Distributed Systems

Unassessed Coursework No. 1

Lecturer: M. Sloman,

PATIENT MONITORING SYSTEM

The intensive care ward in a hospital consists of 16 beds. Patients in each bed are continuously monitored for a number of factors, such as pulse, temperature and blood pressure by means of sensors attached to their body and connected to a bedside computer, which displays current readings. A central nurse station displays the current reading for all beds and can also be used to set the rate of monitoring for each patient.

a) Outline the structure of the patient monitoring system, indicating your breakdown into software components (processes). Indicate the function of each component, the data flows between components in terms of the data types transferred.

   A process component should be provided to handle each device (eg. sensor, display) and provide a message passing interface to the rest of the system. Do not try to minimise the number of processes; this can always be done later if necessary.

b) For each main process component, outline the code (i.e. pseudocode) that you would provide to perform its function. Assume that the following communication primitives are available:

   SEND message TO destination
   - Send a message and continue processing (asynchronous)

   RECEIVE message FROM source DELAY (n)
   - Receive *message* if available from *source*, otherwise wait for up to *n* seconds
   n= -1 implies no timeout

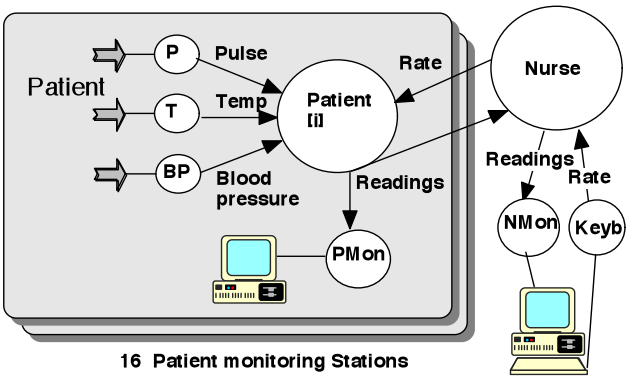
   source = RECIEVEANY message (DELAY n)
   - Receive a *message* from any *source*, otherwise wait for up to *n* seconds

Distributed Systems

Solution Unassessed Coursework No. 1

Lecturer: M. Sloman

PATIENT MONITORING SYSTEM




```
//Data
short pulse;
short blood_pressure;
float temp;
rate short;    //readings per min
struct readings {
    pulse p; blood_pressure bp; temp t; short bed}
```

P, T, BP could be polled or event driven
Polled is simpler as no timing required

Simplifed outline Processes
Assume every message includes a type field

```
Process Patient {
    short delaytime = 5

    loop {
        receive msg from nurse delay (delaytime)
        case msg type
            rate: delaytime = 60/rate;
            null:  //timeout
               send query to P;  receive pulse from P;
               send query to T;  receive temp from T;
               send query to BP;  receive press from BP;
               set up readings message
               send readings to PMon and Nurse
    }
}

Process Nurse {
    loop {
        source = receiveany msg delay (-1)
        case source
            patient[i]: send msg reading to Nmon
            Keyb:  send msg rate to patient[msg.bed]
    }
}
```

Other processes are left as an exercise for you!