Jeremiah Hanson

Problems

3.2

```
var = 0;

sign = true;


process xx {

        Dec(entry, enter);

        If (sign) {

                While (sign): skip;

        }

        Critical section

        Inc(var, sign);

}
```

This will work for two workers since it will let one in then sign will be negative and it will have the other process waiting in a loop till the first process leaves. Since Dec is called first only one can be inside at a time.

4.13

This does await B since the while loop will continue till B becomes true. It will spin through the loop incrementing nd till be becomes true. It does not avoid deadlock if B is never set to true. Since B is changed outside of this code it could deadlock. B does have to be true before anything can get through the while loop.

4.21

a) This solution works because it keeps a semaphore for the critical section and one for writers and readers. Only one writer can ever have write semaphore but all readers can have reader semaphore. If there is a delayed writer then that takes priority because both the writer and the reader code first checks for a delayed writer and wakes it. Also, only the writer wakes readers since they can only be delayed if there is a writer currently.

b) This favors the writer and so I would say it has writers preference.

c) This solution has at least 2 p and v calls and at most 3 p's and 3 v's while the other solution can continue to make p and v calls forever until B becomes true. I think that This solution is easier to understand because there are more variables that all have special meaning and there is no loop that can add complication. Reading this code is easier since it flows well.