UG Exam Performance Feedback
Second Year
2017/2018 Semester 1

COMP25111 Operating Systems

Jim Garside
Toby Howard

Comments Please also see the attached report.

With some earlier marks as 'give-away' on coursework, the paper was intended to be a challenging test. The results statistics show a very wide marks distribution, from a few in single-figure percentages to a few in the 90%+ range.

Questions were marked anonymously and (mostly) independently so therecould be no influence of one good/poor answer affecting judgement of others; the totals were only assembled when everything was complete.

Clearly some concepts, which should be fairly simple and which are generally important, are not appreciated by many students. Perhaps most shocking was the structure of a page table (Q12) which had certainly been introduced; another figure has already been added to the website to try to help with this in future.

Another generally worrying trend is the lack of 'sanity checking' of answers - not relating the answer written to a real system. Q10 gave some good examples of this: given 4 GiB of physical memory (roughly 4,000,000,000 locations) how could 200000000000000000000 4KiB pages fit? On the other hand does 4 pages (the lowest but other answerswere of similar magnitude) yielding a maximum user space of 16 KiB sound -credible- in a modern computer?

- -

1. Briefly describe two advantages of dynamic shared libraries.

Most people understood this. Some people mistakenly thought the purpose of such libraries was to share information between processes.

2. Briefly explain two ways in which a hard real-time process scheduler is likely to differ from the scheduler in a desktop computer.

Easy question, well-answered.

3. Why might a user-mode library buffer data from a character stream on its way to being written into a disk file?

Easy question, with a surprising number of people having no understanding of the fundamental reason -- efficiency.

4. Why might it be useful to be able to prevent instruction fetches (execution) from memory whilst still being able to read it?

Some good answers but some revealing misapprehensions too. Some students are clearly confused by the difference between addressable memory (RAM) and files.

Memory speeds/interaction confused. 'Slowdown' was mentioned; even if this were relevant to the question, why would checking that this was an instruction fetch be different from any other checking. Other topics such as race hazards, atomicity and instruction dependencies all appeared more than once.

5. In a Unix filing system, explain how the following are implemented: a soft link; a hard link.

About 1/2 the class could explain this correctly.

6. A running process on a multiprocessor desktop workstation needs a large block of data which needs to be fetched across a network interface. Explain the sequence of operations which are likely to take place to complete this operation. Your answer should include the roles of:

Device driver
DMA
Interrupt
Process state
Scheduler
System call
(Listed above in alphabetical order.)

The role of the scheduler was enlarged on by many; it is subservient, not the master! a process needs to be running to do something (even if in a syscall at the time). Many variants of the scheduler interrupts ..." The scheduler doesn't STOP things autonomously, it chooses what to do next, when allowed. There may be a pre-emptive (time-slice) interrupt which calls the scheduler but this is not the only way of reaching it.

'The process becomes 'blocked' before finishing its set up', in many instances. Sanity check!

There was a a lot of 'not answering the question' here; just mention lists of keywords and hope some apply. This is not a good approach to an exam, which is testing understanding, not memory.

Device drivers (SW) and peripherals (HW) seem to get confused.

Quite a lot of "buffer"ing; needs clarification.

7. Explain the difference between a file's attributes and its contents.

A very straightforward question, well-answered.

8. List three different attributes a file might have, in each case with a brief explanation of what they represent.

A very straightforward question, well-answered.

9. In a Unix-like file system, describe how a directory entry is related to a file. Explain how this structure enables one file to exist in several different directories at the same time.

There seemed to be a lack of understanding about this, with only a small number of correct answers.

10. Imagine a computer with a 32-bit address space, which is divided into 4 KiB pages. (Each byte has a unique address.) The computer is limited to 4 GiB of physical memory. What is the maximum number of pages a single process could have?

Mostly good. Some Gi/Mi confusion. (Really ought to know the difference by now!)

A few people failed the sanity check of their answer: the given numbers are pretty realistic for a (slightly dated?) system. Extreme values ranged from 4 to 20000000000000000000.

11. For the same computer in Q10, assuming a single-level page table, suggest a sensible size for each page table entry. Outline the various information which might be kept there; include estimates of the number of bits used for each function. (You may need to make some reasonable assumptions, which should be justifiable.)

Q11-13 follow on from Q10 and a student's previous answers were considered in marking.

The basic question was intended to be very straightforward - starting with the number of bits used for address translation - with the challenging parts trying to remember/deduce the other sort of information which might be appropriate. (There is no definitive- answer so a reasonable selection of possible suggestions would earn credit.) Unfortunately, far too many candidates stumbled at this first 'hurdle'.

Many of those answering otherwise sensibly then failed to consider the mapping of the entry onto a physical machine. An answer like 25 bits' may cover the size used but it would be unpleasant slow!) to pack into memory. Rounding to a sensible size is also appropriate. This was only lightly penalised though.

Errors describing a 20 bit field as "2^20" bits were forgiven - it could be a language problem - although the difference is fairly blatant!

Some page 'information' was credited despite it being unlikely to be kept in the 'hardware' page information: an example would be pinning ('do not evict') which is wanted by the paging software but irrelevant to the page translation. It was felt that this distinction was too subtle to penalise.

The virtual address is the -index- to the table, not kept in the table as several people thought.

12. For the same computer in Q10, a particular process uses 5 different pages in its virtual space. Given a single-level page look-up scheme, state how much memory the page table for this process would occupy.

Much misunderstanding as to the structure of a page table. It's an *array*. All the -possible- pages are present and the size is set by the architecture. This seemed to be missed by many candidates who had correctly mentioned the entry containing validity info. in Q11.

The question was intended to be simple, joining (multiplying) the answers from Q10 and Q11; it seems it was found surprisingly hard. The '5' is not important at this point. By far the most popular answer was 20 KiB - which is the (user) memory made available, NOT the page table which is the associated metadata.

Maybe should have emboldened "page table" in the question but, even on rereading it seems unambiguous.

13. For the same computer in Q10, would a two-level page look-up scheme use more or less memory? Explain your reasoning. Give an estimate of how different (if at all) this would be. Exact numbers are not required.

This was intended to inherit the "five pages" from what ended up being Q12. This was not clear after the paper had been coerced into Blackboard and should have been more explicit. Allowance was made for this when interpreting answers although, in the event, it rarely would have affected marks.

Exact quantities were not required but some sense of the order of magnitude is appropriate. This was rarely given.

Many thought two-level tables save time (two memory look-ups being faster than one?!). This reveals some misunderstanding.

There is a general, fundamental principle (also used in, e.g, i-nodes) of using a (pruned) tree to save space, (statistically)which has not been understood by most.

14. What is meant by a "memory mapped" peripheral device?

This question was poorly answered.

15. In a system using paged memory management, what properties might a page associated with I/O devices have?

The intention of the question was to combine address page properties with the needs of I/O.

The idea of memory mapped I/O is clearly unclear to many candidates.

'Pinned' is probably the most popular choice ...

A lot of "probably"s when there is certainty (forgiven).

16. For each of the following types of structure, choose two distinct examples of how they might be used in a typical operating system and say why each application is appropriate.

Linked list
Multi-way tree
FIFO queue

Quite a lot on FIFOs for eviction - possible but not the most effective. Also ready queues - which neglects priority etc. Dubious.

A linked *list* is not the same as a (single level) pointer.

It is possible to be too general "A linked list can be used as a memory data structure." (I exaggerate, but not by much is some cases!) does not demonstrate particular knowledge; some specificity is required.

17. Choose two different examples of exceptions which cause kernel entry and justify why they are required.

Well-answered, but often short on justifications, or just omitted.

18. Choose two distinct examples of virtualisation provided by a typical workstation operating system and justify their use.

About 1/3 of the class mis-read the question, describing virtualisation provided by hypervisors/containers. They are not part ofÂ "a typical workstation operating system"; the question was referring to virtualisation such as virtual memory, time-slicing, and presenting a unified view of disparate devices.

UG Exam Performance Feedback
Second Year
2017/2018 Semester 1

COMP25111 Operating Systems

Jim Garside
Toby Howard

Comments Please see the attached report.

Exam Feedback

This feedback is based on the **COMP25111** exam that took place in **2017/18**.

This report is based on Blackboard Item analysis report that can be found in your course unit (Course Tools > Tests, Survey and Pools – right click on the name of the test/exam and select Item Analysis).

Difficulty

Average Exam score: **30.76 (51.27%) +/- SD 19.8%**

Average Exam Time: **1 hr 48 min**

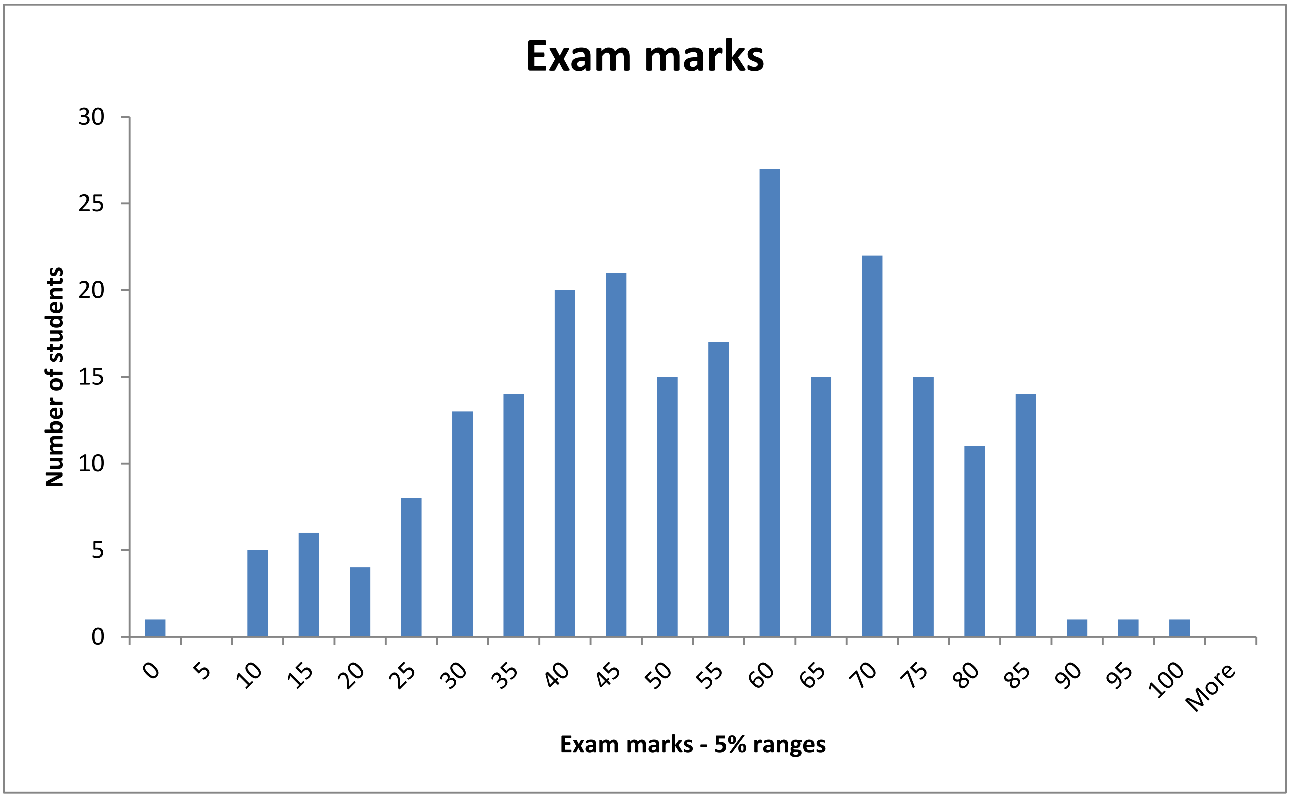
Below is how Blackboard defines question difficulty.

Difficulty: The percentage of students who answered the question correctly. The difficulty percentage is listed along with its category: Easy (greater than 80%), Medium (30% to 80%), and Hard (less than 30%). Difficulty values can range from 0% to 100%, with a high percentage indicating that the question was easy. Questions in the easy or hard categories are flagged for review.

2	**Easy Questions**
13	**Medium Questions**
3	**Hard Questions**

You can identify specific Questions using the Item Analysis Tool in Blackboard.

Distribution of Marks



Discrimination

Discrimination: Indicates how well a question differentiates between students who know the subject matter and those who don't. A question is a good discriminator when students who answer the question correctly also do well on the test. Values can range from -1.0 to +1.0. Questions are flagged for review if their discrimination value is less than 0.1 or is negative. Discrimination values can't be calculated when the question's difficulty score is 100% or when all students receive the same score on a question.

18	**Good questions**
0	**Fair Questions**
0	**Poor Questions**
0	**Cannot Calculate**