Daniel Segal
dls148
EECS 338

Assignment 4

H_2O Production Problem with Monitors

<Monitor>

```
type h2o-bonding = monitor
        var count: array[0..1] of int;
        var ready: array[0..1] of int;

        procedure entry testAvailable (int i)
        begin
                if count[0] >= 2 and counts[1] >= 1
                        then begin
                                ready[0].signal;
                                temp := (i+1)%2
                                ready[temp].signal;
                                counts[0] = counts[0] – 2;
                                counts[1] = counts[1] – 1;
                        end;
                        else
                                ready[i].wait;
                        end;
                end;
```

<Hydrogen>

MONITOR USE:

```
var B: h2o-bonding
B.testAvailable(0)
bond()
```

<Oxygen>

MONITOR USE:

```
var B: h2o-bonding
B.testAvailable(1)
bond(
```

In this solution, I use monitors to negotiate one oxygen atom and two hydrogen for a bond. When the needed materials are available, the appropriate ready condition is signaled, and the pair are allowed to bond, otherwise the materials are held in a waiting state.

Baboon Crossing with Conditional Crucial Regions

<Initialization>

var *counters:* **shared record**
right, left, shared: integer;
rightActive, leftActive, rightStarving, leftStarving: boolean;
//Assuming all integers set to 0, all booleans set to false

<Left Side>

region *counters*
do begin
//Prevents starvation of opposite side
if (*right > 0* **and** *leftActive == true)*
rightStarving := true*;*
end;
left = left + 1;

//Baboons crossing under ideal conditions
while (*shared < 5* **and** *rightActive == false* **and**
rightStarving == false)
leftActive := true;
end;
shared = shared + 1;

cross();

//Update current understanding of location of baboons
shared = shared - 1;
left = left – 1;

if (*shared == 0*) *leftActive* := false;
if (*left == 0*) l*eftStarving* := false;
end;

```
<Right Side>
        region counters
                do begin
                        //Prevents starvation of opposite side
                        if (left > 0 and rightActive == true)
                                leftStarving := true;
                        end;
                        left = left + 1;

                        //Baboons crossing under ideal conditions
                        while (shared < 5 and lefttActive == false and
                                leftStarving == false)
                                rightActive := true;
                        end;
                        shared = shared + 1;

                        cross();

                        //Update current understanding of location of baboons
                        shared = shared - 1;
                        right = right – 1;

                        if (shared == 0) rightActive := false;
                        if (rightt == 0) rightStarving := false;
                end;
```