Examination Feedback for EEE412 – Systems Architectures
Autumn Semester 2010-11

Feedback for EEE412 Session: 2010-2011

Feedback: Please write simple statements about how well students addressed the exam paper in general and each individual question in particular including common problems/mistakes and areas of concern in the boxes provided below. Increase row height if necessary.

General Comments:

Nothing particular apart from some handwriting is very difficult to read!

Question 1:

a) Generally OK although only a few identified the real problem with device drivers: the security vulnerabilities due to running in systems mode.

b) A number of students mentioned "Hardware Virtual Machines" as one of the three types of VM, referring to the original IBM implementation on the CM/VMS system. In fact, this is largely of historical interest.

Much of the discussion on software emulation vs. VMs failed to mention the key fact that VMs execute instructions which are native to the hardware whereas in software emulators, the instructions are 'translated' from some incompatible format. It is the *translation* at the machine-code level that is the critical distinction.

Question 2:

a) Generally well answered.

b) Many failed to identify that interrupts are processOR entities, not part of the operating system.

c) Generally done OK although many did not identify the key issue: Deadlock will only occur is there is a context switch between P_1 opening f_1 and P_2 opening f_2… which is actually a rather rare outcome.

Suggestions in the final part varied but the question did actually say "How can the *system* be modified…" so solutions based on semaphores were not really appropriate. (If you were going to change P_1 and P_2, the easiest way of preventing deadlock is simply to swap the order of the fopen() instructions in, say, P_2.)

Question 3:

a) Generally well answered.

b) Generally well answered, but many did not mention the two (or more, for multi-level page tables!) memory accesses required to access one memory location.

c) Generally well answered, except few properly addressed the final part about the efficiency of the method,. Namely the effect of the period between clearing the extra bit.

d) Generally well answered. Many even suggested the improvement of examining the extra bit (c) to prevent discarding old but recently-used pages.

Question 4:

a) Many thought that events triggered interrupts. This is not strictly correct: Events which originate from hardware (e.g. a mouse click) typically trigger an interrupt which is handled by the OS which then routes a message to the window which holds focus. But the destination application does not perform any interrupt handling – it polls its message queue. Messages can be sent from one application to another (or an application can place a message on its own message queue!) which does not involve any interrupts at any point.

The structure of the message was not mentioned by most.

b) The second part about how the appropriate event handler is invoked (via a LUT) was not well answered, in general.

c) Well answered.

d) Generally well answered, but the action is not triggered by the combination of left mouse down and left mouse up events, only the latter. The left mouse up handler is usually the default which redraws the “OK” button with a shadow around it making it look like the button has been pressed into the screen.

e) Generally well answered apart from a number who omitted to mention low-level functionality, like the message loop poling, which is embedded in a base class.