The University of Western Australia
Department of Electrical and Electronic Engineering

FIRST SEMESTER EXAMINATIONS **JUNE 2000**

REAL-TIME DISTRIBUTED SYSTEMS 408 **623.408**

ONLY PAPER

Time Allowed: Three Hours

This papers contains: 12 questions; 7 pages.

Instructions to Candidates:
You are required to answer 10 questions. If you do more than 10 questions the highest marked 10 questions are used in the final mark. Each question carries equal marks and the mark associated with each part of a question is as shown.

QUESTION 1

a) What are the three simple performance metrics for commercial RTOSs that are commonly used? Just considering one of those metrics, also distinguish between the performance of currently available RTOS kernels and conventional OS kernels, and identify the main reasons for this variation. [5]

b) Identify the two classes of Real-Time Operating System (RTOS) that provide the greatest support for meeting explicit real-time constraints and describe some of their distinguishing features? [5]

QUESTION 2

a) Under what circumstances do exceptions propagate and how do they propagate when performing multitasking in the Ada programming language? [4]

b) In the Java programming language what is the scheduling approach used in multi-threading and what control over thread priority is there? [3]

c) Discuss what you understand by the object *serialisation* and *marshalling* function in the context of RMI, DCOM and CORBA implementations. [3]

QUESTION 3

a) In the LynxOS Real-time Operating System, describe the purpose of *kernel threads* and what benefit they bring to system responsiveness. In particular, mention the mechanism of *priority tracking*. [5]

b) The approach used in the *Hartstone* benchmark suite is significantly different to that used in other RTOS benchmarks. What is the basic philosophy used in this benchmark, and what scheduling discipline is used and why? [5]

QUESTION 4

a) At a brief summary level only, identify the three main approaches to distributed processing supported by non-concurrent and concurrent programming languages (giving an example of each) [3]

b) In the SMB (Server Message Block) protocol, briefly identify what *oplocks* are and how they differ from the basic protocol? [2]

c) In the NFS (Network File System) protocol, describe with the aid of a simple diagram the I/O completion process for a write client-server operation. Identify the purpose of the *daemons* on the both client and server. [5]

<u>QUESTION 5</u>

a) Consider a distributed system operating a master-slave clock synchronisation algorithm. Suppose the slave clock returns the time $C_j(T)$ and the master clock returns the time $C_i(T)$. The interprocessor communication delays are u_i^j (master to slave) and u_j^i (slave to master). What is the clock skew (assuming the slave clock error can be modelled as a zero-mean random noise process) that can be used to correct the slave clock $C_j(T)$? Briefly mention what can be done to improve the clock skew estimation accuracy for the scheme. [4]

b) As an example, suppose a master clock had a time of 00:00:00.010000. After a master to slave communication time of 1 msec, the slave clock reads 00:00:00.012000. At a slave time of 00:00:00.080000, a communication delay of 4 msec is incurred, after which the master clock reads 00:00:00.076000. What clock update is applied to the slave clock after <u>one</u> cycle of the synchronisation algorithm? [3]

c) Consider the situation where the above clock synchronisation algorithm is operating with multiple slave processors. Suppose that all clock drift rates could be bounded at 10^{-3} sec/sec from specifications, and the clock update rate was at least once every 1 second. In addition, the skew error derived for the data of part b) above can be assumed to bound all slave clock update errors. Under these conditions, what is the bound on the maximal clock difference between any slave clock in the distributed system? [3]

QUESTION 6

a) In a real-time distributed system, where time constraints imposed at one node must propagate to another, suppose we have the time constraints TC_1 and TC_2:

TC_1 (*begin*): $1.4 \rightarrow 2.2$ TC_1 (*end*): $1.8 \rightarrow 2.6$ C_{id1}: $0.2 \rightarrow 0.6$
TC_2 (*begin*): $0.2 \rightarrow 3.0$ TC_2 (*end*): $1.2 \rightarrow 4.0$ C_{id2}: $1.0 \rightarrow 2.0$

Draw a diagram that shows both time constraint laxity windows, show TC_1 propagating onto TC_2, and then identify the regions where $TC_1 \sqcap TC_2 \neq \varnothing$ and $TC_1 \prec TC_2$. [6]

b) Suppose we have an incoming time constraint imposed by a remote object of TC_{in} and the constraint imposed on the remote object is TC_{out}, and we have a specification using temporal object relations of the form:

$$TC_{in} \ \varnothing \vee \Uparrow \ TC_R \ \succ \vee \Downarrow \vee \varnothing^u \vee \uparrow^u \vee \gg \vee \downarrow^u \ TC_{out}$$

Show that the service to be invoked on the remote object will succeed TC_{in} by less than or equal to γ time units, where $\| \ TC_R \ \| = \gamma$ by drawing appropriate inferences from the above relations. [4]

QUESTION 7

a) In the CODARTS approach to the design of Real-Time Systems, briefly describe the purpose of the cohesion criteria, identify a few of the important issues in the temporal cohesion criterion, and give a simple example of the application of this criterion (with appropriate data flow and task architecture diagram segments). [5]

b) Using a simple elevator controller example with two passengers on different floors, show how a UML sequence diagram can be used to define at least two typical relationships between temporal events involving the passengers and the elevator controller system. [5]

QUESTION 8

Suppose we have an information system module that processes *two* channels of data concurrently. Each channel takes $T_2 = 4$ and $T_4 = 3$ time units respectively to input the data, and then takes $T_3 = 2$ and $T_5 = 3$ time units respectively to convert the data. After processing, the two channels are combined to produce a composite stream (taking $T_6 = 3$) which is passed through a moving average filter (taking $T_7 = 2$). The filter averages over several data points and eliminates older data points from the average (taking $T_8 = 1$). The input data on each channel is processed quite differently in other processes (taking $T_9 = 3$ and $T_{10} = 4$), and combined with the averaged data before recording. The recording process takes $T_{11} = 6$ time units. Input data is acquired synchronously by the system every $T_1 = 7$ time units.

For the system as described, produce a time-augmented Petri net graph to model the processes, and derive the constraints imposed on the process times using the notion of *safeness in the presence of time*. Can the module achieve the specified time constraints?

[10]

QUESTION 9

You are given that the distributed control software for a Flexible Manufacturing System (FMS) consists of three assembly control processes and a shared common resource process. A Stochastic Petri net (SPN) model for this system has been devised with the following transition rates:

$\lambda_1 = 6$ (after process 1 waiting)	$\lambda_4 = 2$ (after process 1 active)
$\lambda_2 = 2$ (after process 2 waiting)	$\lambda_5 = 4$ (after process 2 active)
$\lambda_3 = 5$ (after process 3 waiting)	$\lambda_6 = 7$ (after process 3 active)

Draw the SPN model, evaluate all marking probabilities and determine the worst case FMS assembly process cycle time.

[10]

QUESTION 10

Consider the case of three periodic tasks (with context switch times included):

Task t_1: C_1 = 25 ms; T_1 = 100 ms

Task t_2: C_2 = 30 ms; T_2 = 150 ms

Task t_3: C_3 = 50 ms; T_3 = 300 ms

a) Apply the *Utilization Bound Theorem* from Real-Time Scheduling Theory to determine if these tasks are schedulable using a rate monotonic scheduling strategy. [2]

b) Suppose the computation time for task t_1 increases to 50 msec. Now determine if the tasks are schedulable, and then apply the less conservative *Completion Time Theorem* if required. [4]

c) Suppose an additional interrupt driven aperiodic task is added to the set in a) above: Task t_a: C_a = 10 ms; T_a = 200 ms
Assuming that all the other tasks are likely to share access to common data at some time, use the *Generalized Utilization Bound Theorem* to determine if the new task set is schedulable. [4]

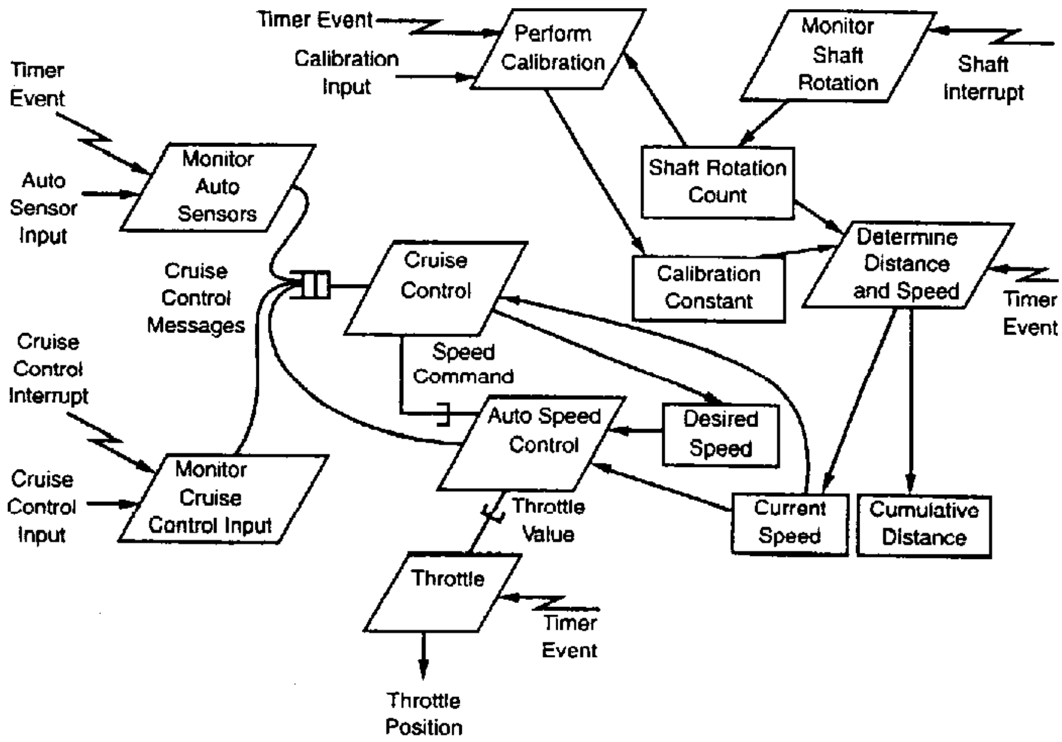
QUESTION 11

a) In a formal specification for a real-time software component in a nuclear reactor control system, suppose the initial reactor SCRAM action must be completed within 15 seconds of the operator pressing ABORT. And suppose the SCRAM process must take at least 10 seconds. In testing the system the following safety property must be met: If the reactor DUMPS within 6 seconds of completion of the SCRAM process then the emergency will be handled in 25 seconds. Show how these time constraints could be represented as RTL axioms, draw the constraint graph and reduce it to show if the safety assertion is formally consistent. [7]

QUESTION 11 continued over page

QUESTION 11 continued

b) Use any segment of the following task architecture diagram of a robot controller task architecture diagram to briefly discuss the purpose of *event sequence analysis* in real-time performance modelling:



[3]

QUESTION 12

a) In a client/server distributed architecture what are the two main approaches to design the server subsystems for low and high client demand, and identify the two main mechanisms to support data distribution. [5]

b) Discuss how Microsoft's DCOM implements interfaces and objects and in doing so make some comparisons with OMG's CORBA. [5]