UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1997

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

COMPUTER SYSTEMS AND PROGRAMMING

Thursday, April 24th 1997, 10.00 - 12.00

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

For admin. only: paper contains 5 questions

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

1a. The following C function returns the minimum of two integer parameters

```
int min2(int a, int b)
{
if (a < b)
return a;
else return b;
}
```

i Using the LINK instruction to create a stack frame and saving the contents of any registers used, write an equivalent 68000 assembler subroutine.

ii Make a sketch of the stack immediately prior to the UNLK instruction, showing all the relevant contents and their size in bytes.

b The following C function returns the minimum of three integer parameters

```
int min3(int a, int b, int c)
{
return min2(a, min2(b, c))
}
```

Using the LINK instruction to create a stack frame and saving the contents of any registers used, write an equivalent 68000 assembler subroutine.

c Integers are usually represented using 2's complement. Describe how a 16 bit pattern would be interpreted under this scheme. What is the range of numbers that can be represented in 16 bits? What advantages and disadvantages does such a scheme have?

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

2 When describing microprocessors, the internal architecture size is often quoted, with typical values of 16 or 32 bits. However, the external architecture size is often omitted in such descriptions.

a What is meant by the internal and external architecture size?

b If a microprocessor has an internal architecture of 32 bits and an external architecture of 16 bits, what effect will this have on the various phases of the basic machine cycle?

c Carefully describe how a microprocessor, such as the M68000, responds to an externally generated interrupt. You should include in your description what would happen if a higher priority interrupt is received before the current interrupt is fully handled.

The three parts carry, respectively, 20%, 40%, 40% of the marks.

© University of London 1997

3a A C++ program includes the following constant declaration and type definition:

```
const int MAXIMUM_ARRAY_SIZE = 100;
typedef int int_array[MAXIMUM_ARRAY_SIZE];
```

and the following definition for the function `average(...)`:

```
/* Function to return the average of the
   first 'n' elements in 'list':          */
float average(const int_array list, int n)
{
    if (n > 0) return sum(list, n)/n;
    cout << "Error: incorrect call to 'average'";
    exit(1);
}
```

i Briefly explain the use of the identifier '`const`' in the definition of `average(...)`.

ii Briefly explain the meaning of the term *range bound error.*

iii Write a definition for the function `sum(...)` used in the program fragment above, which appropriately avoids range bound errors.

b i Briefly explain what is meant in C++ by a *recursive function definition.*

ii The function `reverse_up_to(...)`, whose first argument is of type `int_array` as defined in part (a) above, is defined in terms of an auxiliary function `reverse_between(...)` as follows:

```
void reverse_up_to(int_array list, int n)
{
    reverse_between(list, 0, n-1);
}
```

A call to `reverse_up_to(...)` results in the order of the values of the first `n` elements of `list` being reversed (so that, for example, with `n`=4 the array [10,20,30,40,50,60,70,..] is changed to [40,30,20,10,50,60,70,..]). Write an appropriate recursive definition of `reverse_between(...)`, using at most one local variable, of type `int`. Your definition should avoid range bound errors.

The two parts carry, respectively, 55% and 45% of the marks.

Turn over ...

© University of London 1997

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

4a i) What is meant by *batch processing?*

ii) When batch processing was introduced, it required new features in machine architecture and instructions. Explain why they were needed and what they were.

b How is an *associative store* used to improve performance in address decoding for paged virtual memory?

What would be the consequences of having an associative store that is too small, or too large?

c The Simple Kernel described in lectures is for many processes running on a single processor. Its mechanisms for achieving mutual exclusion for memory access would fail in a multi-processor system. Explain why, and suggest what hardware features would be needed to get round this.

© University of London 1997

5a What is a *context switch* in a multiprocessing environment? What are the *source* and *target* of a context switch?

Explain (without going into machine-dependent detail) how the Switch procedure in the Simple Kernel performs a context switch.

b An application program, written to run using the Simple Kernel presented in lectures, contains the following (and no other) calls of Create and Install:

```
Create(Stir, 500, high);
FOR i := 1 TO 5 DO
    Create(Cook, 500, normal);
    Create(Eat, 500, normal)
END;
Install(Mouse, 32);
Install(Cat, 40)
```

i) What processes will be in existence under the kernel and what code will they normally be executing? What distinguishes between them?

The following sequence of events takes place in these processes:

- A Cook process C is running, executing code in Cook.
- A Mouse interrupt occurs.
- Mouse makes a system call, the semaphore operation V, as its final command.
- A Cook process C' (different from C) is scheduled.

ii) At which point or points is there a context switch? For each one, identify the source and target processes, and describe approximately the values of the program counter and stack pointer which will be saved.

iii) Later on, C is rescheduled and there is a context switch back to it. Describe the actions that then lead to the resumption of execution of Cook, paying particular attention to flow of control through procedures.

iv) In the whole of the above sequence, between interruption of C and its resumption, at what points would interrupts get disabled or enabled?

The two parts carry, respectively, 20%, 80% of the marks.

End of paper

© University of London 1997