UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2001

BEng Honours Degree in Computing Part II
MEng Honours Degrees in Computing 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 C211

OPERATING SYSTEMS II

Thursday 10 May 2001, 14:00
Duration: 90 minutes
(Reading time 5 minutes)

Answer THREE questions

Paper contains 4 questions
Calculators not required

1) a. Explain why using a *server process* solves the problem of mutual exclusion.

b. What are the essential differences between *threads* and *processes*?

c. Explain why the kernel of a system implementing *asynchronous messages* needs to provide *buffering*.

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

e. Describe the effect of the Unix system calls **`fork`** and **`exec`** and how they interact.

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

The five parts carry, each, 20% of the marks.

2) a. Compare the *Unix* and *session* file semantics.

b. Let *bestand* be a file that occupies the disk blocks 12, 456, 60, 34 and 523 (in that order). Give the names of three methods that connect blocks to files, and show for each, in a diagram, how these methods would represent the file *bestand.*

c. Explain the functionality of a mount operation in a file system. Explain how, in a system like Unix, mount could be implemented, i.e. discuss the features that need to be added to the file system to adequately deal with mount?

d. Give, in a diagram, the Unix/Minix file sharing model.

The four parts carry, respectively, 20%, 35%, 15%, and 30% of the marks.

3) a. In the context of processes that run concurrently, when protecting access to critical regions, the following instruction can be used:

```
tsl reg,lock
```

What are '**tsl**' and '**lock**' here?

Give the implementation of protection of critical regions, i.e. give assembler specifications of both '**enter_cr**' and '**leave_cr**', using **lock**s.

b. Describe each of the following access control methods, and explain how they are used (for each use no more than 30 words)

- Protection Domain
- Access Matrix
- Access Control List
- Global Table
- Capability

c. Discuss how capabilities can be revoked in a distributed system.

d. Discuss the use of a *lock table*, and what the data it stores represent.

The four parts carry, each, 25% of the marks.

4) a. Specify the four different layers in I/O handling, together with their function and/or abstraction mechanisms.

b. Give an example of a situation in Minix where the OS caters for the fact that messages can get lost.

c. Give the design (pseudo code) for both the *synchronous send* and the *receive* procedure as used in a message passing mechanism.

d. Explain the chain of events (in terms of *messages*, *interrupts* and *scheduling*) that take place when a user process does a system call in Minix.

The four parts carry, each, 25% of the marks.