Informatics and Mathematical Modelling
Technical University of Denmark
Building 321
DK–2800 Lyngby
Denmark

HHL 30–10–2008

02152 CONCURRENT SYSTEMS FALL 2008

CP Exercise Class 7

Monday November 3

Rendezvous

1. Do Andrews Ex. 8.14. Use rendezvous (**in**) to implement the operations.

2. Do Andrews Ex. 8.15 using rendezvous.

3. A monitor-implementation of a semaphore-like mechanism is given below. In the monitor, *posinteger* is the type of all positive (> 0) integers.

```
monitor Event

  var S : integer := 0;
      Q : condition;

  procedure Pass;
    if S = 0 then wait(Q)

  procedure Clear(var r : integer);
    r := S;
    S := 0

  procedure Release(v : posinteger);
    S := S + v;
    signal_all(Q)

end
```

Define a module with the same interface as *Event* and implement it using rendezvous. Make sure that you get an effect similar to *signal_all* in *Release*.

4. The *Event* monitor corresponds to the semaphore mechanism found in the (now bygone) operating system OS/2. Show how to use (an instance e of) *Event* to implement a classical semaphore s. Hint: $\mathtt{V}(s)$ can be implemented simply as $e.Release(1)$.

5. Do Andrews Ex. 8.12. Assume that guards (including ?*op*) are reevalutated whenever they may have changed.