CMU 15-671 Models of Software Systems Fall 1995

Shared Resources

Garlan & Wing Homework 9 Due: 1 November 1995

The first problem is very easy. The second problem tests whether you've been keeping up with the reading. Read it now and start thinking about it early!

Problem 1.

Suppose P and Q are sequential processes that share the integer variable x, initialized to 0. P's code looks like:

$x := 1;$
$x := x + 5$

and Q's code looks like:

$x := 2;$
$x := 2x$

Suppose P and Q are run in parallel. Consider each assignment statement to be atomic.

a. Give all the possible interleavings of P's and Q's statements. What are all the possible final values of x?

b. Suppose each of P's and Q's code is atomic:

P's code:

$<< x := 1;$
$x := x + 5 >>$

and Q's code:

$<< x := 2;$
$x := 2x >>$

Now what are the possible final values of x?

Problem 2.

A local bar has has only one bathroom and patrons are asked to obey the following rules of etiquette for using it:

- A card that has the face of the Queen of Hearts on one side and that of the King of Hearts on the other is in a box to the right of the bathroom door. If the bathroom is empty and a lady (gentleman) goes in, then she (he) hangs the card on the nail on the bathroom door with the Queen (King) face visible.
- More than one person may be in the bathroom at once, but they must be of the same sex.
- The last person out of the bathroom places the card back in the box.

Besides being a full-time student, you moonlight as the barkeeper. You just learned about mutexes and condition variables in 15-671 and decide to design a Unisex Bathroom Protocol that satisfies the above rules. In addition, you attempt to ensure that the following safety and liveness properties hold:

- Mutual exclusion is enforced. Ladies and gentlemen are never in the bathroom at the same time.
- No indefinite postponement of one sex over another can occur.
- Fairness: Ladies (gentlemen) do not have priority over gentlemen (ladies).
- No starvation or deadlock can occur. A person is never prevented from entering the bathroom.

a. Write out pseudo-code (don't feel obligated to write Modula-3 code!) that implements your Unisex Bathroom Protocol. You may introduce any kind and as many auxiliary data structures (possibly protected by mutexes) as you want, e.g., queues of waiting ladies (gentlemen) (or a single queue of both sexes) and/or queues of exiting ladies (gentlemen). Stick to just mutexes and condition variables with Modula-3 semantics as your sychronization primitives. You may use the LOCK statement of Modula-3 if you find it convenient. Assume a strong fairness scheduler. For example, your solution may (but need not) have a general structure that looks like:

```
m: mutex
gentsOut, ladiesOut: condition
ladiesWaiting, gentsWaiting: int

procedure LadyEnter();
    begin
        lock m DO
            fill in this bit
        end
    end

procedure LadyExit();
    begin
        lock m DO
            fill in this bit
        end
    end

(et cetera)
```

Hint: This problem is similar to the readers-writers problem except more than one "writer" may access the resource at the same time.

b. Argue informally that your solution satisfies the rules of etiquette.

c. Argue informally that the additional safety and liveness properties hold.

d. Can deadlock occur in your solution? Why or why not?

e. Does your solution allow maximal concurrency (e.g., if ladies (gentlemen) are allowed in then no ladies (gentlemen) are waiting outside)?