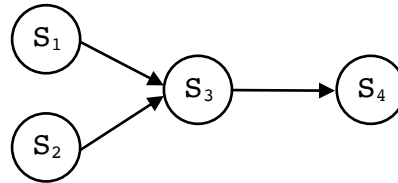


第二章作业

姓名：袁昊男 学号：2018091618008

1. 试画出下面四条语句的前驱图：

S1: $a:=x+y$; S2: $b:=z+1$; S3: $c:=a-b$; S4: $w:=c+1$



2. 说明进程在基本状态之间转换的典型原因。

- ①进程接收到了来自调度程序分配的处理机。可以使就绪态转变为执行态。
- ②分配给进程的时间片已完而被剥夺处理机，可以使执行态转变为就绪态。
- ③发生某事件导致当前进程受阻（如访问临界资源失败），可以使执行态转变为阻塞态。

3. 在进行进程切换时，所要保存的处理机状态信息有哪些？

- ①通用寄存器；②指令计数器；③程序状态字PSW；④用户栈指针。

4. 在创建一个进程时所要完成的主要工作是什么？

- ①申请空白PCB，为新进程申请获得唯一的数字标识符，并从PCB集合中索取一个空白PCB；
- ②为新进程分配其运行所需的资源，包括各种物理和逻辑资源，如内存、文件、I/O设备和CPU时间等；
- ③初始化进程控制块PCB（包括标识信息、处理机状态信息、处理机控制信息）；
- ④如果进程就绪队列能够接纳新进程，便将新进程插入就绪队列。

5. 同步机制应遵循哪些基本准则，为什么？

- ①空闲让进。如果临界区空闲，则只要有进程申请就立即让其进入，以有效利用资源。
- ②忙则等待。每次仅允许一个进程处于临界区，保证对临界资源的“互斥”访问。
- ③有限等待。进程只能在临界区内逗留有限时间，不得使其他进程在临界区外陷入“死等”。
- ④让权等待。进程不能进入临界区时，应立即释放处理机，以免陷入“忙等”状态。

6. 在生产者-消费者问题中，如果将两个wait操作互换，或者将两个signal互换，结果会如何？

①wait操作互换：将wait(full)和wait(mutex)互换位置后，可能引起死锁。如缓冲区全满时，某个生产者进程首先执行wait(mutex)操作并成功，再执行wait(empty)操作时，由于缓冲区已满，操作失败进入阻塞状态，等待某个消费者进程执行signal(empty)来唤醒。从而后续执行wait(mutex)操作的其他生产者和消费者进程全部因为mutex被占用而进入阻塞状态，引起系统死锁。

②signal操作互换：signal(mutex)和signal(full)互换位置后只影响进程对临界资源的释放次序，不会引起系统死锁，因此可以互换位置。

7. 尝试解释引入线程的意义。

①把线程作为调度和分派的基本单位。在线程切换时，仅需保存和设置少量寄存器内容，切换代价远低于进程。

②统一进程或不同进程中的线程都能并发执行，提高了OS的并发性。

③多处理机系统中将一个进程中的多个线程分配到处理机上并发执行，加速进程的完成。

④基本思想：不把调度和分配的基本单位也同时作为拥有资源的单位，“轻装上阵”；对于拥有资源的基本单位，又不对之施以频繁的切换。