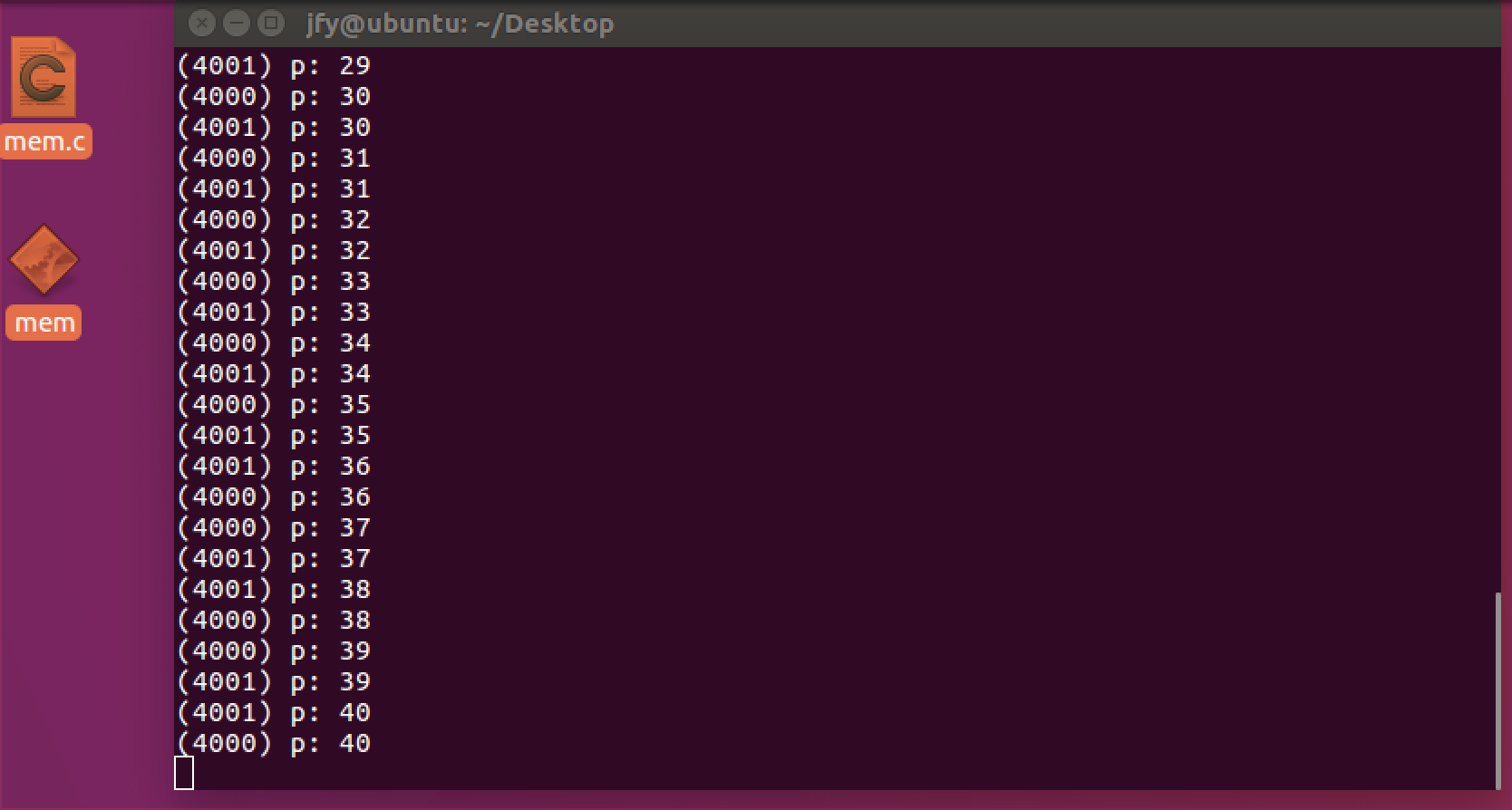
（1）进行如题所示的编译命令和执行命令，结果如下。根据代码和输出的信息可以看出，这个程序的作用是输出命令行参数中的内容。如果输入的参数个数为0（代码中设置的条件为不等于2，因为函数的名称在是命令行参数中的第一个，用户参数从第二个算起），则输出提示信息。

（2）进行如题所示的编译命令和执行命令，结果如下。可以明显的看到输出的顺序与输入参数的顺序（ABCD）是不一致的。且这个不一致是变化的，并不仅是除了ABCD这个顺序之外的单个顺序。四个程序在宏观上是一同运行的，微观上还是在cpu交替运行的，他们的优先级相同，因此先进行的程序不一定是先结束，这就造成了这四个程序同时运行的时候，结束的顺序是不确定的，那么输出顺序也是不一定的。

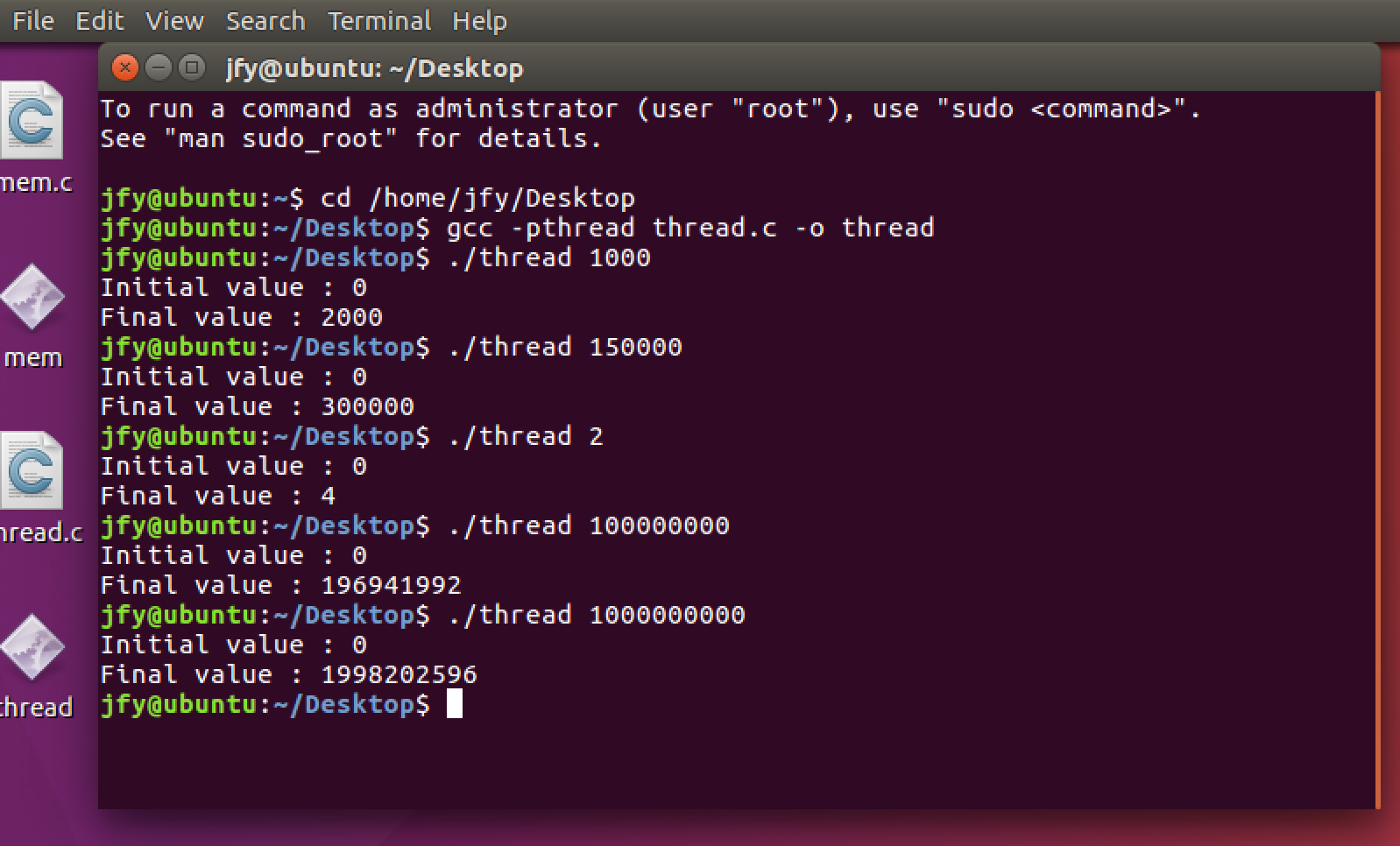
**第1.3题解答：**

（1）进行如题所示的编译命令和执行命令，结果如下。该程序做了如下几件事：程序分配了内存并打印出内存地址，然后将0放入内存的第一个位置。在一个无限循环中，延迟一秒并递增存储在p中保存的地址的值。对于每个print语句，它还会打印出正在运行的程序的进程标识符（每个运行过程中都唯一）。

（2）进行如题所示的编译命令和执行命令，结果如下。可以从运行结果看到，两个独立运行的程序并不共享一块物理地址。这是因为操作系统虚拟化了内存。每个进程访问自己的私有虚拟地址空间，操作系统以某种方式映射到机器的物理内存。一个正在运行的程序中的内存引用不会影响其他进程（或OS本身）的地址空间。



### **第1.4题解答：**

（1）进行如题所示的编译命令和执行命令，结果如下。调用时输入的参数为1000。程序使用Pthread .create()创建两个线程，每个线程在worker（）的例程中运行，该函数的作用是循环递增的计数器，计数区间为1。当两个线程完成时，计数器的最终值为2000，因为每个线程将计数器递1000次。 所以可以推断当循环的输入值设置为N时，程序的期望输出为2N。

（2）在以上解释的基础下，增大输入的参数，会发现得到的输入不再是2N。原因是：计数器递增，需要三个指令：计数器的值从存储器加载到寄存器中、递增、存储回内存。这意味着这三条指令不是同时执行的，所以这“一组”操作与其他操作之间的执行顺序是不固定的。而且countor、loops这两个全局变量是被这两个线程共享的。当两个的线程一起被调用执行的时候，共享全局变量countor，在编写多线程的程序时，同一个变量可能被多个线程修改，而程序通过该变量同步各个线程。所以当参数变得很大时，就会因为指令调用顺序和参数共享的缘故导致错误。