**实验一: 操作系统初步**

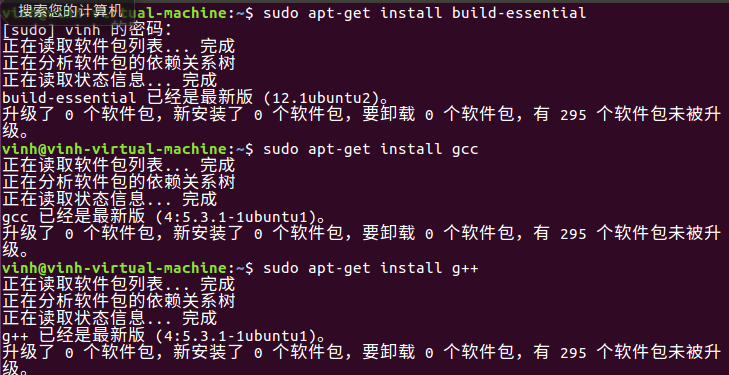
苏荣天

16281141

安全1601

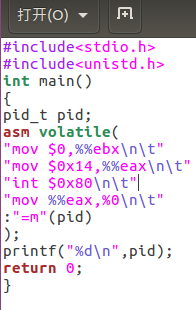
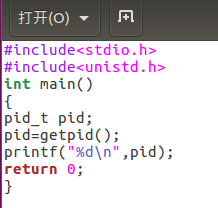
一、（系统调用实验）了解系统调用不同的封装形式。

要求：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？)。



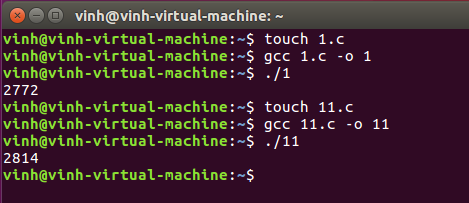
检查必要的软件包。

编译两段代码。



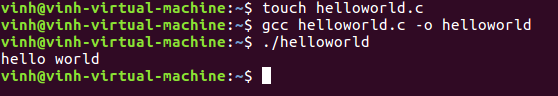
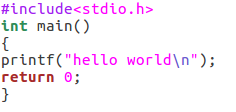
执行结果如下：

getpid的系统调用号是20，中断向量号80H，系统调用号14H。

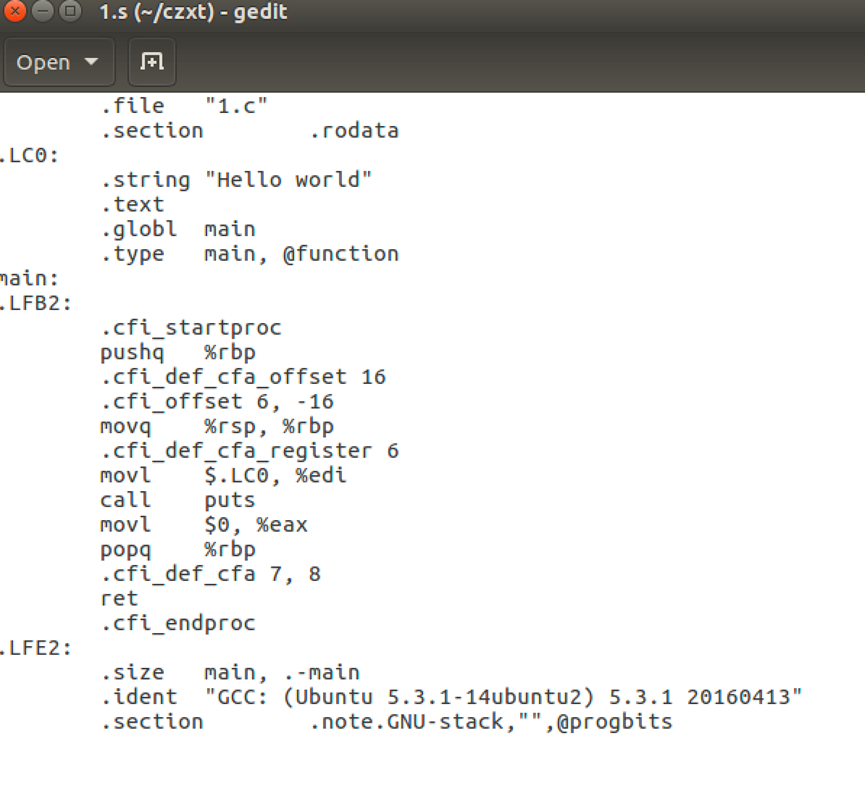


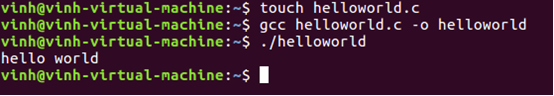
2、上机完成习题1.13。

C:

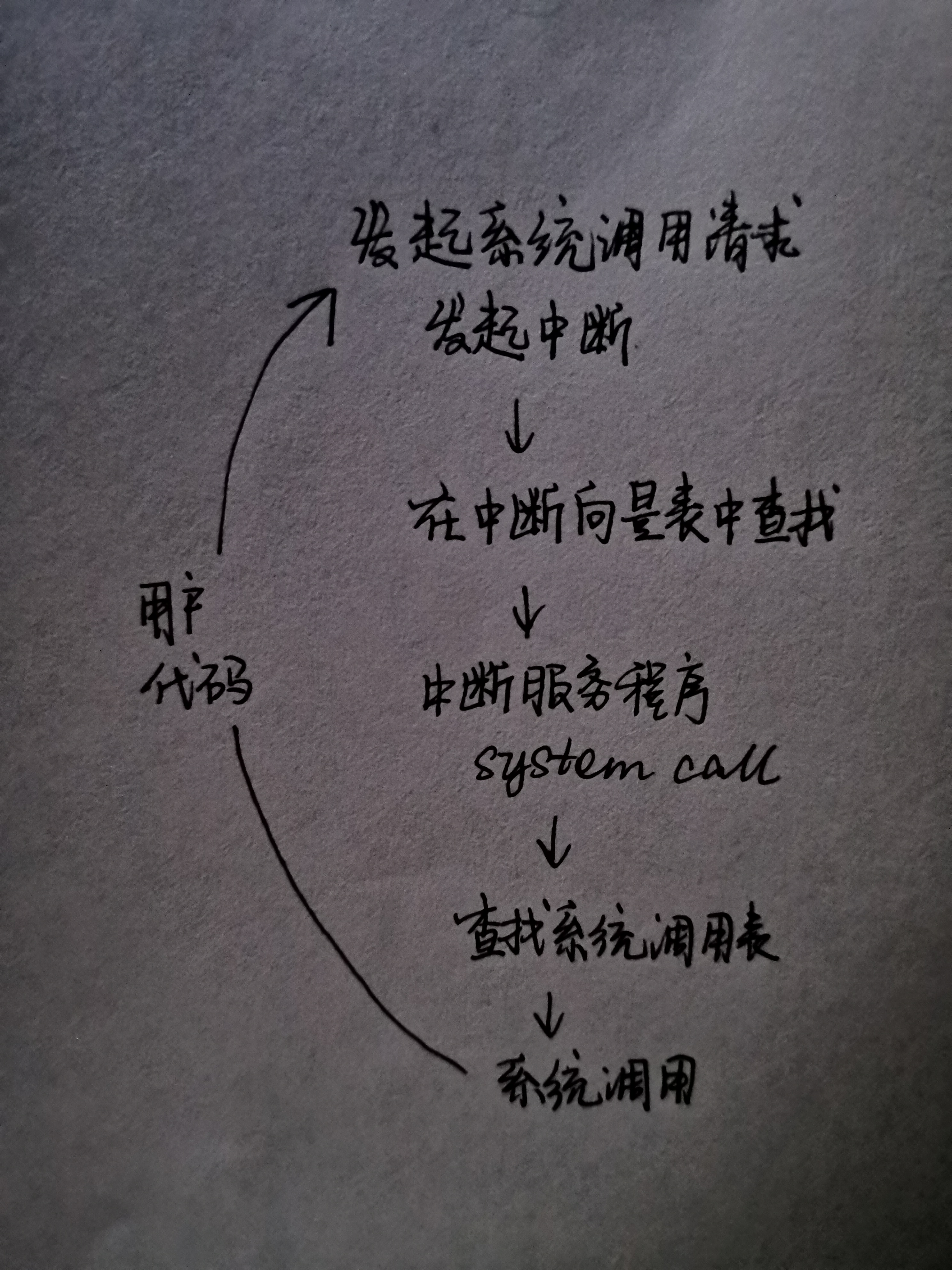


汇编：





3、阅读pintos操作系统源代码，画出系统调用实现的流程图。



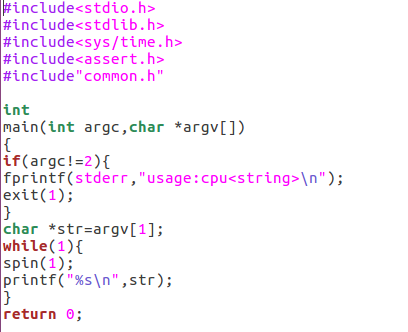
二、（并发实验）根据以下代码完成下面的实验。

要求：

1、 编译运行该程序（cpu.c），观察输出结果，说明程序功能。

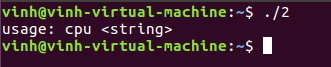
(编译命令： gcc -o cpu cpu.c –Wall)（执行命令：./cpu）

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。



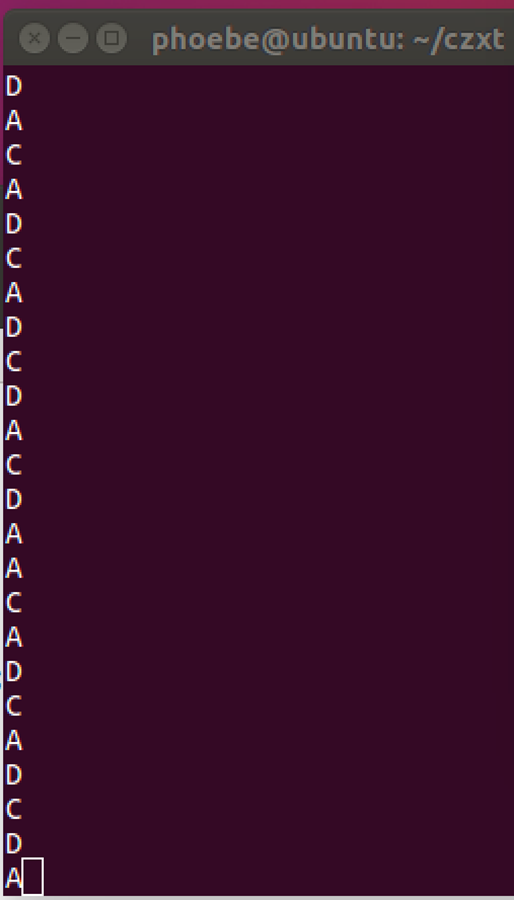
1.程序功能为每隔一秒输出一次参数

若传入参数正确，则输出；若传入参数不正确，则输出错误提示usage: cpu <string>。



2. 对于4个完全相同的程序，CPU的优先级是相同的，因此会随机地顺序执行。并发是指一个时间段中有几个程序都处于已启动运行到运行完毕之间，且这几个程序都是在同一个处理机上运行，但任一个时刻点上只有一个程序在处理机上运行。





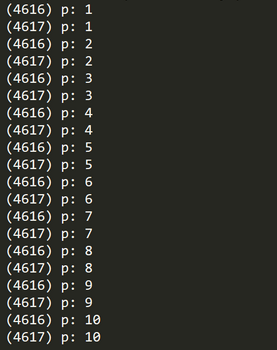
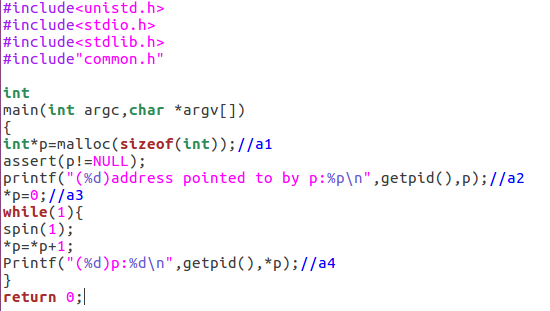
三、（内存分配实验）根据以下代码完成实验。

要求：

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.



程序功能为分配内存，打印指针p指向的内存地址，以及每隔一秒将指针加一后打印指针的相对地址

2. 两次指针p指向的地址不相同，不是同一内存区域。

操作系统为不同的指针分配不同的地址，同时为了避免指针之间的冲突，所以不能把不同的指针指向同一内存区域。

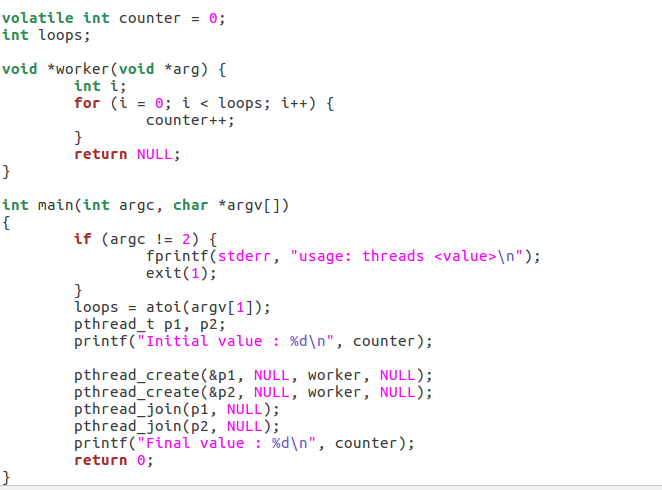
四、（共享的问题）根据以下代码完成实验。

要求：

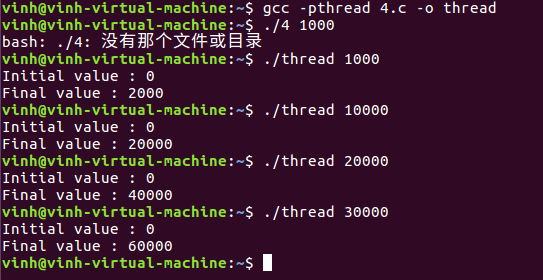
1、 阅读并编译运行该程序，观察输出结果，说明程序功能。（编译命令：gcc -o thread thread.c -Wall –pthread）（执行命令1：./thread 1000）

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。



执行结果：



可见终值与初值的差为输入参数的两倍。虽然开了两个线程，但是两个线程均对同一变量进行加一操作，参数即为循环的次数，所以为输入参数的两倍。

3. counter和loops是各个线程共享的。两个线程同时访问可能会导致数据的丢失，导致两次自增表现为一次自增