UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1999

MSc Degree in Computing Science
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Diploma of Membership of Imperial College

PAPER COMP II

ARCHITECTURE AND OPERATING SYSTEMS

Friday, April 30th 1999, 2.30 – 4.30

Answer THREE questions
Answer at least ONE question from Section A
Answer at least ONE question from Section B

For admin. only:
paper contains 4 questions

Section A *(Use a separate answer book for this Section)*

1a The following are the data declarations for an 8086 assembler program

```
.data
hi      db 'hot$'           ; whew
lo      db 'cold$'          ; brrr
h       dw  100             ; boiling point
l       dw -273             ; absolute zero
```

List the contents of the corresponding byte locations in memory in hexadecimal. (ASCII 'a' = 97 decimal, '$' = 36 decimal)

b Explain the function of the 1-bit status flags on the 8086 microprocessor. State which conditions are reflected by the C, O, S and Z flags.

c i) The following C function returns the maximum of two integer parameters

```
int max2(int a, int b)
{
    if (a > b)
        return a;
    else return b;
}
```

Write the equivalent subroutine in 8086 assembler. Your solution should use a stack frame and EQUate statements, preserve the contents of any registers used and include informative comments.

The following C function returns the maximum of three integer parameters

```
int max3(int a, int b, int c)
{
    return max2(a, max2(b, c));
}
```

Write the equivalent subroutine in 8086 assembler. Your solution should use a stack frame and EQUate statements, preserve the contents of any registers used and include informative comments.

The three parts carry, respectively, 30%, 30%, 40% of the marks.

2 a I have a microprocessor that has a basic 8 bit register structure. Currently, the only data types supported by hardware are unsigned and twos complement integers. I need to be able to represent signed floating point numbers with upto 4 significant binary digits.

i Describe a representation for floating point numbers that would meet my requirements but maximise the range of numbers available.

ii What are the *largest* Positive and Negative numbers that can be represented under your format and what are the 8 bit patterns that represent them?

iii What are the *smallest* Positive and Negative numbers that can be represented under your format and what are the 8 bit patterns that represent them?

iv What is the representation of zero in your format?

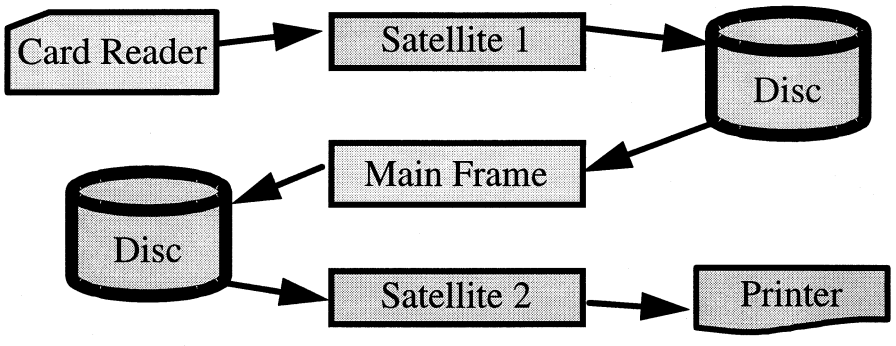
b Most microprocessor instruction sets support some form of subroutine call and return, the 8086 instructions for example are CALL and RET. The implementation of these instructions usually allows for recursive calls. Describe the micro operations that the control unit would need to perform to fetch *and* execute these two instructions.

The two parts carry equal marks.

Section B *(Use a separate answer book for this Section)*

3a Briefly describe *four* ways in which 3rd generation computer systems were an advance on the 2nd generation. (One sentence for each should be enough.)

b A 2nd generation directly coupled off-line batch system used two satellite computers to handle slow peripheral I/O such as card reader input and printer output. Each satellite communicated with the main frame via a disc.



A batch of 3 jobs has input, compute and output times (in seconds) as follows:

Job	Input	Compute	Output
1	5	20	10
2	10	20	5
3	10	15	10

The I/O times given are for the slow peripherals only. Disc transfer rates are 10 times as fast.

i) What is the turnaround time in minutes for job 3?

ii) Suppose the system is continuously executing batches similar to the one described. What is the throughput in jobs per minute? Explain why there is no need to include an interbatch gap in the calculation.

c A process executing on a system supporting paging makes references to the following page numbers:

0 1 2 3 0 1 4 0 1 2 3 4 2 0 1 2 0 3 4 0

If the page replacement policy is Least Recently Used (LRU) and main memory contains 4 page frames, specify which references cause page faults, and, where a page must be removed from main memory, give its identity.

You can give your answer in a form such as the following example, which applies to the First In First Out (FIFO) policy in 3 page frames:

```
                0 1 2 3 0 1 4 0 1 2 3 4 2 0 1 2 0 3 4 0
Faults:         F F F F F ...
Replaced page:      0 1 ...
```

d The space overheads introduced by paging arise from the following sources:

- The page table takes up space.
- A process is unlikely to fit into an integral number of pages, and therefore, on average, half the last page will be empty.

Show that if the average size of a process is s bytes, and that the size of a page table entry is e bytes, then the space overhead is given by:

$$\text{Space overhead} = \frac{se}{p} + \frac{p}{2}$$

where p is the size of a page.

Show that the page size which minimizes the space overhead is given by:

$$p = \sqrt{2se}$$

(*Hint:* differentiate with respect to p.)

What is the optimum page size if the average size of a process is 768K bytes and the space required for a page table entry is 8 bytes? What size is likely to be chosen in practice?

The four parts carry, respectively, 25%, 25%, 20%, 30% of the marks.

Turn over

4 The *Simple Kernel* referred to in this question is that of the lecture course, a multiprocess operating system that supports priority preemption, semaphores and timed delay.

a i) What is meant by *mutual exclusion* for two concurrent processes? Explain clearly what things are excluding each other.

ii) How can a *lock* be used to enforce mutual exclusion? What hardware feature is needed to implement locks correctly?

iii) Why does the Simple Kernel not use locks? What are the situations in which they are needed?

b Two concurrent processes P1 and P2 each have a critical region consisting of an instruction to add 1 to a shared variable x. Describe what extra code is needed to protect these critical regions using

i) a lock

ii) a semaphore

Remember to include data declarations and initialization. You may use any suitable language or pseudocode.

c Suppose the Simple Kernel is supporting three processes, H, L and I, with priorities high, low and idle respectively. H and L run code as follows:

```
H:  Delay(1000)        L:  P(s)
    V(s)                   V(s)
    P(s)
    V(s)
```

Initially, all three processes are ready and the semaphore s has value 0.

Complete a history of the execution of these processes in the following format:

Running process	Operation	Ready queue	Delay queue	s queue	s value
	Initially	H, L, I			0
H	Delay(1000)	L, I	H		0
:	:	:	:	:	:
:	:	:	:	:	:

(The first running process is H because it has highest priority. It calls Delay(1000) and as a result goes on the delay queue. L and I are still ready, and s still has value 0. The four columns on the right show the system after the operation, immediately before any rescheduling (i.e. call of Dispatch).)

d The Simple Kernel uses a simple priority-preemption scheduling strategy. Explain why this is unsuitable for a time sharing system used to support multiple online users.

Outline how the Simple Kernel might be modified so that it could incorporate time slicing.

The four parts carry, respectively, 30%, 20%, 25%, 25% of the marks.

End of paper