



5. 为了使程序在多道程序环境下能并发执行,并能对并发执行的程序加以控制和描述。影响:使程序的并发执行得以实行。

8. 进程管理: 通用寄存器、指令计数器、程序状态字PSW、用户栈指针
 进程调度: 进程状态、进程优先级、所需其他信息、事件、阻塞原因

9. 线性方式、链接方式、索引方式

11. 就绪 → 执行: 进程调度
 执行 → 就绪: 时间片完
 执行 → 阻塞: I/O请求
 阻塞 → 就绪: I/O完成

13. 通用寄存器、指令计数器、程序状态字PSW、用户栈指针

二、

如果缺少了 $\text{signal}(\text{full})$, 那表明从第一个生产者进程开始就没有改变信号量 full 值, 导致即使缓冲池已满, full 值还是0, 导致消费者 $\text{wait}(\text{full})$ 时认为缓冲池为空, 一直等待状态。

如果缺少 $\text{signal}(\text{empty})$, 表明生产者向缓冲区投满产品后消费者才开始取, 因为 $\text{empty}=0$, $\text{full}=n$, 消费者进程取走产品后 empty 值也不变, 导致缓冲区为空时 empty 还是0, 生产者进程又会因为申请不到空缓冲区而被阻塞。



```
// 第三题
// 至多只允许有四位哲学家同时进餐
semaphore chopstick[5] = {1, 1, 1, 1, 1};
semaphore room = 4;
void philosopher(int i)
{
    while (true)
    {
        think();
        wait(room);
        wait(chopstick[i]);
        wait(chopstick[(i + 1) % 5]);
        eat();
        signal(chopstick[(i + 1) % 5]);
        signal(chopstick[i]);
        signal(room);
    }
}
```