THE UNIVERSITY OF BRITISH COLUMBIA

Department of Electrical & Computer Engineering

EECE 314 – System Software Engineering

Winter Session 2004

Date: **April 17th 2004**

Time: **8:30 am**

Instructions to Candidates

Answer **two** questions from **Section A** plus **two** questions from **Section B.** If the candidate answers an incorrect number of questions from each section, or in total, then there is no obligation on the part of the examiner to award marks for those questions.

All questions carry equal marks

This is a **closed book** examination. No course work material whatsoever may be used during the examination.

Time Allowed: 2½ Hours

This question paper must be returned at the end of the exam
Write your name and Student ID Below before returning it

Name:

Student ID:

SECTION A

Q1 **Part A**

(i) Explain what is meant by time-slicing in the context of a single CPU running several processes/threads at the same time and explain how the CPU can give the impression that it is running several tasks concurrently.

(3 Marks)

(ii) Explain what is meant by 'Numerically intensive' and 'interactive' processes and explain how you would optimize your time-sliced system to work with either one kind of process or the other. Show any assumptions or formula you need to support your position.

(5 Marks)

(iii) Further, from your explanation above, explain why there is a conflict when it comes to running both 'Numerically intensive' and 'interactive' processes on a time-sliced (you don't have to show how your solve it).

(2 Marks)

Part B

(i) Explain what is meant by the term 'Process Priority' and 'pre-emptive scheduling'.

(4 Marks)

(ii) Explain what is meant by priority inversion and give an example of how it could occur.

(4 Marks)

(iii) The following 3 processes A, B and C need to be scheduled to run on a system. Process A is triggered by a periodic event every 30mS, process B every 40mS and process C every 50mS. Processes A, B and C each need 10, 15 and 5 ms respectively to generate their response to these events.

Show whether or not such processes can actually be scheduled using rate monotonic scheduling techniques coupled with a pre-emptive priority based scheduler to ensure that each process can generate its response prior to being triggered again. What priorities would you allocate to these processes to achieve this?

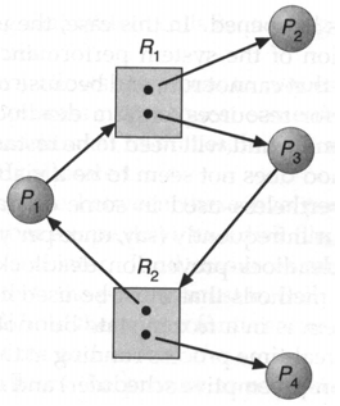
(7 Marks)

Q2

i. Explain what is meant by the term 'Deadlock'. Explain why can deadlock only occur in concurrent and not single-tasking systems? (4 Marks)

ii. Outline the 4 conditions proposed by 'Coffman' that must arise before deadlock can occur. (4 Marks)

iii. The illustration below is that of directed '*resource-allocation*' graph, showing four processes, P1 – P4, and two non pre-emptable resources R1 and R2 each having two instances.



Explain from this illustration exactly which resources have been granted and to which processes they have been given, and further, which processes are currently blocked and what resources they are waiting for.

(4 Marks)

Explain further, why the above system is NOT deadlocked and how it would be possible for all processes to eventually acquire the resources they need to complete their programming.

(3 Marks)

Outline 3 different deadlock avoidance techniques (not including the Ostrich Algorithm) that the developer could employ to reduce the likelihood of deadlock arising. Outline the drawbacks to each of these schemes

(10 Marks)

Q3 i)

a) What, in the computing sense, is meant by the term ‘synchronisation’ and give an example of day to day human synchronisation

(4 Marks)

b) Explain the functional differences between an ‘event’ a ‘condition’ and a ‘semaphore’

(8 Marks)

ii) Imagine that you are simulating a very simple 5 storey elevator controller as a thread in software. Imagine also that you have created a number of other threads to simulate passengers wishing to travel via your elevator from one floor to another. How could you employ ‘condition’ variables to synchronise the passengers to the arrival of the lift at the correct floor such that when the elevator arrived at each floor, those persons waiting to get on or off the elevator do so at the correct floor.

Outline briefly the pseudo code you would employ within

i. The passenger thread to wait for the arrival of an elevator at a particular floor and

ii. The elevator controller thread to alternately travel up and down, stopping and signaling its arrival at each floor.

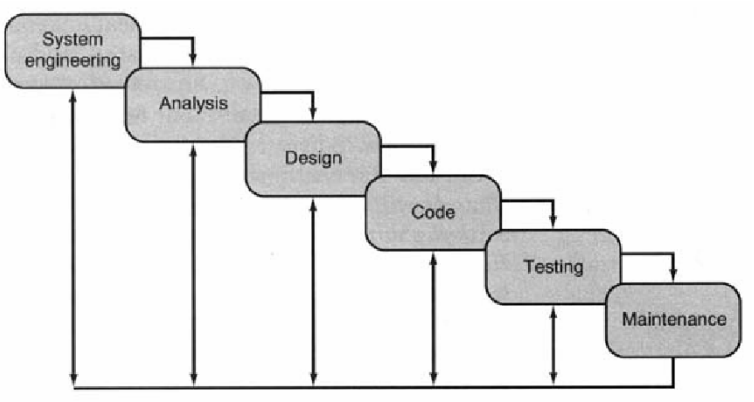
(8 Marks)

iii) Imagine that 3 passengers (i.e. friends) have all agreed to meet to go out for lunch together and have to get on the elevator. How could you arrange to synchronise these friends such neither gets on the elevator until all others have arrived?

(5 Marks)

Section B

Q4



The illustration above identifies the various stages associated with the typical Software Engineering Lifecycle or Waterfall model. For each step in the Lifecycle model, describe the sorts of issues they are trying to address, i.e. who is involved, what activities and/or development takes place. Furthermore outline some commonly employed **analysis** techniques.

(12 Marks)

It has been suggested that software products do not wear out in the conventional engineering sense, but simple degrade due to the effects of maintenance. Explain the validity of otherwise of this observation.

(4 Marks)

What is meant by **iterative** software development? Give an example of a popular iterative process model and outline a few of its key features. Explain why this approach is better for developing large scale projects than the lifecycle model, i.e. why does it lead to software that is more likely to be acceptable to the customer in the form it is delivered. Explain further why iterative development makes more sense for the developer than the big-bang approach to software development?

(9 Marks)

Q5

Describe what is meant by the following terms

1. *Software requirement*
2. *Use Case*.

(4 Marks)

What is meant by the term '*use-case scenario*' and by the terms *primary* and *secondary* scenarios?

(4 Marks)

Using a **Library** as an example, identify three common user requirements and draw a *use-case diagram* to capture them. Now document one of your use-cases in detail, i.e. describe what actions take place as a result of the user initiating that use-case and document any interesting secondary scenarios. Explain how you might capture interesting secondary scenarios on your use-case diagram?

(6 Marks)

A primary objective of Object Oriented Analysis is to identify candidate objects that could collaborate in realising the requirements laid out in our use-cases. Outline 4 popular tried and tested techniques for identifying such objects and using these techniques (where appropriate) produce a list of potential 'interesting' objects for one of your Library use-cases.

(11 Marks)

Q6

What is meant by a **sequence diagram** in UML and what does it attempt to model?
(2 marks)

The following is a use-case description for the "Withdraw cash" use-case of an ATM

- The user inserts their ID card into the system.
- The system reads the magnetic strip from the card.
- The system contacts the banks central computer to request the PIN number for the card and their account details.
- The system prompts the user for their PIN.
- The user enters their PIN.
- If PIN is authenticated, the user is prompted for the amount of the withdrawal. If not, the card is returned to the user.
- The user enters the amount of withdrawal.
- The system checks with the banks central computer to ensure that the user has sufficient funds to make the withdrawal.
- If there are sufficient funds, the cash is dispensed and the customer's account at the Bank Central Computer is debited accordingly.
- The card is returned to the user and a receipt issued.

Draw a sequence diagram showing how a user interacts with an ATM in order to obtain Cash. The diagram should include the following objects and actors

- The Customer, the ATM itself, a receipt printer, the bank central computer, the actual money dispenser, (i.e. a mechanical device able to dispense money). The magnetic card reader.

(10 Marks)

Now identify what **relationships** and **multiplicities** exist between the various objects in your sequence diagram and from this create a **UML class diagram** to represent those relationships and multiplicities along with any member functions, you can identify.

(13 marks)