Haegang Yang

hzy0052@auburn.edu

1.

1-1.

P1 : x = x – 1; 9

P1 : x = x + 1; 10

P2 : x = x – 1; 9

P1 : if (x != 10); 9

P2 : x = x + 1; 10

P1 : printf(“x is %d”, x); 10

“x is 10”

1-2.

P1 : LD R0, x

P1 : DECR R0

P1 : STO R0, x

P2 : LD R0, x

P2 : DECR R0

P2 : STO R0, x

P1 : LD R0, x

P1 : INCR R0

P2 : LD R0, x

P2 : INCR R0

P2 : STO R0, x

P2 : if(x != 10) printf(“x is %d”, x)

P2 : “x is 9”

P1 : STO R0, x

P1 : if(x != 10) printf(“x is %d”, x)

P1 : “x is 9”

P1 : LD R0, x

P1 : DECR R0

P1 : STO R0, x

P2 : LD R0, x

P2 : DECR R0

P2 : STO R0, x

P1 : LD R0, x

P1 : INCR R0

P1 : STO R0, x

P1 : if(x != 10) printf(“x is %d”, x)

P1 : “x is 8”

2.

Binary semaphore is literally a binary. They can have only 1 and 0 as the value of semaphore.

On the other hand, obviously, general semaphore can take more than two values. It can be any numbers.

3.

Monitor is a high level abstraction for process synchronization.

4.

Basically there are two operations on semaphores, which are Wait() and Signal(). Wait() makes the value of the semaphore decrease. When it becomes negative, the process that Wait() calls would be a block, while the process can carry on if it is not negative. In case of Signal(), it increases the value of the semaphore. If it is not positive, it wakes the blocked process.