UNIVERSITY OF EDINBURGH

FACULTY OF SCIENCE AND ENGINEERING

M.Sc./DIPLOMA IN ARTIFICIAL INTELLIGENCE
M.Sc./DIPLOMA IN COGNITIVE SCIENCE
M.Sc./DIPLOMA IN COMPUTER SCIENCE
M.Sc./DIPLOMA IN INFORMATICS

# OPERATING SYSTEMS

Tuesday 23 April 2002

09:30 to 11:00

Examiners:
N. Chater (External)
N. Jennings (External)
E. Hancock (External)
K. Hawick (External)
R.N. Ibbett (Convener)
J. Oberlander (Co-Convener)

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

1. (a) The following program code represents an attempted solution to the Mutual Exclusion problem:

```
shared variable c1 = 1, c2 = 1;

process 1:                    process 2:
  while (TRUE) {                while (TRUE) {
    c1 = 0;                       c2 = 0;
    while (c2==0) { }             while (c1==0) { }
    CRITICAL SECTION 1            CRITICAL SECTION 2
    c1 = 1;                       c2 = 1;
  }                             }
```

i. Is mutual exclusion ensured? If so, explain how. If not, explain why not.

ii. What other problem might arise using this solution?

[*5 marks* ]

(b) Write out Peterson's modification to the above solution and explain informally why it is satisfactory. Indicate briefly a technique that might be used to prove it formally. (You are *not* required to give such a proof.) [*10 marks* ]

(c) Describe two examples of machine instructions (i.e. *not* **P** and **V**) that have been provided by computer architects which permit indivisible operations to be performed on shared variables to assist in solving the Mutual Exclusion problem. For each, write out program code for a process which protects a critical section and explain how it works.

What are the advantages and disadvantages of such an approach? [*10 marks* ]

2. (a) Explain briefly what is meant by each of the following terms:

   i. Process
   ii. Thread
   iii. Interrupt
   iv. Exception
   v. Signal

   [*10 marks* ]

   (b) *Recall that the* `mprotect` *Linux system call allows a program to set read/write access permissions on individual virtual pages of its process space.*

   Using `mprotect`, how might a process be able to monitor which virtual pages were used by an application? [*7 marks* ]

   (c) *Now assume the existence of a* `sleep` *system call that suspends the execution of a thread for some number of milliseconds.*

   How could your scheme of part (b) be extended, using such a `sleep` system call, to produce a *page-usage trace* of the running application which would monitor the virtual pages it uses in *every* successive time-slice? [*8 marks* ]

3. (a) Describe problems which virtual memory segmentation overcomes and how it does so. *[6 marks]*

   (b) Briefly describe the segmentation system on the Intel Pentium processor and discuss its effectiveness. *[7 marks]*

   (c) What is meant by the term *Rings of Protection*? Explain how this is implemented on the Pentium. *[6 marks]*

   (d) Describe how paging can also be incorporated into an address translation mechanism in addition to segmentation. What benefits does this bring? *[6 marks]*