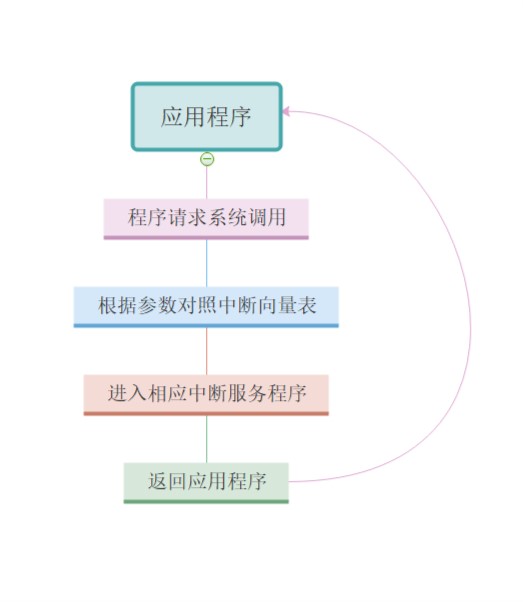
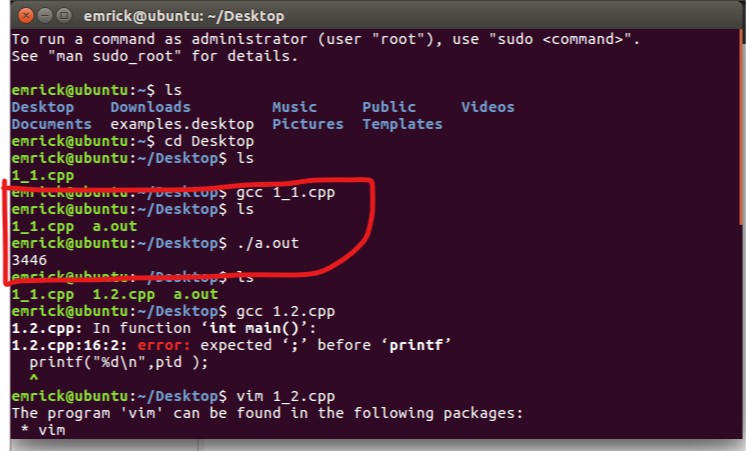
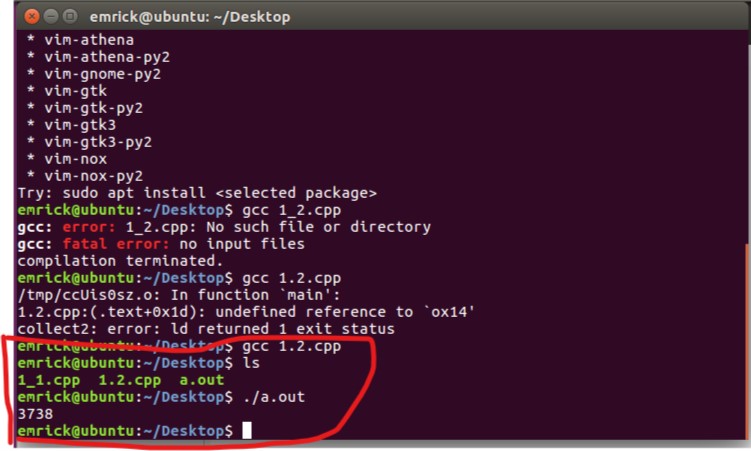
一、（系统调用实验）了解系统调用不同的封装形式。

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？)。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. getpid的系统调用号是 14H、中断向量号是80H。
2. 

实验截图：



二、（并发实验）根据以下代码完成下面的实验。

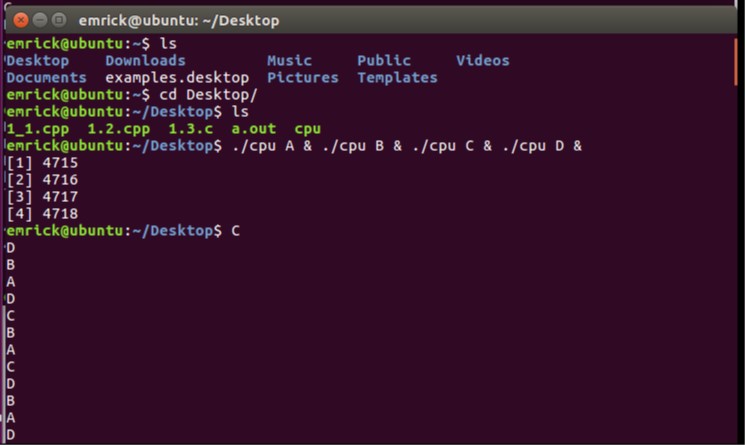
要求：

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：循环输出传入参数到屏幕，如果无参数则输出usage:cpu<string>。
2. 运行结果：



只要不手动停止，CPU就一直运行，运行的特点：可以看到程序运行的次序是不确定的，原因可能是Ubuntu系统是支持并行的，因此我们的四个程序在CPU内同时运行，但速度有快有慢，因此才会有这样的输出结果。

三、（内存分配实验）根据以下代码完成实验。

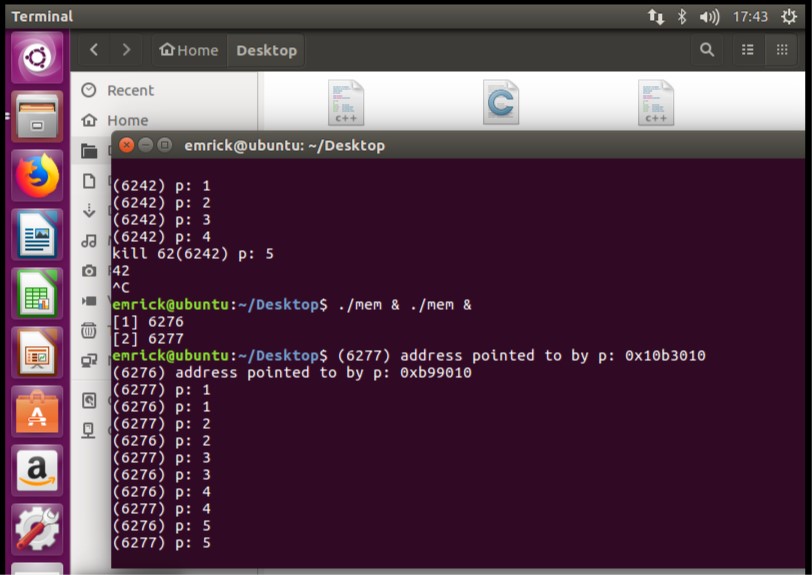
要求：

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. 程序的功能是新建进程并打印进程识别码：

输出结果：



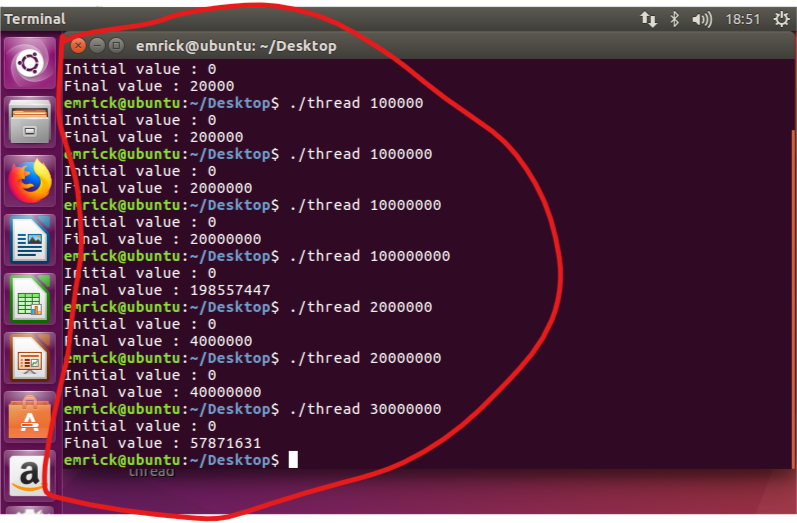
1. 内存地址不相同，不共享同一物理内存区域，因为在同一物理内存区的两个进程识别码不可能相同，他们可能是运行在不同内核里面的程序。

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. 程序作用：创建两个线程对counter进行累加操作，并输出初始值和累加后的值：

运行截图：



2、可以看到当参数大道一定程度时，输出的值就不再是传进参数的2倍了，可能是因为CPU处理的过程时间太长，两个线程的运行发生了时间上的重合，其中一个线程在进行累加操作时，另一个线程读入了过时的共享变量counter的数据，造成了累加值不足的情况。