UNIVERSITY OF LONDON

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1999

BEng Honours Degree in Computing Part II
MEng Honours Degrees in Computing Part II
BEng Honours Degree in Information Systems Engineering Part II
MEng Honours Degree in Information Systems Engineering Part II
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute

PAPER 2.5 / I 2.5

OPERATING SYSTEMS II

Wednesday, April 28th 1999, 2.00 – 3.30

Answer THREE questions

For admin. only:
paper contains 4 questions

1 Try to be to the point for each case. 100 words per part should be the maximum.

a Is the code in the OS that does the actual scheduling part of a task or not? Explain your answer.

b What is spooling?

c How does scheduling for a system with strong real-time requirements differ from scheduling in multi-user systems?

d When should a busy-wait be used rather than blocking a process and rescheduling?

e Give the main difference between processes in MINIX and UNIX.

f Using procedure calls and pseudo-code, discuss the flow-of-control inside MINIX when a user wants to print a file (MINIX has *no* spooling facility).

The five parts carry, respectively, 15%, 15%, 15%, 15%, 15%, and 25% of the mark.

2 a Describe in which way the blocks used for a file are administered within a MINIX inode. Describe at least three other data contained in an i-node located on disk. Give also the data that are kept extra for open files.

b Desribe, step by step, the actions that take place when opening the file `/usr/home/test.c`. Focus on i-node handling and block search.

c What feature is called `mount`? What are the consequences for the design of a filesystem when this service is added to a system (e.g., discuss the features that UNIX provides to deal with `mount`)?

d Discuss the UNIX/MINIX file sharing model. In your answer, describe briefly the data that are kept in the datastructures involved, as well as how opening a file will change, and what the effect of the systemcall `fork` is.

e Give the names of at least four disk scheduling algorithms and describe them in a few words.

Turn over . . .

3 a Discuss the differences between *synchronous* and *asynchronous* send, and their implications for the kernel. The answer should not be longer than a few sentences (about a 100 words).

b An appartment building of many floors has an advanced elevator system, controlling many elevators that occur in groups in the building.

- A central computer `Central` deals with the distribution of the system, i.e., where to send which elevator carriage.
- Each single elevator has a special-purpose computer `Elevator`, that handles the buttons pressed by passengers inside the elevator (i.e., where they want to go), and passing that information on to `Central`.
- For each elevator, on each floor there is a special purpose computer `Floor`, that handles the buttons pressed, and passes that information on to `Central`. It also controls the local door.
- Each floor has a single button; pressing it will generate a message of type BUTTON-PRESSED to arrive in the `Floor` process. It will pass the request on to `Central`.
- Each elevator carriage has a set of buttons, one for each floor. Pressing one will cause a message to arrive with type FLOOR-BUTTON-PRESSED, with the field `msg-number` set to the desired floor number.
- When a carriage arrives at the right floor, a message will be generated (automatically) of type CARRIAGE-ARRIVES.
- Make sure that `Floor` does not close the door too quickly.

Using the following message passing primitives

```
send (processname, message) :

receive (processname, message) :

sendreceive (processname, message) :

processname = receiveany (message) :
```

give a pseudo-code outline implementation for the processes `Central` and `Elevator` and `Floor`.

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

4 a Explain the principle of a *mailbox*, and the discuss its behaviour in the context of both *synchronous* and *asynchronous* sends.

b Give the *process state diagram*. Give an extension that deals with the notion of a process being *swapped out*, i.e., whos process descriptor exists, but no longer has its code in memory.

c How does a *distributed operation system* (DOS) distinguish itself from a number of independent computers connected via a local network? Give some characteristics of the DOS *name / directory* service, that distinguishes it from a stand-alone system?

d Specify the four different layers in I/O handling, together with their function and abstraction mechanisms.

End of Paper