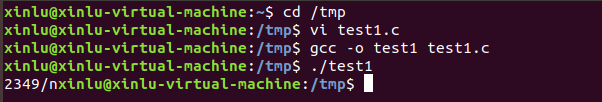
**操作系统实验一：操作系统初步**

16281259 鲁鑫

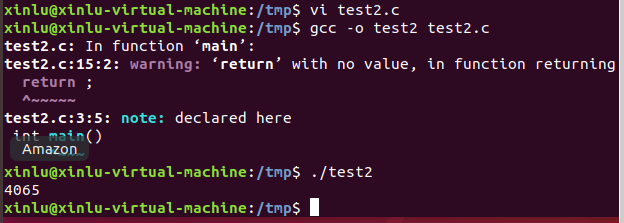
# 一、系统调用实验

## 1. API接口函数getpid()直接调用



可以看到getpid的系统调用号是2349

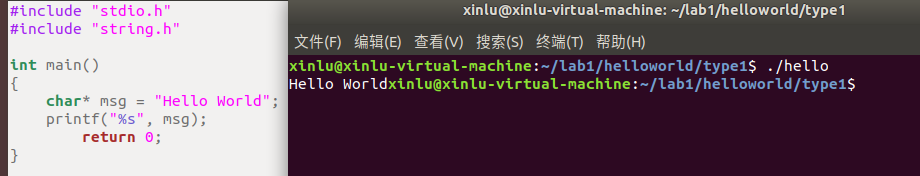
## 汇编中断调用



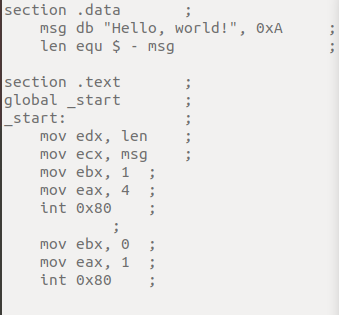
可以看到linux系统调用的中断向量号是4065

## 2. 上机完成习题1.13.3

Linux系统下的c语言实现



汇编代码实现



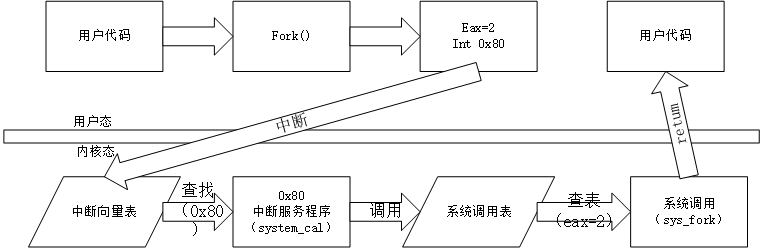


因为没有学过汇编，所以这段代码是从网上搜索的，参照网上的操作过程，nasm -f elf64 -o helloworld.0 helloworld.asm命令为编译这段汇编代码；

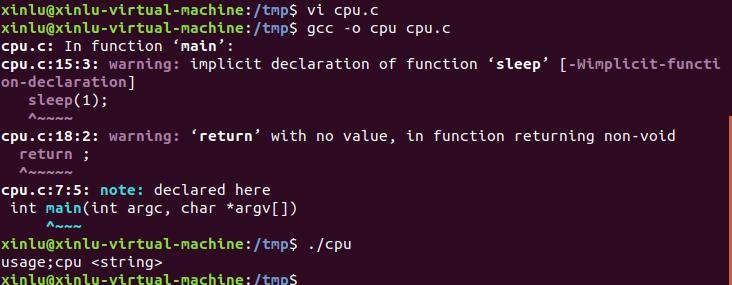
Ld -o helloworld helloworld.0 为链接；

./helloworld为运行。

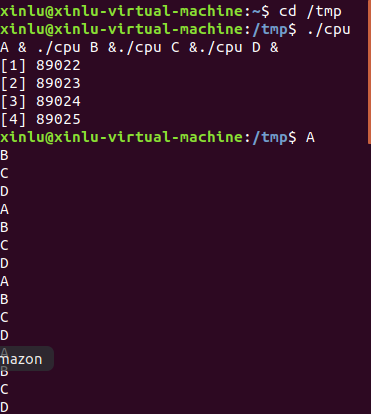
## 3.画出系统调用实现的流程图



# 二、并发实验

1. 编译运行该程序（cpu.c），观察输出结果，说明程序功能

## 2.



程序cpu一直在运行。运行的顺序为谁A B C D，也就是按照请求cpu的顺序依次进行。按照操作系统的并发性，在一段时间内，可以有多个程序运行。

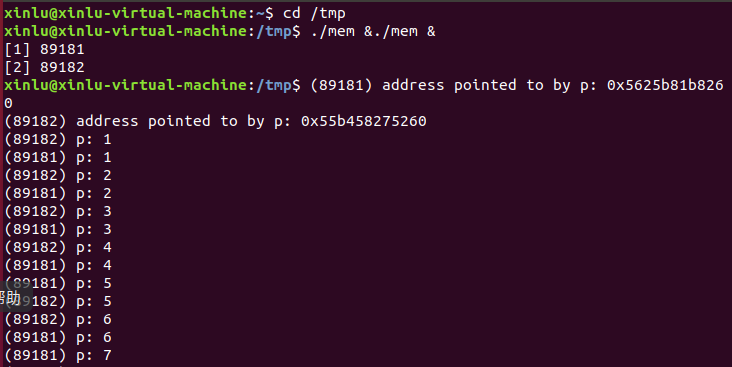
# 三、内存分配实验

## 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



程序的功能是输出一个系统调用号对应的程序的运行结果，即循环输出一个指针对应系统分配的内存地址内存储的数据。

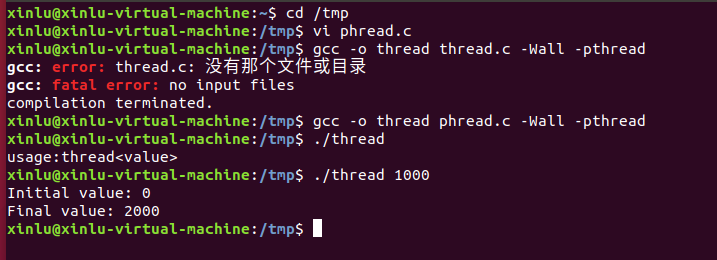
## 2. 再次按下面的命令运行并观察结果



两个分别运行的程序分配的内存地址不相同，但是共享一块物理内存区域，因为这两个程序是交替执行的，说明二者分配的物理内存区域是相同，系统按照两个程序对物理内存区域的请求依次调度。

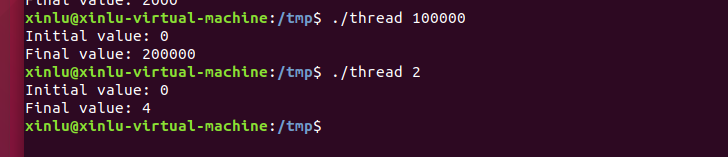
# 四、共享的问题

## 1. 阅读并编译运行该程序，观察输出结果，说明程序功能



从运行结果可以看到程序的功能是把输入程序的值\*2，然后输出。

2.



3.loops,counter变量是各个线程共享的，不会导致意想不到的问题。