p1 q3
SMH

Operating Systems (Part IA): Tripos Questions ~~1998~~/2000

Paper 1 Question 4 (10 Marks)

Operating systems need to be able to prevent applications from crashing or locking up the system, or from interfering with other applications. Which three kinds of hardware support do we require to accomplish this? [6 marks]

How do applications request that the operating system perform tasks on their behalf? [2 marks]

What could we do if we did not have the requisite hardware support? [2 marks]

Answers to Paper 1 Question ~~4~~ 3

Required Hardware Support

Two marks for each of the below:

First and foremost we require processor support for at least two modes of operation: user and kernel/supervisor modes. This allows us to have instructions which may only be used in supervisor mode (such as direct access to I/O devices via `inb` etc.). 2

Secondly we require memory protection hardware to prevent applications from trashing each other / the OS. Configuring this hardware can only be done in supervisor mode. 2

Thirdly we require a timer device so that we can implement preemptive scheduling. Otherwise an application can execute `while(1);` and lock the system. 2

Interface with OS

Applications request system services by means of *system calls*. These are implemented by using special instructions which cause a secure transition to supervisor mode. 2

Coping without hardware support

Two marks for any plausible solution here. 2

i.e. use language-level techniques

p1 q 11
SMH

Main thing we could do would be to use language-level techniques. E.g. we could require that all applications were written in a "safe" language like java, and only load such applications. This would involve severely constraining the interface to the user (i.e. no "poke" command!), but might just work.

Paper 1 Question 11 (20 Marks)

Describe with the aid of a diagram the life-cycle of a process. You should describe each of the states that it can be in, and the reasons it moves between these states. [4 marks]

What information does the operating system keep in the process control block? [4 marks]

What information do the shortest job first (SJF) and shortest remaining time first (SRTF) algorithms require about each job or process? How can this information be obtained? [2 marks]

Give one advantage and one disadvantage of non-preemptive scheduling. [2 marks]

What steps does the operating system take when an interrupt occurs? Consider both the programmed I/O and DMA cases, and the interaction with the CPU scheduler. [4 marks]

What problems could occur if a system experienced a very high interrupt load? What if the device[s] in question were DMA-capable? [4 marks]

Answers to Paper 1 Question 11

Process Life-cycle

Looking for a five-state (or more) picture like that in notes. The states involved will be something like:

1. *New*: state for newly created processes. Moves from here to *Ready*.
2. *Ready*: state for any runnable process (i.e. one which could use CPU if granted it). Typically many processes in this state. Moves into *Running* when dispatched by scheduler.
3. *Running*: state for processes currently executing; only one of these per CPU. Moves to state *Ready* if preempted, *Exited* if exits, or *Blocked* if waits on I/O or some other event.
4. *Blocked*: state for all those processes currently awaiting an event of some sort. Typically many processes here. Moves into *Ready* once the requisite event occurs.