

操作系统实验报告

学号：201220139

姓名：艾冷州

邮箱：201220139@smail.nju.edu.cn

日期：2022年5月16日

1. Exercises

1. 请回答一下，什么情况下会出现死锁。

```
#define N 5                // 哲学家个数
semaphore fork[5];        // 信号量初值为1
void philosopher(int i){  // 哲学家编号：0-4
    while(TRUE){
        think();           // 哲学家在思考
        P(fork[i]);        // 去拿左边的叉子
        P(fork[(i+1)%N]);  // 去拿右边的叉子
        eat();             // 吃面条
        V(fork[i]);        // 放下左边的叉子
        V(fork[(i+1)%N]);  // 放下右边的叉子
    }
}
```

答：当所有哲学家都拿起了左边（或右边）的叉子时，他们都将进入等待右边（或左边）的叉子放下的状态，此时将出现死锁。

2. 说一下该方案有什么不足？（答出一点即可）

```
#define N 5                // 哲学家个数
semaphore fork[5];        // 信号量初值为1
semaphore mutex;          // 互斥信号量，初值1
void philosopher(int i){  // 哲学家编号：0-4
    while(TRUE){
        think();           // 哲学家在思考
        P(mutex);          // 进入临界区
        P(fork[i]);        // 去拿左边的叉子
        P(fork[(i+1)%N]);  // 去拿右边的叉子
        eat();             // 吃面条
        V(fork[i]);        // 放下左边的叉子
        V(fork[(i+1)%N]);  // 放下右边的叉子
        V(mutex);          // 退出临界区
    }
}
```

答：当其中一名哲学家在吃面条时，程序将一直处于临界区状态，其他结束了思考的哲学家将无法去取身边的叉子。

3. 正确且高效的解法有很多，请你利用信号量 PV 操作设计一种正确且相对高效（比方案 2 高效）的哲学家吃饭算法。

答：如下，在哲学家开始吃面条时退出临界区即可。

```
#define N 5           // 哲学家个数
semaphore fork[5];    // 信号量初值为1
semaphore mutex;       // 互斥信号量，初值1
void philosopher(int i){ // 哲学家编号：0-4
    while(TRUE){
        think();        // 哲学家在思考
        P(mutex);        // 进入临界区
        P(fork[i]);      // 去拿左边的叉子
        P(fork[(i+1)%N]); // 去拿右边的叉子
        V(mutex);        // 退出临界区
        eat();            // 吃面条
        V(fork[i]);       // 放下左边的叉子
        V(fork[(i+1)%N]); // 放下右边的叉子
    }
}
```

4. 为什么要用两个信号量呢？emptyBuffers 和 fullBuffer 分别有什么直观含义？

```
class BoundedBuffer { // buffer
    mutex = new Semaphore(1);
    fullBuffers = new Semaphore(0);
    emptyBuffers = new Semaphore(n);
}

BoundedBuffer::Deposit(c) {
    emptyBuffers->P();
    mutex->P();
    Add c to the buffer;
    mutex->V();
    fullBuffers->V();
}

BoundedBuffer::Remove(c) {
    fullBuffers->P();
    mutex->P();
    Remove c from buffer;
    mutex->V();
    emptyBuffers->V();
}
```

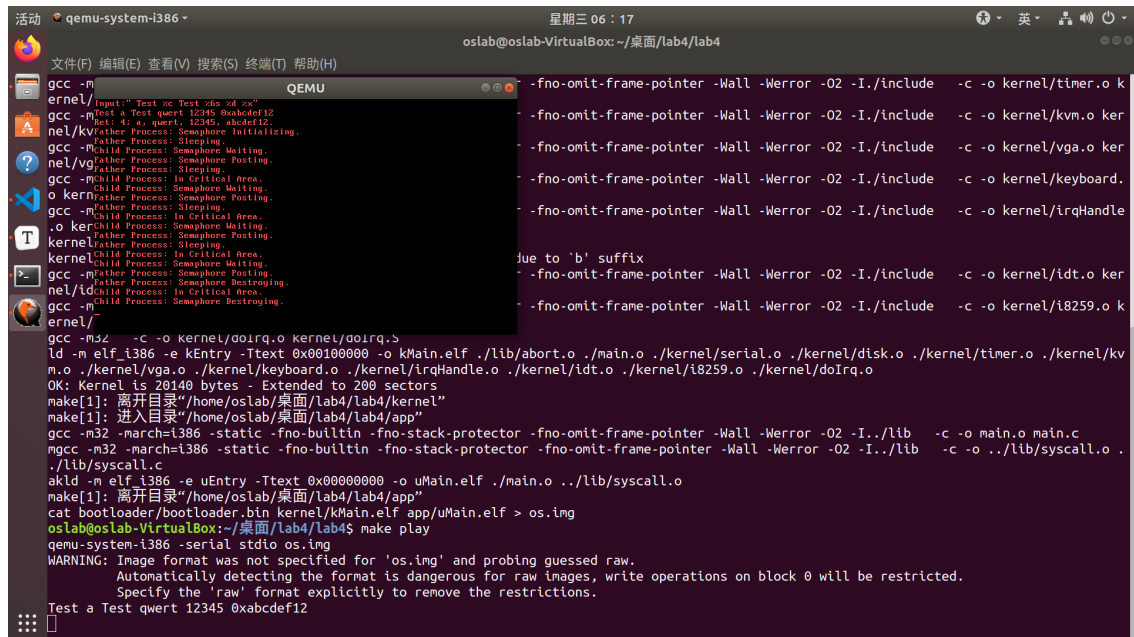
答：因为只有一个信号量时，将无法同时判断缓冲区「是否为空」和「是否为满」。

emptyBuffers 在缓冲区为空时将处于等待状态，同理 fullBuffers 在缓冲区已满时将处于等待状态。

2. Tasks

1. 完成格式化输入。

如下图：



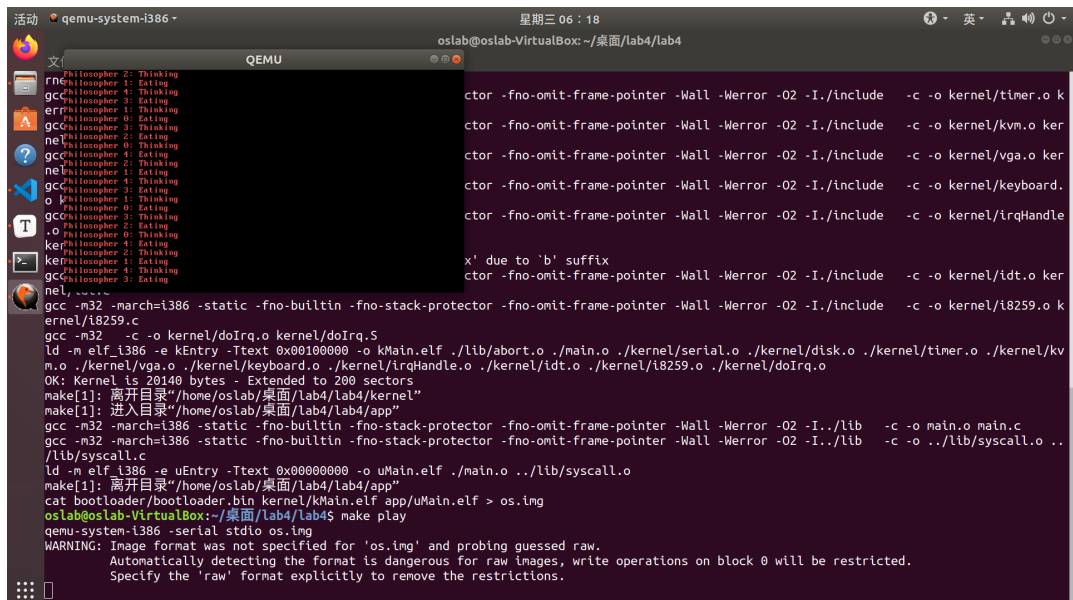
```
oslab@oslab-VirtualBox: ~/桌面/lab4/lab4
gcc -melf_i386 -fno-omit-frame-pointer -Wall -Werror -O2 -I./include -c -o kernel/timer.o k
gcc -melf_i386 -fno-omit-frame-pointer -Wall -Werror -O2 -I./include -c -o kernel/kvm.o ker
gcc -melf_i386 -fno-omit-frame-pointer -Wall -Werror -O2 -I./include -c -o kernel/vga.o ker
gcc -melf_i386 -fno-omit-frame-pointer -Wall -Werror -O2 -I./include -c -o kernel/keyboard.
gcc -melf_i386 -fno-omit-frame-pointer -Wall -Werror -O2 -I./include -c -o kernel/irqHandle
gcc -melf_i386 -fno-omit-frame-pointer -Wall -Werror -O2 -I./include -c -o kernel/idt.o ker
gcc -melf_i386 -fno-omit-frame-pointer -Wall -Werror -O2 -I./include -c -o kernel/i8259.o k
ld -m elf_i386 -e kEntry -Ttext 0x00100000 -o uMain.elf ./main.o ./kernel/serial.o ./kernel/disk.o ./kernel/timer.o ./kernel/kv
m.o ./kernel/vga.o ./kernel/keyboard.o ./kernel/irqHandle.o ./kernel/idt.o ./kernel/i8259.o ./kernel/doIrq.o
OK: Kernel is 20140 bytes - Extended to 200 sectors
make[1]: 离开目录"/home/oslab/桌面/lab4/lab4/kernel"
make[1]: 进入目录"/home/oslab/桌面/lab4/lab4/app"
gcc -m32 -march=i386 -static -fno-builtin -fno-stack-protector -fno-omit-frame-pointer -Wall -Werror -O2 -I../lib -c -o main.o main.c
gcc -m32 -march=i386 -static -fno-builtin -fno-stack-protector -fno-omit-frame-pointer -Wall -Werror -O2 -I../lib -c -o ../lib/syscall.o
ld -m elf_i386 -e uEntry -Ttext 0x00000000 -o uMain.elf ./main.o ../lib/syscall.o
cat bootloader/bootloader.bin kernel/kMain.elf app/uMain.elf > os.img
oslab@oslab-VirtualBox:~/桌面/lab4/lab4$ make play
qemu-system-i386 -serial stdio os.img
WARNING: Image format was not specified for 'os.img' and probing guessed raw.
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
Specify the 'raw' format explicitly to remove the restrictions.
Test a Test qwert 12345 0xabcd12
```

2. 实现下面四个函数。

- o `syscallSemInit()`
- o `syscallSemWait()`
- o `syscallSemPost()`
- o `syscallSemDestroy()`

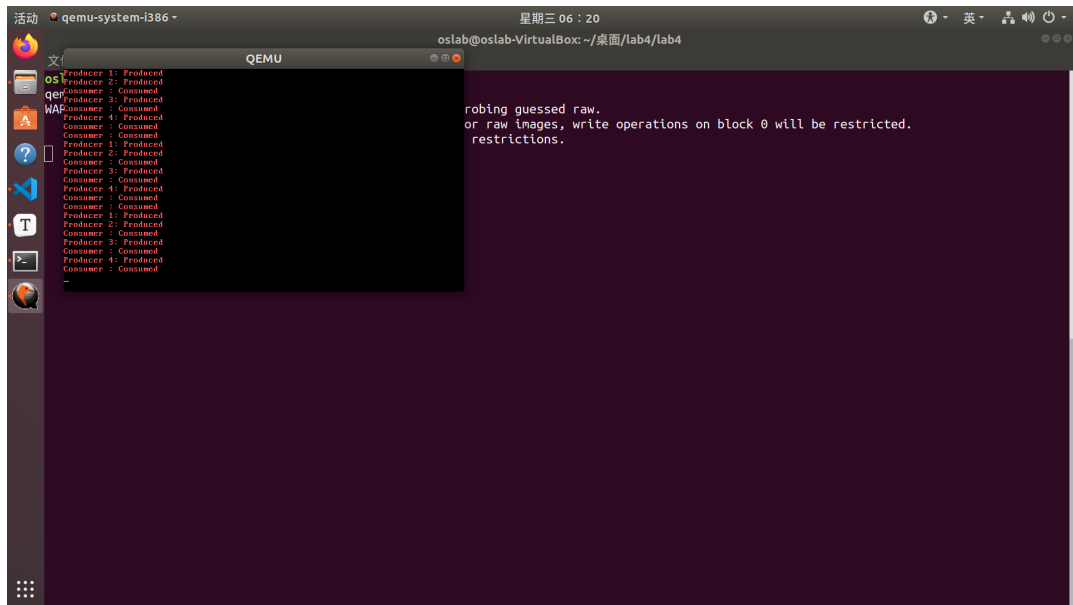
3. 完成 app 里面的下列问题，在报告里放上运行截图。

- o 哲学家就餐问题



```
oslab@oslab-VirtualBox: ~/桌面/lab4/lab4
gcc -melf_i386 -fno-omit-frame-pointer -Wall -Werror -O2 -I./include -c -o kernel/timer.o k
gcc -melf_i386 -fno-omit-frame-pointer -Wall -Werror -O2 -I./include -c -o kernel/kvm.o ker
gcc -melf_i386 -fno-omit-frame-pointer -Wall -Werror -O2 -I./include -c -o kernel/vga.o ker
gcc -melf_i386 -fno-omit-frame-pointer -Wall -Werror -O2 -I./include -c -o kernel/keyboard.
gcc -melf_i386 -fno-omit-frame-pointer -Wall -Werror -O2 -I./include -c -o kernel/irqHandle
gcc -melf_i386 -fno-omit-frame-pointer -Wall -Werror -O2 -I./include -c -o kernel/idt.o ker
gcc -melf_i386 -fno-omit-frame-pointer -Wall -Werror -O2 -I./include -c -o kernel/i8259.o k
ld -m elf_i386 -e kEntry -Ttext 0x00100000 -o uMain.elf ./main.o ./kernel/serial.o ./kernel/disk.o ./kernel/timer.o ./kernel/kv
m.o ./kernel/vga.o ./kernel/keyboard.o ./kernel/irqHandle.o ./kernel/idt.o ./kernel/i8259.o ./kernel/doIrq.o
OK: Kernel is 20140 bytes - Extended to 200 sectors
make[1]: 离开目录"/home/oslab/桌面/lab4/lab4/kernel"
make[1]: 进入目录"/home/oslab/桌面/lab4/lab4/app"
gcc -m32 -march=i386 -static -fno-builtin -fno-stack-protector -fno-omit-frame-pointer -Wall -Werror -O2 -I../lib -c -o main.o main.c
gcc -m32 -march=i386 -static -fno-builtin -fno-stack-protector -fno-omit-frame-pointer -Wall -Werror -O2 -I../lib -c -o ../lib/syscall.o
ld -m elf_i386 -e uEntry -Ttext 0x00000000 -o uMain.elf ./main.o ../lib/syscall.o
cat bootloader/bootloader.bin kernel/kMain.elf app/uMain.elf > os.img
oslab@oslab-VirtualBox:~/桌面/lab4/lab4$ make play
qemu-system-i386 -serial stdio os.img
WARNING: Image format was not specified for 'os.img' and probing guessed raw.
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
Specify the 'raw' format explicitly to remove the restrictions.
Test a Test qwert 12345 0xabcd12
```

- o 生产者-消费者问题（选做）



- 读者-写者问题 (选做)