

第五章作业

复习题

5.8

1. 初始化操作，其初始化值一般为正整型数字。
2. wait 原语操作，使得信号量减 1，如果这时其值为负数，则进程进入阻塞状态。
3. signal 原语操作，使得信号量加 1，如果这时其值为非正数，则进程解除阻塞状态。

5.11

管程是由一个或多个过程，一个初始化序列和局部数据组成的软件模块。

5.13

1. 一次只能有一个写者进程可以写入这个文件。
2. 若一个写者进程正在写入文件，则禁止任何读者进程读取这个文件。
3. 任意多的读者进程可以同时读取同一文件。

习题

5.3

a.

进程	执行的语句	x
P2	x=x-1	9
P1	x=x-1	8
P2	x=x+1	9
P2	if (x!=10)	9
P1	x=x+1	10
P2	printf(“x is %d”, x)	10

b.

进程	执行的语句	内存中的 x	寄存器 R0
P2	LD R0, x	10	10
P2	DECR R0	10	9

P2	STO R0, x	9	9
P1	LD R0, x	9	9
P1	DECR R0	9	8
P1	STO R0, x	8	8
P2	LD R0, x	8	8
P2	INCR R0	8	9
P1	LD R0, x	8	8
P1	INCR R0	8	9
P2	STO R0, x	9	9
P1	STO R0, x	9	9
P1	if(x!=10)	9	9
P1	printf(“x is %d”, x)	9	9
P1	LD R0, x	9	9
P1	DECR R0	9	8
P1	STO R0, x	8	8
P2	if(x!=10)	8	8
P2	printf(“x is %d”, x)	8	8

5. 10

a.

可以，该处语句能执行到的条件为 `keyi==0`，此时执行 `exchange` 语句，可将 `bolt` 值置为 0，同时 `keyi==1` 不会影响其余语句的执行。

b.

直接 `bolt` 值置为 0 更好，这样做不仅可以使得代码一目了然，而且 `exchange` 交换相对于直接赋值而言需要的资源更多一些。

5. 21

a, c 会发生严重的甚至致命的错误，因为这时如果当缓冲区为空 (`n.count==0`) 时消费者曾经进入过临界区，那么任何一个生产者都不能继续往缓冲区添加数据项，系统发生死锁。

b, d 尽管不会产生严重的问题，但是会使得无阻塞进入临界区等待的时间有所延迟，从而造成一定的资源浪费。