THE UNIVERSITY OF WESTERN AUSTRALIA

School of Electrical, Electronic and Computer Engineering

SURNAME: ________________________________ STUDENT NO: __________

GIVEN NAMES: ________________________________ FACULTY: _________

1ST SEMESTER EXAMINATIONS 2003

REAL-TIME DISTRIBUTED COMPUTER SYSTEMS 423 (623.423)

This Paper Contains: 8 Pages (including title page)
12 Questions
Answer 10 questions only (each question is marked out of 10)

Time Allowed: 3 Hours
Reading Time: 10 Minutes

Exam papers are to be collected with the examination answer booklets.

PLEASE NOTE

Examination candidates may only bring authorised materials into the examination room. If a supervisor finds, during the examination, that you have unauthorised material, in whatever form, in the vicinity of your desk or on your person, whether in the examination room or the toilets or en route to/from the toilets, the matter will be reported to the head of school and disciplinary action will normally be taken against you. This action may result in your being deprived of any credit for this examination or even, in some cases, for the whole unit. This will apply regardless of whether the material has been used at the time it is found.

Therefore, any candidate who has brought any unauthorised material whatsoever into the examination room should declare it to the supervisor immediately. Candidates who are uncertain whether any material is authorised should ask the supervisor for clarification.

QUESTION 1

a) In the *Time-Driven Scheduling* (TDS) class of Real-Time Operating System, what is the role of *value functions* and illustrate with some typical functions for common TDS policies (make sure you also include the policy for the *general time constraint model* in your illustration). [6]

b) Testing of Real-time Distributed Systems can be particularly difficult and complex. Briefly discuss some of the key approaches and issues that arise in testing, illustrating with at least one example of the typical problems than can arise from the case studies considered. [4]

QUESTION 2

a) In the Java programming language, identify the mechanisms to support the distribution of code and data, and give a diagram to show how client/server operation is supported through distributed applet code. [5]

b) How are *threads* created in the Java language? A code segment will help to illustrate. Also identify what the scheduling approach used in multi-threading is and indicate what control over thread priority is there? [5]

QUESTION 3

a) Describe with the aid of a sketch how HTTP/CGI provides a simple 3-tier client/server model, and compare it with the emerging *Object Web* based on the Common Object Request Broker (CORBA) standard. [5]

b) Discuss two of the key features of the new trend in distributed object infrastructure that is referred to as *reflective middleware*. [5]

QUESTION 4

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). Develop an expression for 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 08:06:00.542000. After a master to slave communication time of 13 msec, the slave clock reads 08:06:00.638000. At a slave time of 08:06:00.904000, a communication delay of 5 msec is incurred, after which the master clock reads 08:06:00.986000. What clock update is applied to the slave clock after <u>one</u> cycle of the synchronisation algorithm? [2]

c) Now consider a fully *distributed clock algorithm* (without a designated master), that uses a *minimize maximum error approach.* Determine what (if any) clock update is performed at node *j* given the following conditions:

<u>At node *i*</u>: Let the reset time be 22:03:00.000000, the time node *i* is queried is 22:03:00.023200, the estimated drift rate is 0.003 sec/sec, and the estimated residual error is 20 μsec.

<u>At node *j*</u>: Let the reset time be 22:03:00.000000, the time node *j* attempts to synchronize is 22:03:00.023300, the estimated drift rate is 0.004 sec/sec, and the estimated residual error is 30 μsec. The response delay from node *i* to node *j* is 15 μsec. [4]

QUESTION 5

a) In the *Rhealstone* benchmark, describe in detail with the assistance of diagrams, the two resource based performance benchmarks of *semaphore shuffle* and *deadlock breaking*? [5]

b) A *Rate Monotonic* scheduling policy is commonly used in Real-Time Systems as a reference priority assignment strategy. Why is this? Consider a two task example with a simple diagram to clearly show the benefits of this strategy. [5]

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*): $3.0 \rightarrow 4.6$	TC_1 (*end*): $3.6 \rightarrow 6.0$	C_{id1}: $0.8 \rightarrow 2.1$
TC_2 (*begin*): $0.8 \rightarrow 6.8$	TC_2 (*end*): $4.2 \rightarrow 9.3$	C_{id2}: $2.6 \rightarrow 5.2$

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 $TC_2 \prec TC_1$. [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:

$$TC_{in} \Uparrow \vee \prec \; TC_R \Uparrow TC_{out}$$

Show that the service to be invoked must succeed TC_{in} by γ time units or more, where $\|TC_R\| = \gamma$, by drawing inferences from the above relations. [4]

QUESTION 7

a) In the Common Object Request Broker Architecture (CORBA) distributed object standard explain the purpose of *callbacks* and illustrate with a diagram. [4]

b) Describe the purpose of the Server Message Block (SMB) protocol, and illustrate one of the key features of the protocol for supporting protocol negotiation. [6]

QUESTION 8

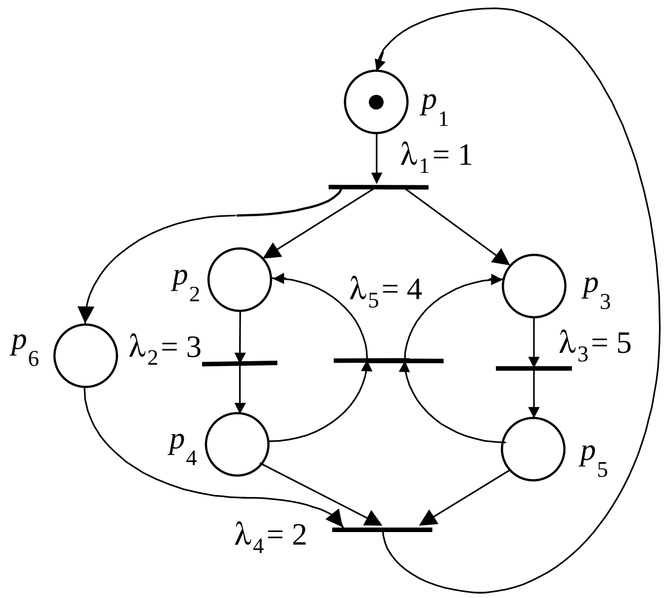
Suppose we have an information system module that processes *three* channels of data concurrently. The channels take $T_2 = 2$, $T_4 = 2$ and $T_6 = 3$ time units to input the data and then take $T_3 = 2$, $T_5 = 3$ and $T_7 = 4$ time units respectively to convert the data. After processing, the three channels are combined to produce a composite signal (taking $T_8 = 3$) which is passed through a moving average filter (taking $T_9 = 1$). The filter averages over several data points and eliminates older data points from the average (taking $T_{10} = 2$). The input data on only the first channel is processed quite seperately in another process (taking $T_{11} = 4$), and combined with the averaged data after filtering but before recording. The final recording process takes $T_{12} = 2$ time units. Input data is acquired synchronously by the system every $T_1 = 5$ 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

Given that the distributed control software for a flexible manufacturing system has the following simplified Stochastic Petri Net (SPN) model, determine the average cycle time of the system.



[10]

QUESTION 10

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

Task 1: C_1 = 20 ms; T_1 = 75 ms

Task 2: C_2 = 25 ms; T_2 = 100 ms

Task 3: C_3 = 30 ms; T_3 = 225 ms

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

b) Suppose the computation time for task 1 increases to 40 ms. Now determine if the tasks are guaranteed to be schedulable, and then apply the less conservative *Completion Time Theorem* if required. [4]

QUESTION 10 continued over page

QUESTION 10 continued

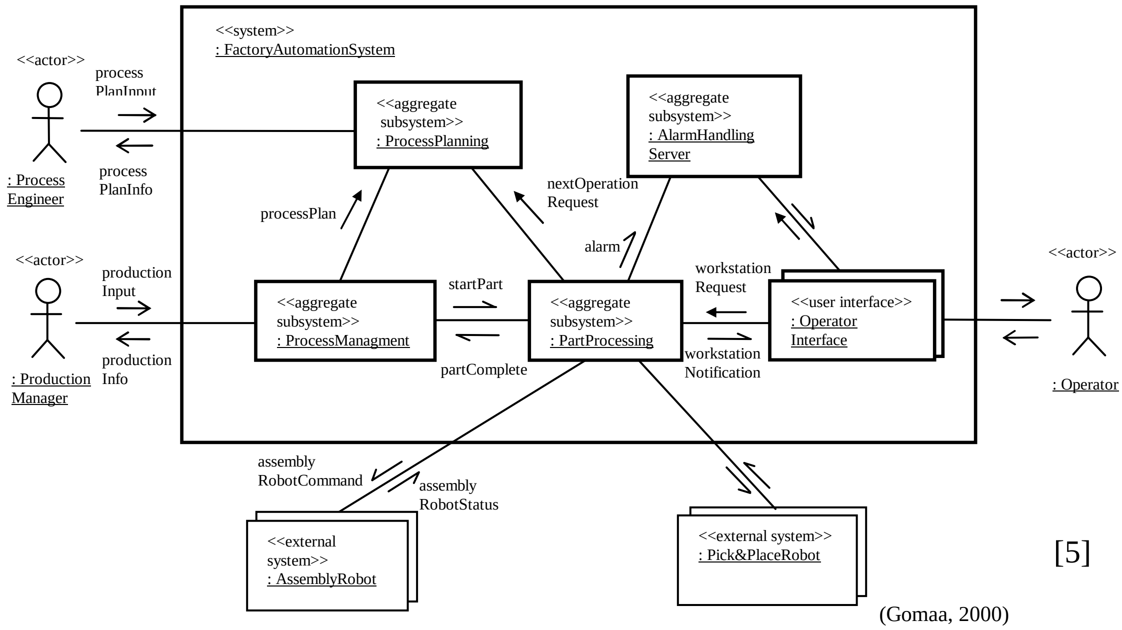
c) Suppose an additional interrupt driven aperiodic task is added to the first task set in a) above: Task A: C_A = 6 ms; T_A = 200 ms
Assuming that all the other tasks are likely to share access to common data at some time, use the *Generalized Utilization Bound Theorem* to determine if the new task set is guaranteed to be schedulable. [4]

QUESTION 11

a) In a formal specification for a real-time system software component, suppose some DISPLAY action must be complete within 36 time units of an INTERRUPT event. A HANDLER module is activated by an INTERRUPT event. The result of HANDLER (which has a computation lower bound of 28 time units) is conveyed to the DISPLAY module (which has a computation lower bound of 6 time units). Show how these time constraints could be represented as RTL expressions, produce a constraint graph and reduce it to show if the specification is satisfiable.

[5]

b) Use any segment of the following task architecture diagram for part of a factory automation system to discuss the purpose of *event sequence analysis* in real-time performance modelling:



[5]

(Gomaa, 2000)

<u>QUESTION 12</u>

a) When the software architecture design phase is considered in the development of a real-time distributed system, describe the criteria that are used to guide subsystem decomposition, indicate what form of UML diagram is used to capture subsystem software architecture, and develop a simple example to illustrate this type of diagram and what it usually shows. [5]

b) When restructuring a multi-tasking sub-system design using task inversion, detail at least two of the key guidelines that are used to reduce the number of tasks in a systematic manner, and give a simple example to illustrate application of the guidelines. [5]