Operating Systems

Revision

Exam Content

- 100 marks, 120 minutes
- 8 MCQs

Questions?

Week 1 ctd

- What is a Process Control Block?
- What does it contain? Why does it need these things?
- Explain what happens when I switch from one process to another.
- What are the three main process queues and what do they do?

Using these slides

How I think

- The purpose of these slides is to give a sense of the scope of the assessable material and level of analysis required for the exam.
- Many questions here will not be covered in the exam (unsurprisingly, it is a subset, the exam is not 8 hours long).
- Exam questions will be more precise.

Week 1

- What is an OS and what does it do?
- What is meant by virtualisation? What is ‘virtualised’?
- What benefits does the OS providing abstraction give?
- Give an example of OS abstraction…
- Why must OS’s provide protection? Protection from what?
- I have two processes on one machine… what could go wrong?
- What is a process? What is a program? What is a thread?
- What is the difference between a program, process and thread?

Week 1 ctd

- What are some ‘pretty important’ registers that processes definitely need? And why?
- What does starting a process look like? What does it mean?
- How do you kill a process? What does that mean?
- Kernel mode? User mode? What should be run when?
- What is **direct execution**? What is the alternative and why might we bother with **direct execution**?
- What about **limited direct execution**? What is the word **limited** doing here?

Week 1 ctd

- What is a trap? What does interrupt vs polling mean?
- Should we use polling or interrupts?
- Explain how a system call works. What is a system call?
- What are registers for and why are they important for an OS?
- What is a stack pointer? A program counter?
- How do I decide when to switch processes?
- Why do I not let processes decide when to switch?
- How do I stop a wayward process?

Week 2 - Virtualisation

- What is virtualisation?
- Schedule vs Dispatch?
- What might I want to maximise/minimise when scheduling and why? Turnaround time? Response Time? Throughput?
- Oooh… I have some jobs: calculate stuff! Make a GANTT chart!
- RR, STCF, SJF, STCF, FCFS, …. There are many practice them…

Process	Priority	CPU	IO	CPU	IO	

Week 2 - ctd

- RR, STCF, SJF, STCF, FCFS… tell me about:
  - Scheduling, switching, bad, good, convoy, time slice times
- What is a Multilevel Feedback Queue (MLFQ) and how does it work? What problem is it attempting to solve?
- Lottery scheduling? What is the idea? How does it work?
- Completely Fair Scheduler (CFS)? What is? How works?
- What is priority for? Pre-emption?
- What problems does each algorithm solve/cause.
- Multi-CPU, what works, what doesn’t global queue/local queues?

Week 3 - Memory

- I have bytes, I have bytes, how many Megabytes are in a byte?
- Uniprogramming, multi-programming
- Draw a stack, draw a heap, explain how it works…
- Why heap annoying?
- What does the Memory Management Unit (MMU) do?
- How do we stop two processes from colliding with each other in memory?

Week 3 - ctd

- Base and bounds. How works? What does? Good/bad?
- What issues could arise from poorly written system calls when considering process memory access?
- Segmentation. How works? What does? Good/bad?
- Multi-threaded segmentation!!
- Paging!
- Internal/external fragmentation (this will come up a few times)
- Big page/little page (good/bad)

Week 3 - ctd

- Page tables and addressing them…
- Single level/multi-level page tables
- See this table… decode an address

Page number	Frame number
0x??	0x??
0x??	0x??
0x??	0x??
0x??	0x??

Week 4 – Paging and stuff

- What is the big problem with a naïve implementation of paging?
- What is locality? How does it apply to paging?
- What is caching? How can we use it to improve… <insert topic>?
- Translation Lookaside Buffer (TLB). What does? How works? Why bother?
- Miss rate!

Week 4 - ctd

- What is page fault? How is resolved? Demand paging?
- Page fault happens, what need do?
- Page replacement. TLB replacement. (very similar but with some minor nuance differences).
- Optimal, FIFO, Random, LRU
- Here are some numbers… do a thing

1, 2, 4, 2, 5, 1, 6,1, 4, 3, 2, 1, 1, 3

Week 4 - ctd

- What is thrashing?
- Guard pages?
- Fork: Copy-on-write? Distributed machines?
- Dirty bits? Present bits?

Week 5 – Concurrency

- Atomicity. What is? Why problem?
- Race condition? Show example.
- Spin lock, here code, how break?
- Lock/Mutex? What is? What does? How works?
- Here some code… where break? Please fix.
- Disable interrupts? Does work? Why bad?
- Peterson’s algorithm? How works?
- What is a memory barrier? Why useful?

Week 5 – ctd

- TestAndSet? What does? How works? How can use to make good?
- CompareAndSwap? What does? How works? How can be used to make good?
- Ticket locks? What are they and what problem do they try to solve?
- Why do we sometimes have a ‘guard’ lock?
- Which is better? Spin lock or blocking process?
- Condition variables? Explain the logic.
- Write a simple multi-threaded program which uses mutual exclusion (of some form) – pseudocode
- Producers and consumers? What is going on here?

Week 6 - Semaphores

- Semaphore?
- Semaphore vs Lock/Mutex vs Condition Variable
- Deadlock
- Livelock
- Here is some code… deadlock??
- Deadlock has four conditions… what are they? Why do you need all four?
- Do table!

	Allocation				**Max**				**Available**			
	A	**B**	**C**	**D**	**A**	**B**	**C**	**D**	**A**	**B**	**C**	**D**
P_0	0	0	1	2								
P_1	1	0	0	0								
P_2	1	3	5	4								

Week 7 – I/O and Hard Drives

- Polling vs Interrupts for I/O?
- Port-mapped I/O? Memory-mapped I/O?
- What is direct memory access (DMA)? Why good?
- Device drivers? What are they? What do you need in a device driver?
- Everything is a file? What does this mean? Give an example of a non-file as a ‘file’.
- How can we communicate with I/O devices? Blocking? Non-blocking? Asynchronous

Week 7 - ctd

- Access patterns of I/O, what design decisions would this impact?
- Hard disks, how do they work?
- Seek, rotate, transfer!
- What are the main limitations of HDDs?
- Why might a HDD have a buffer? What benefits/drawbacks?
- Solid State Drives (SSDs). What problems do they fix?
- How does an SSD work?
- Wear levelling? How can it occur and how to mitigate?

Week 7 - ctd

- Many hard drive writes
- How to order the requests?
- FCFC, SCAN, LOOK,

Week 8 - RAID

- RAID – 0, 1, 4, 5
- What are they?
- What benefits do they grant? When do they fail?
- RAID – 0
  - How best to lay out? Why? IS reading/writing faster?
- RAID – 1
  - How affects read/write time? Disk failures?

Week 8 - ctd

- RAID 4
  - What is parity? How to calculate efficiently?
  - Why is writing so slow?
- RAID 5
  - What is different to RAID 4?
  - What advantage does that provide to reading and writing?

Week 8 - ctd

- What is a file? What is a file path?
- File permissions, what are they, how do they work?
- Access control lists, what are they, what advantages do they convey
- Hard links, soft links. What are they. I delete and rename some files with links involved… what happens?

Week 8 - ctd

- File descriptors. What are they?
- File descriptors and forking? What happens?
- I-nodes. What are they? What is inside? What do they store? How are they arranged? Big files? Directories and I-nodes?
- What is mounting?

Week 8 – ctd

- Contiguous allocation
- Extent allocation
- Linked list allocation
- File Allocation Table (FAT)
- Indexed allocation
- Multi-level Indexed Allocation
- Multi-level Indexed Extents

Here is a design for a file system. What problems are there with this style of file system?

Week 10 – FFS & Crash Recovery

- What is the structure of FFS vs a regular file system.
- What is the best way to distribute data between groups in FFS?
- How to deal with big files in FFS?
- What are sub-blocks/fragments? Why good/bad? What problem are they attempting to solve?
- Why is having all memory blocks layed out in contiguous order bad for HDD?

Week 10 - ctd

- What is FSCK?
  - How does it work?
  - How do you know the computer has crashed?
  - Here is an example of an inconsistency. Can it be fixed and how?
- Journalling?
  - What are transactions? Checkpoints?
  - Here is a scenario, there is a crash… what can be fixed. What must be discarded.
  - Data Journalling, ordered journalling, writeback journalling, why might you use one of these, why and what could go wrong?

Week 11 - LFS

- What is a Log-structured File System?
- What problem is it trying to solve? How does it achieve this?
- Should segments be big or small? What advantages/disadvantages does each entail? What governs this decision?
- How does LFS handle i-nodes? What is an imap? What is the checkpoint region?
- When a file is updated in LFS, what happens? (note what happens to the old files)
- Garbage collection? How to administer?

Good Luck