US202

Two Hours

UNIVERSITY OF MANCHESTER

INSTITUTE OF SCIENCE AND TECHNOLOGY

CT202 Concurrent Systems

For candidates taking:

BSc IN COMPUTING SCIENCE

SECOND YEAR SESSIONAL

MEng, BEng IN SOFTWARE ENGINEERING

SECOND YEAR SESSIONAL

MEng, BEng IN COMPUTER SYSTEMS ENGINEERING

SECOND YEAR SESSIONAL

MEng, BEng IN ELECTRONIC AND MICROELECTRONIC SYSTEMS ENG

SECOND YEAR SESSIONAL

MEng, BEng IN COMPUTING AND COMMUNICATION SYSTEMS ENG

SECOND YEAR SESSIONAL

Monday 15 May 2000 9.30-11.30

Answer **three** questions

The use of electronic calculators is NOT permitted

Note: Do not answer more than the required number of questions. Clearly cross out anything you do not wish to be marked.

PTO

1) Answer all parts

(a) Discuss how monitors provide support for communication and synchronisation between concurrent processes.

(8 marks)

(b) This part of the question involves developing pseudo-code for a simple concurrent program that simulates the activity in a barber's shop. The shop consists of n chairs to seat waiting customers and a single barber's chair where a customer receives a haircut. If there are no customers available, then the barber goes to sleep. If a customer arrives whilst the barber is sleeping, then the barber is awakened and performs the haircut. If a number of customers arrive whilst the barber is cutting hair, they take one of the available seats. If all the seats are occupied, then the customer leaves the shop.

In the simulation, the barber and the customers are implemented as concurrent processes. Write a monitor to provide the necessary synchronisation. The interface of the monitor is :

```
monitor barberShop is
    procedure awaitCustomer; -- called by the barber process
```

```
        procedure getHaircut; -- called by the customer processes
end barberShop
```

(8 marks)

(c) Consider a system with 4 processes (P_1, P_2, P_3, and P_4) and 2 monitors (M_1 and M_2). One of M_1's functions, fm1, calls one of M_2's functions, fm2. During the execution of the system, P_1 calls fm1, which then calls fm2. However at this point, fm2 is already being executed by P_2. Explain which, if any, processes hold mutually exclusive access to the monitors, and briefly consider what happens if P_3 and P_4 now attempt to access M_1.

(4 marks)

2) Answer all parts

(a) In their book on concurrent programming, Burns and Davies make the following observation : *'The operations of a concurrent program are partially ordered.'* Explain what this means. Briefly discuss (with the aid of an example) how this partial ordering can lead to non-deterministic results i.e. different results being generated by different executions of the same concurrent program using the same input data.

(4 marks)

(b) With reference to part (a), explain what is meant by the term *atomic action*, and explain the significance of atomic actions in concurrent programming. Why is the following C program statement not atomic :

```
/* x, y, and z are integers */
x = y + z;
```

How would you construct an atomic action using semaphores?

(6 marks)

QUESTION 2) CONTINUED ON NEXT PAGE

PTO

QUESTION 2) CONTINUED

(c) The Petri Net below models the readers-writers problem, which concerns concurrent access to a database, or some other shared resource. The resource access protocol is as follows : multiple reader processes are allowed to access the resource concurrently, but only one writer is allowed in at once. In addition, during a write, no readers are permitted access.

The transitions are labelled with the events that they represent. The k tokens in place p1 represent the k processes in the system performing activities unconnected with access to the shared resource. When a process wishes to access the resource, it makes a request (represented by transition t1), and then declares whether the access is to be for reading or writing (t2 and t3).

(i) Explain how the above Petri Net models the above readers-writers access protocol.

(ii) The above model is prone to a problem known as *writer starvation*. Explain what do you think this means and indicate, with reference to the above Petri Net, and your answer to part (a), how it arises.

(iii) Modify the above Petri Net to eliminate the problem of writer starvation. Do this in two stages.

First do not allow a new reader to start if a writer is waiting. What problems does this introduce?

Second, develop a model where a new reader is not allowed to start if a writer is waiting AND also give readers preference over waiting writers at the end of a write operation.

(*Hints for the second stage :* (1) you will need to modify the net in the vicinity p5 (2) do not put the token back into p5 immediately after a write access ends)

(10 marks)

3) Answer all parts

(a) Explain the difference between synchronous and asynchronous message passing communication.

(2 marks)

(b) Compare and contrast message passing communication in Ada with message passing along CSP/occam channels.

(8 marks)

(c) Write the body of an Ada task type that implements a semaphore by making use of rendezvous-based communication. The specification (interface) of the task is as follows :

```
task type SEMAPHORE is
  entry WAIT;
  entry SIGNAL;
end SEMAPHORE;
```

In a blocking implementation of semaphores, processes that are suspended on the semaphore are placed on a queue associated with the semaphore. With reference to the operation of the Ada rendezvous, explain how is this queuing is accomplished in your implementation.

(10 marks)

PTO

4) Answer all parts

(a) Briefly explain the operation of the following scheduling algorithms :

Shortest job first (SJF)
Time-slice or round robin scheduling (RR)
Priority-based pre-emptive scheduling

(6 marks)

(b) Define the terms turnaround time and waiting time.

(2 marks)

(c) Consider the following set of processes, with the length of the CPU burst time given in milliseconds :

Process	Burst time	Priority
P1	5	3
P2	1	1
P3	2	3
P4	1	4
P5	3	2

The processes are assumed to have arrived in order P1, P2, P3, P4, P5 at time 0.

(i) Draw 2 Gantt charts showing the execution of these processes using SJF and RR (with a time-slice=1) scheduling.

(ii) What is the turnaround time of each process using each of the scheduling algorithms in (i).

(iii) What is the waiting time of each process using each of the scheduling algorithms in (i).

(iv) Which of the schedules in part (i) results in the minimum average turnaround time, over all processes, and which results in the minimum average waiting time, over all processes.

(10 marks)

(d) Explain the meaning of the term *busy-waiting*. Consider a single processor system in which the kernel provides a semaphore facility. The pause associated with executing wait on a zero-valued semaphore is implemented by busy-waiting. What are the implications of this decision for the scheduling mechanism?

(2 marks)

5) Answer all parts

(a) Explain what is meant by *relocation* in the context of memory management, and why it is an important issue in memory management. Discuss how relocation can be accomplished using base and limit registers, and in an operating system that that uses paging in memory management.

(7 marks)

(b) Explain the difference between blocking, non-blocking and asynchronous input/output (I/O). Why is blocking I/O favoured in many situations? Which form of I/O would you use for :

(i) Access to a hard disk?
(ii) Mouse input to an interactive program with a Graphical User Interface.

Briefly explain your answers to (i) and (ii).

(8 marks)

(c) Discuss how non-blocking I/O may be achieved in a system that only supports blocking I/O, but which does support threads (or light-weight processes). To what extent does the success of your scheme depend on the way in which threads are managed in the system?

(5 marks)

END OF PAPER