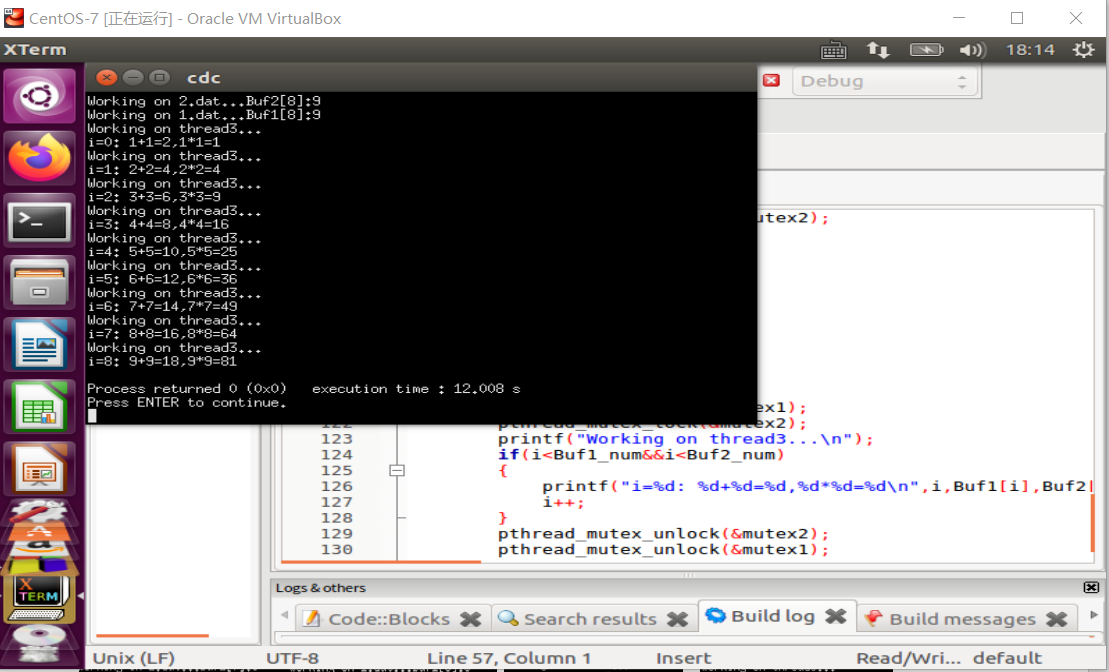
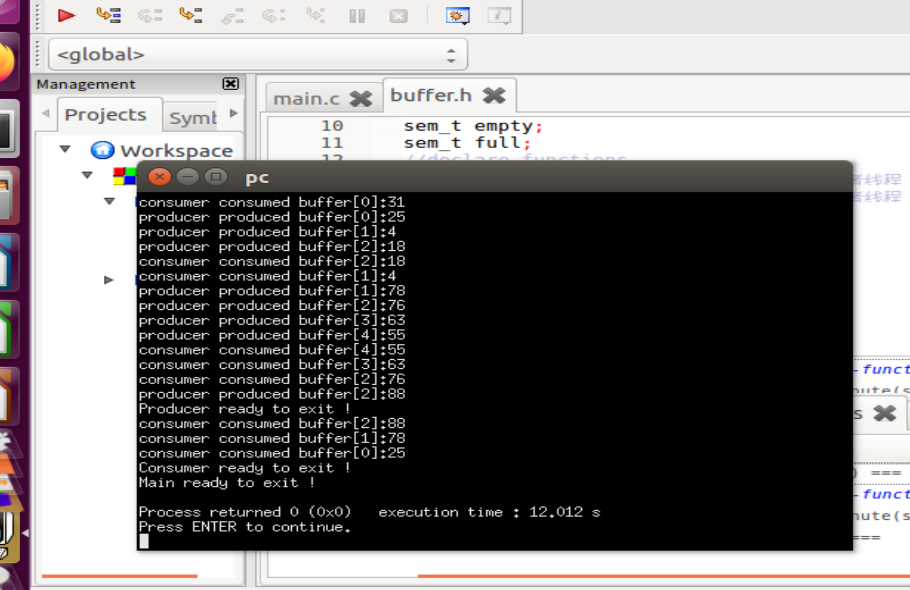
1.cdc



分析：由于编译器和电脑环境的不同，在不同的环境下，会出现线程3一直抢占到资源的情况，从而出现死锁。

2.pc



1. 请说出练习a,c的相同和不同之处（此处指未移除临界区互斥保护的程序c）。

相同之处：都是在不同的线程中对共享的数据进行处理，都是生产者消费者问题，需要等待生产者生产出数据内容之后消费者才可以使用数据

不同之处：练习a对某一个数据块只会进行一次读取，而练习c会对一个地方的数据反复读取。练习c取消了数据临界区，使用信号量实现互斥。而练习a对于共享数据的操作位于临界区内，防止线程中操作同时刻交叉

2.请说出练习b,c的相同和不同之处（此处指未移除临界区互斥保护的程序b和c）。

相同之处：均通过线程互斥锁实现了多线程数据临界区的保护，同时也避免了死锁的情况出现（每次执行consumer函数必定在producer函数执行之后执行）

不同之处：练习b中数组有5个空槽，empty一开始设置为5，并且必须借助mutex才能实现互斥。而练习c中将空槽设为1，empty一开始为1，不需要借助mutex也一样能够实现互斥

3.请解释练习b“分析与总结探索” 移除临界区互斥后为什么会出现不正常的现象?

因为当取消临界区以后，在insert函数中，在index变量改变后添加了一句sleep（1），此时程序挂起去执行remove函数，由于缺少了临界区的保护，index会一直减少至小于0，因此会打印出错误的信息。

4.你是否推荐使用二元信号量来代替互斥锁？为什么？

当只有一个插槽的情况，二元信号量在功能上可以替代mutex，但是在诸如练习b这样的情况，有多个插槽的情况，必须借助mutex=1来实现互斥，因此不是所有的情况都可以替代。