COMPUTER SCIENCE TRIPOS Part IB

Monday 31 May 1999 1.30 to 4.30

Paper 3

Answer **five** *questions.*

Submit the answers in five **separate** *bundles, each with its own cover sheet. On each cover sheet, write the numbers of* **all** *attempted questions, and circle the number of the question attached.*

Write on **one** *side of the paper only.*

1 Concurrent Systems

The runtime system of a concurrent programming language includes semaphore management and user-thread management modules. The design assumption was that the operating system on which the concurrent programs are to run supports only one process per address space. Outline the functionality and interaction of semaphore and user-thread management. [6 marks]

The runtime system is to be ported to run on an operating system which supports multi-threaded processes and symmetric multiprocessing.

(*a*) Describe how the two modules must be changed in order to support

- one kernel-thread per user-thread [5 marks]

- scheduler activations [5 marks]

State any assumption you make about the functionality of the operating system.

(*b*) Discuss the problem of concurrent execution of the semaphore management module and how this concurrency might be controlled. [4 marks]

2 Continuous Mathematics

Many important problems in mathematical modelling and scientific computing require the use of complex variables. Unfortunately, popular programming languages like C do not have a complex variable type, and so we must construct them from floating-point types. Assuming that the quantities a, b, c, d are all real numbers and $i = \sqrt{-1}$, resolve the following expressions, or explain the following operations, involving complex variables $\mathcal{Z}_1 = a + ib$ and $\mathcal{Z}_2 = c + id$:

(*a*) Let $\mathcal{Z}_3 = \mathcal{Z}_1\mathcal{Z}_2$. What is the real part of $\mathcal{Z}_3$, and what is its imaginary part? [2 marks]

(*b*) What is $\|\mathcal{Z}_1\|$, the modulus of $\mathcal{Z}_1$, and what is $\|\mathcal{Z}_3\|$, the modulus of $\mathcal{Z}_3 = \mathcal{Z}_1\mathcal{Z}_2$? [2 marks]

(*c*) What is $\angle\mathcal{Z}_2$, the angle of complex variable $\mathcal{Z}_2$? [2 marks]

(*d*) Express $\mathcal{Z}_1$ in complex polar form, not using the quantities a or b but rather the modulus $\|\mathcal{Z}_1\|$ and angle $\angle\mathcal{Z}_1$. [2 marks]

(*e*) Suppose that $\mathcal{Z}_1$ and $\mathcal{Z}_2$ both have a modulus of 1. Explain, with the aid of a diagram, how their product $\mathcal{Z}_3 = \mathcal{Z}_1\mathcal{Z}_2$ amounts to a rotation in the complex plane. Why is the multiplication of these complex variables reduced now to addition? Without using the quantities a, b, c, d, what is the value of $\|\mathcal{Z}_3\|$? [4 marks]

(*f*) Suppose that in complex polar form, $\mathcal{Z} = \exp(2\pi i/5)$. What do you get if $\mathcal{Z}$ is multiplied by itself 5 times? Give the simplest possible answer that you can. [2 marks]

(*g*) Consider the complex exponential function $f(x) = \exp(2\pi i\omega x)$. What function is its real part? What function is its imaginary part? [2 marks]

(*h*) If the above function $f(x)$ passes through a linear system, i.e. is operated upon by any conceivable linear differential or integral operator, what is the most dramatic way in which $f(x)$ can possibly be affected? [4 marks]

3 Further Java

Describe the facilities in Java for defining classes and for combining them through composition, inheritance and interfaces. Explain with a worked example how they support the principle of encapsulation in an object-oriented language. [15 marks]

What is meant by *reflection* or *introspection* in Java? Give an example of its use. [5 marks]

4 Compiler Construction

A programming language has expressions e with the following syntax:

$$\begin{aligned} e ::= x \ &\mid n \mid e + e' \mid e(e') \mid (e) \\ &\mid \texttt{let}\ x = e\ \texttt{in}\ e' \\ &\mid \texttt{letsta}\ f(x) = e\ \texttt{in}\ e' \\ &\mid \texttt{letdyn}\ f(x) = e\ \texttt{in}\ e' \end{aligned}$$

where f and x range over identifiers and n ranges over numbers. The three *let* variants introduce simple variables (`let`) and (non-recursive) functions whose variables are statically (`letsta`) or dynamically (`letdyn`) bound.

Using e itself (or any related language whose relationship to e is explained) as abstract syntax define an evaluator *eval* which, when given an expression e and an environment ρ, yields the value of evaluating e in ρ. The evaluator can be written in a language of your choice or in mathematical pseudo-code. [12 marks]

Explain carefully in one sentence each:

(a) the forms of value which *eval* may return;

(b) the form(s) of value which constitute the environment;

(c) the use(s) of environment(s) in `letsta` and in a call to a function defined by `letsta`;

(d) the use(s) of environment(s) in `letdyn` and in a call to a function defined by `letdyn`.

[8 marks]

Hint: because both `letsta` and `letdyn` functions may be applied using the same function call syntax, you may find it helpful to use separate forms of value for the two forms of functions.

5 Introduction to Security

Define *access control lists* and *capabilities*, and discuss their relative strengths and weaknesses. [5 marks]

Describe how the access control list mechanisms work in Unix. [5 marks]

You have been asked to build a funds transfer system in which a payment is authorised only once it has been approved by both a manager and an accountant at a bank branch. How would you implement this system using Unix security mechanisms as the foundation? [10 marks]

6 Data Structures and Algorithms

Describe Larsen's method of dynamic hashing that enables a record to be located on a disk given its key using just one disk transfer and only a modest amount of information held in main memory. [10 marks]

In Larsen's method each key has associated pseudo-random sequences of probe and signature values. Discuss what properties these sequences should have. Outline an algorithm that could be used to compute the n^{th} probe–signature pair for a given key. You may assume that the key is a character string. [6 marks]

Briefly discuss why Larsen's method is not used in most current filing systems. [4 marks]

7 Computer Design

Why is MIPS (millions of instructions per second) a poor measure of a computer's performance? [4 marks]

Explain why high-performance processors use pipelines to increase the MIPS rating and yet pipelines tend to increase the time to execute an instruction. [4 marks]

instruction fetch	decode/ register fetch	execute	memory access	register write back

With reference to the classic RISC pipeline above, explain what a data hazard is. [6 marks]

How are feed-forward paths used to reduce pipeline stalls? [6 marks]

8 Operating System Functions

FIFO, LRU, and Clock are three page replacement algorithms.

(*a*) Briefly describe the operation of each algorithm. [6 marks]

(*b*) The Clock strategy assumes some hardware support. What could you do to allow the use of Clock if this hardware support were not present? [2 marks]

(*c*) Assuming good temporal locality of reference, which of the above three algorithms would you choose to use within an operating system? Why would you not use the other schemes? [2 marks]

What is a *buffer cache*? Explain why one is used, and how it works. [6 marks]

Which buffer cache replacement strategy would you choose to use within an operating system? Justify your answer. [2 marks]

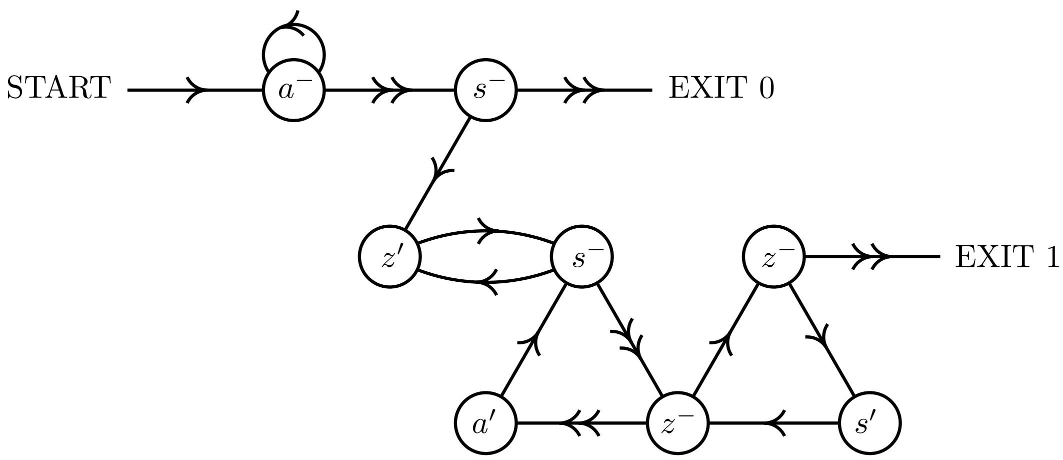
Give *two* reasons why the buffering requirements for network data are different from those for file systems. [2 marks]

9 Computation Theory

Define computation by a register machine, explaining the action of the program. [4 marks]

What is meant by the *current configuration* during a register machine computation? [2 marks]

In the following program, assume that register Z holds 0 initially. What is its effect?



[2 marks]

Show how to encode a general register machine program and the initial configuration of one of its computations into a pair of natural numbers. [6 marks]

Outline the design of a register machine that simulates a general register machine computation specified by a single natural number. Your machine should take appropriate action for all possible inputs. [6 marks]

10 Numerical Analysis I

Define *absolute error*, *relative error* and *machine epsilon* ε_m. Although ε_m is defined in terms of absolute error, why is it useful as a measurement of relative error?

[4 marks]

For a floating-point implementation with $p = 4$, $\beta = 10$, explain the *round to even* method of rounding using the half-way cases 7.3125, 7.3175 as examples.

Now consider $p = 4$, $\beta = 2$. What is the value of ε_m? What should each of the following numbers be rounded to, using *round to even*?

1.0101 1.1100 1.0011 1.1001 [6 marks]

Suppose $\cos 6$ is calculated by summing the series

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots$$

Estimate the value of the term with largest magnitude. Assuming this term can be computed with a relative error of 10^{-7}, what is the *absolute error* in computing this term? Hence, assuming $\cos 6 \simeq 1$, estimate the *relative error* in the computed value of $\cos 6$ to the nearest power of 10. [5 marks]

What are *guard digits*? How would you compute $\sqrt{x^2 - 2^{24}}$ if there was a danger that x^2 might overflow? If both x and powers of 2 are exactly represented, and guard digits are used, estimate the relative error in the result if $\varepsilon_m = 10^{-7}$.

[5 marks]