THE AUSTRALIAN NATIONAL UNIVERSITY

Second Semester 2000

COMP2310
(Concurrent and Distributed Systems)

Writing Period: 3 hours duration

Study Period: 15 minutes duration

Permitted Materials: None

All your answers must be written in the boxes provided in this booklet. You will be provided with scrap paper for working, but only those answers written in this booklet will be marked. Do not remove this booklet from the examination room. There is additional space at the end of the booklet in case the boxes provided are insufficient. Label any answers you write at the end of the booklet with the number of the question they refer to.

Greater marks will be awarded for answers that are simple, short and concrete than for answers of a sketchy and rambling nature. Marks will be lost for giving information that is irrelevant to a question.

Name (family name first):

Student Number:

Official use only:

Q1 (25)	Q2 (25)	Q3 (25)	Q4 (25)	Q5 (25)	Q6 (25)	Total (150)

QUESTION 1 [25 marks]

(a) Define the term *SIMD*. (Certainly, you should say what the four letters stand for, but don't leave it at that!) Give an example of a SIMD computer.

QUESTION 1(a) **[3 marks]**

(b) Define the term *virtual shared memory*. Give an example of how memory addresses might work in such a system.

QUESTION 1(b) **[3 marks]**

(c) Consider a device driver with a lower half and an upper half. Which half uses the semaphore `wait` call? Which half is interrupt-driven? How do the two halves pass data between them?

QUESTION 1(c) **[3 marks]**

(d) Describe how the state of a Unix process changes upon calling the `exec()` system call. Also indicate what stays the same.

QUESTION 1(d) **[4 marks]**

(e) Explain what it means for Ethernet to be *CSMA/CD*. (Yes, you should say what the six letters stand for, but don't leave it at that!)

QUESTION 1(e) **[4 marks]**

(f) Compare and contrast Unix processes with Java threads.

QUESTION 1(f) **[4 marks]**

(g) Define the *producer-consumer* problem. Give an example of a typical operating system activity that is an instance of this problem.

QUESTION 1(g) **[4 marks]**

QUESTION 2 [25 marks]

(a) (i) One of the standard capabilities of a Unix shell is to interpret command lines of the form:

$$c_1 \;\mid\; c_2 \;\mid\; \ldots \;\mid\; c_n$$

where each c_i is a command consisting of a program name followed by the arguments for the program. Give a useful example of such a pipeline, made up of three or four commands, and explain what it does. Use only standard Unix programs, such as `grep`.

QUESTION 2(a)(i) **[3 marks]**

(ii) Explain how the shell program causes such pipelines to be formed. In doing so, show how the library calls `fork()`, `exec()`, and `wait()` are used. Also explain (again, with reference to the necessary library calls) how the shell directs the flow of data from each stage of the pipeline to the next.

QUESTION 2(a)(ii) **[12 marks]**

QUESTION 2(a)(ii) continued

(b) Semaphores can be implemented using message passing. In this scheme there is a special process, *S*, that acts as the holder of all the semaphores; client processes invoke the `signal` and `wait` operations by sending appropriate messages to *S* and then waiting for a reply. Show how this can be done with pipes under Unix. Describe how clients communicate with the semaphore process and suggest a suitable message format.

QUESTION 2(b) **[10 marks]**

QUESTION 3 [25 marks]

(a) Show how to use eventcounts to achieve a rendezvous between two processes.

QUESTION 3(a) **[5 marks]**

(b) Many CPUs have a special instruction whose sole purpose is to assist in the construction of *locks* (also known as *spin locks*). Describe such an instruction and show how it can be used to implement a lock. Also show the implementation of the corresponding unlock operation. (Your answer does not have to use the machine instructions of a real CPU, but the machine instructions you do use should be realistic. For example, you may use instructions that load and store values from/into individual memory locations.)

QUESTION 3(b) **[5 marks]**

(c) Semaphores and locks can each be used to achieve mutual exclusion. Contrast the two, in terms of CPU utilization, when used for this purpose.

QUESTION 3(c) **[5 marks]**

(d) Explain, with the aid of an example, what the operations on monitor condition variables (i.e. `wait` and `signal`) do.

QUESTION 3(d) **[10 marks]**

QUESTION 4 [25 marks]

(a) List and explain briefly the four conditions which must hold for a set of tasks to be deadlocked.

QUESTION 4(a) **[8 marks]**

1.

2.

3.

4.

(b) List and explain briefly the four options examined in lectures for dealing with deadlock.

QUESTION 4(b) **[8 marks]**

1.

2.

3.

4.

(c) Explain how the centralized algorithm for deadlock detection in a distributed system works.

QUESTION 4(c) **[4 marks]**

(d) The Chandy-Misra-Haas algorithm is an algorithm for distributed deadlock detection. Explain, with the aid of an example, how it works. (Hint: it uses probes.)

QUESTION 4(d) **[5 marks]**

QUESTION 5 [25 marks]

(a) Describe two-phase locking. Which of the ACID properties does it ensure? Can it deadlock?

QUESTION 5(a) **[6 marks]**

(b) Which type of transaction scheduler uses shadow copies? Why does it use them?

QUESTION 5(b) **[4 marks]**

(c) List and explain briefly the four ACID properties of transactions.

QUESTION 5(c) **[8 marks]**

1.

2.

3.

4.

(d) What is an uncommitted dependency? Give an example of one.

QUESTION 5(d) **[4 marks]**

(e) What are the two types of lock used in semantic locking? How are they are used?

QUESTION 5(e) **[3 marks]**

QUESTION 6 [25 marks]

(a) Describe, with the aid of an appropriate diagram, how MPI can be used to implement the bully algorithm. Your description should include the various MPI calls (including the appropriate parameters) and the tags you use, but you don't need to write all of the necessary code – just demonstrate that you know what you are doing.

QUESTION 6(a) **[10 marks]**

(b) What is Lamport's Algorithm? Describe (a) what it achieves, and (b) how it works.

QUESTION 6(b) **[5 marks]**

(c) List two algorithms that we examined in lectures which require Lamport's Algorithm or something equivalent to it.

QUESTION 6(c) **[2 marks]**

(d) Sun RPC and Java RMI are two examples of RPC facilities. Compare and contrast them.

QUESTION 6(d) **[6 marks]**

(e) Explain what it means to *marshal* the actual parameters of a remote procedure call.

QUESTION 6(e) **[2 marks]**

Additional answers. Clearly indicate the corresponding question and part.

Additional answers. Clearly indicate the corresponding question and part.

Additional answers. Clearly indicate the corresponding question and part.

Additional answers. Clearly indicate the corresponding question and part.