周心瑜 181870290

181870290@smail.nju.edu.cn

实验四 实验报告

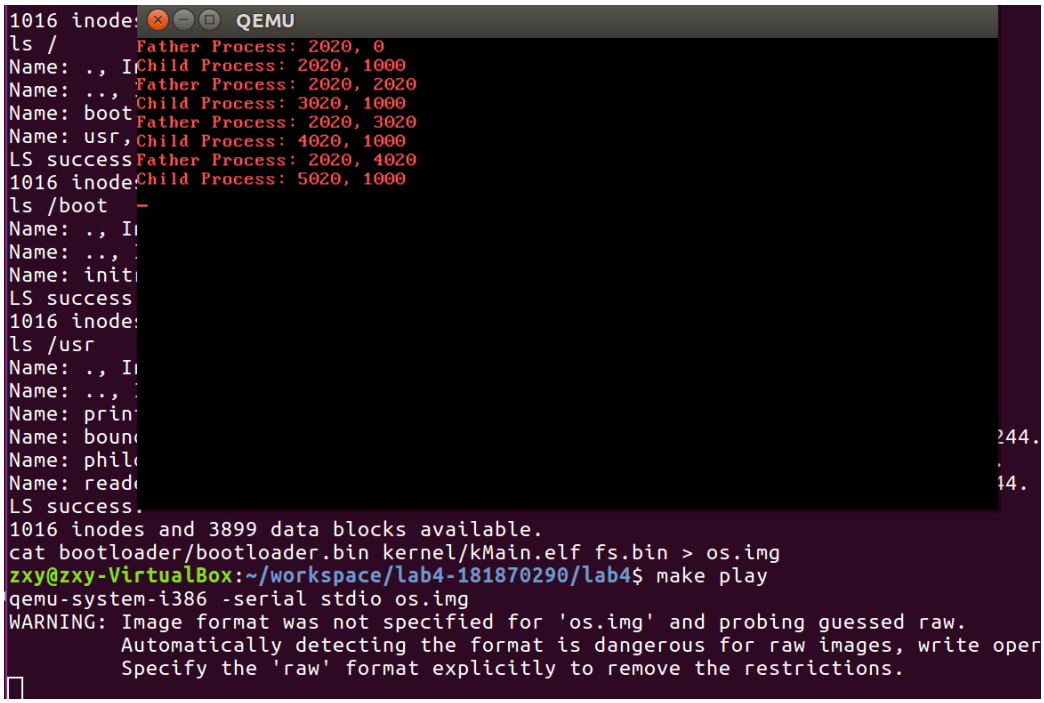
一、实验进度

完成了所有必做任务。

二、实验结果

2.1 实现格式化输入函数

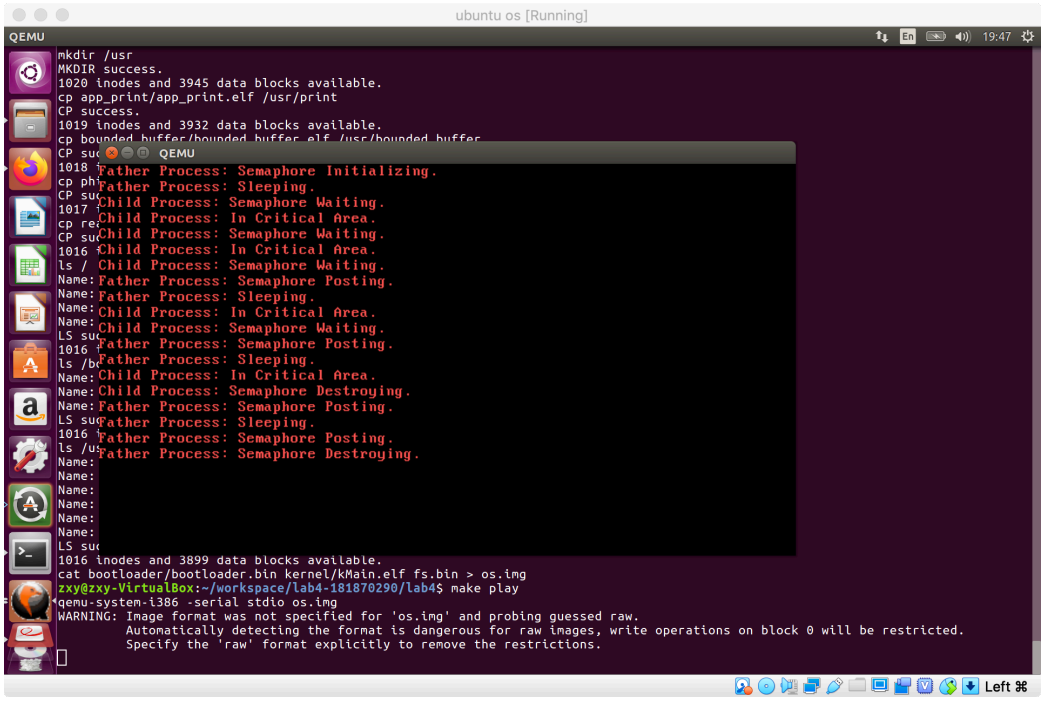
2.2 实现进程通信



周心瑜 181870290

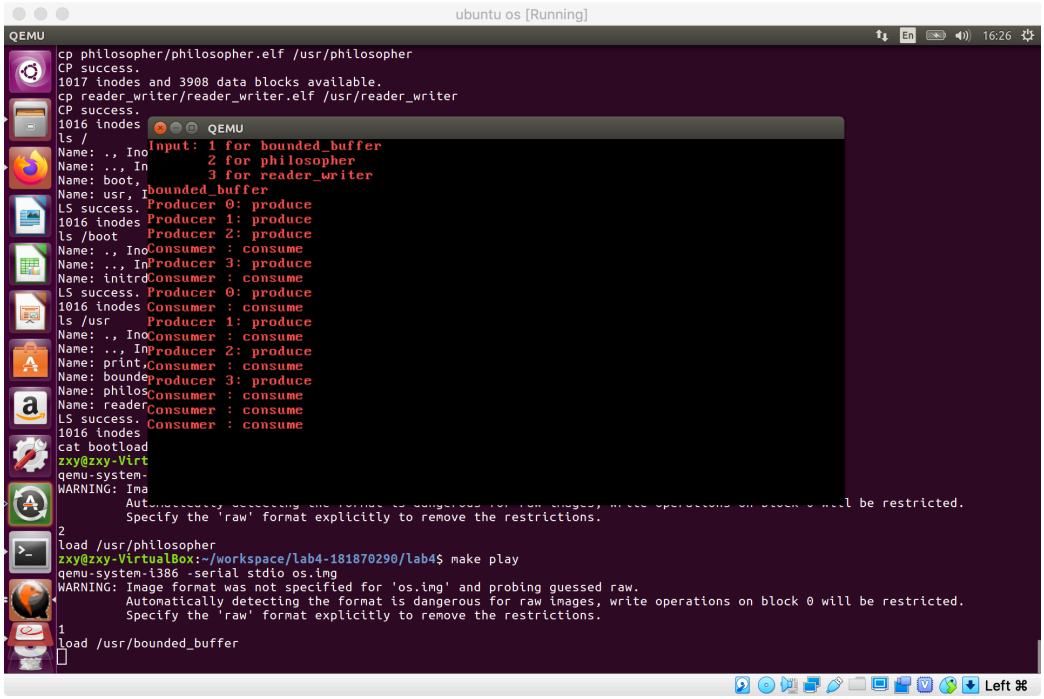
181870290@smail.nju.edu.cn

2.3 实现信号量相关系统调用



2.4 基于信号量解决进程同步问题

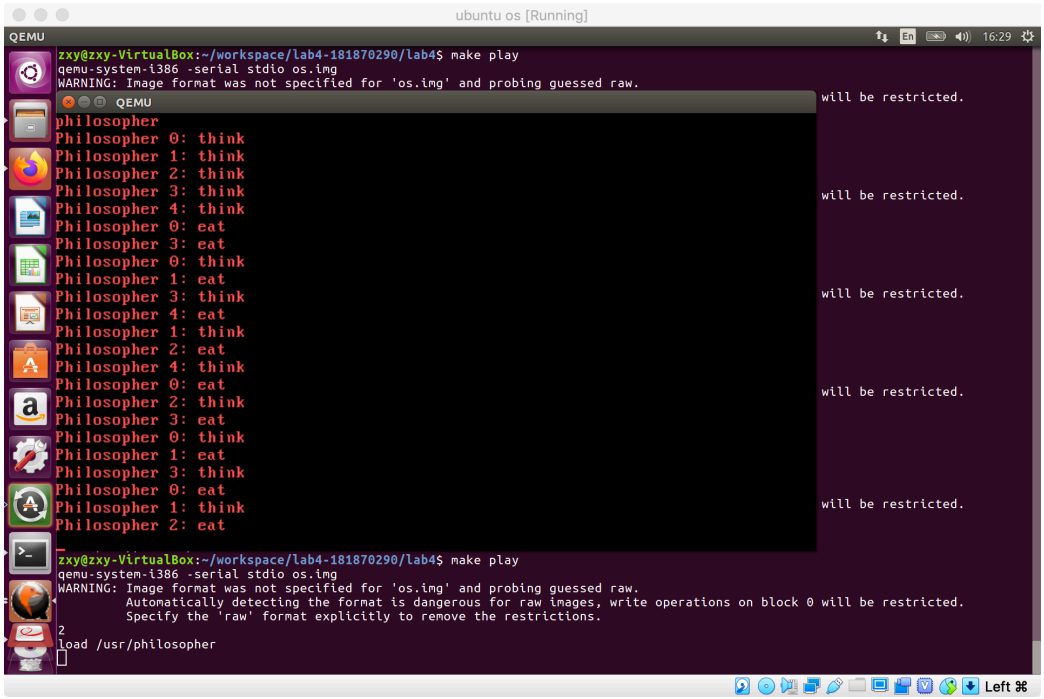
生产者-消费者问题:



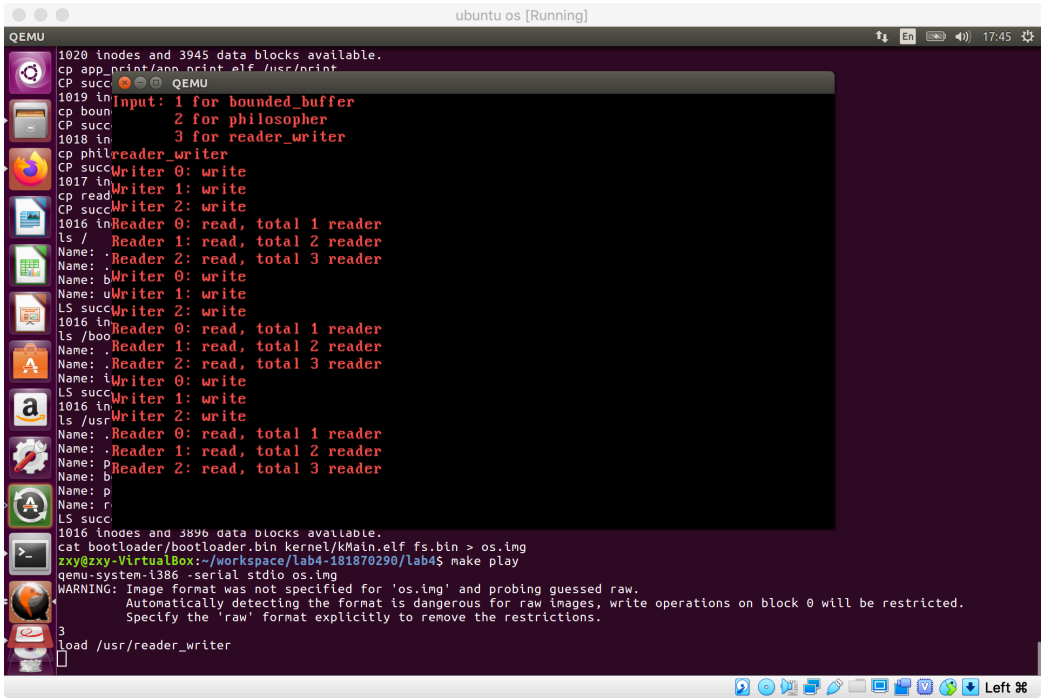
周心瑜 181870290

181870290@smail.nju.edu.cn

哲学家就餐问题：



读者-写者问题：



周心瑜 181870290

181870290@smail.nju.edu.cn

三、实验代码

3.1实现格式化输入函数

代码修改：kernel/kernel/irqHandle.c

实现函数：

keyboardHandle 将键盘输入存入缓冲区。在接收到合法的键码后将keyCode放入keyBuffer，并在设备STD_IN上被阻塞的进程中取出一个进程pt唤醒。代码结构类似于V操作。

syscallReadStdIn 从缓存区中读取数据。当设备STD_IN未被阻塞时，syscallReadStdIn将阻塞当前进程。在进程被唤醒时读keyBuffer中所有数据。代码结构类似于P操作。

3.2 实现进程通信

代码修改：kernel/kernel/irqHandle.c

实现函数：

syscallWriteShMem / syscallReadShMem 例程处理。根据syscall.c中封装函数read/write于syscall的参数接口获取读写内容的buffer起始、size与index。参考irqHandle.c中的读写对共享内存进行读写。

3.3 实现信号量相关系统调用

代码修改：kernel/kernel/irqHandle.c

实现函数：

syscallSemInit 子例程。根据syscall.c中封装函数，选择一个未使用的信号量初始化并返回该信号量的编号。

syscallSemWait 子例程。如果value<0则阻塞进程。

syscallSemPost 子例程。当value<=时，唤醒一个被阻塞的进程。

syscallSemDestroy 子例程。将给定编号的信息量状态设为未使用。

3.4 解决进程同步问题

生产者-消费者问题

代码修改：bounded_buffer/main.c

选择参数：缓冲区最大限制buffer = 3，轮次 value = 2

哲学家就餐问题

代码修改：philosopher/main.c

选择参数： 哲学家用餐人数 = 5，轮次无限

读者-写者问题

代码修改：reader_writer/main.c

选择参数： 读者人数 = 写者人数 =3，轮次无限

四、实验中遇到的问题以及解决方法

1. 在实现格式化输入函数时，无法正确显示完全字符或是显示空字符。问题在于在判断keycode是否合法时只把keycode == 0 判断为非法，使得一些非法keycode在getChar之后转化为'\0'使得读取buffer时的字符串不正确。

2. 在解决读者-写者问题时，在第一个reader后卡住了。发现问题时Rcount没有存储在共享内存空间中，导致多个读者申请占用时Rcount均为0，使得信息量CountMutex被占用而无法释放WriteMutex造成死锁。参考实现进程通讯部分解决问题。

六、实验心得

通过对代码结构的理解，读懂每一个程序的含义及其起到的作用是顺利完成实验的关键。实验中需要对实验指南进行研读，并能积极的对一些功能进行实践，保持良好学习心态，这是健康的学习成长经历。