

## Drop in notes

Olaf!!! → Compiled lecture slides relevant to exam, but he's not going to upload them 😞

### OS Basics

- Virtualising machine
  - Isa
  - Threads
  - Concurrency
  - Address spaces
  - ???
  - Sockets
- Protection
  - 2 methods: user/kernel mode

### Virtualisation

- CPU
  - 3 states of process: running ready blocked
  - Transition from user to kernel - system calls
    - Trap table - special instructions to reset to the kernel
    - Program in user mode cannot turn off interrupts
  - Multitasking
    - 2 ways to achieve: 1. timer interrupts → context switch, 2. Yield → program voluntarily gives CPU away
    - Store:
      - PCB - see slide Multiplexing processes: The Process Control Block
- Memory
  - Process address space: typical structure
  - Share space in memory: memory management unit
    - Converts virtual and physical address spaces
    - Base + bounds
      - Kernel mode → physical address
      - User mode → check if below bound, then add base
      - Buffer overflow → overflow buffers overwrite other memory - this happens inside a process but not to the whole memory storage because of memory isolation (virtualizing of memory)
  - Segmentation
    - Separate base and bound segments
  - Fragmentation
    - Internal vs external
  - Paging
    - Mapping virtual page no to physical frame no
    - Uses page tables to store map

- Pros: no external fragmentation, free pages are equivalent
- Cons: page tables are too big (there's more, see slide Paging Pros and Cons)
- TLB
  - Table page look-up needs to read twice, to reduce this, we have TLB
  - Provides spatial and temporal locality
- Multi-level page tables
  - Why do it? Benefits?
- Page fault
  - Demanding page
  - Process of a page fault
- Page replacement
  - Cold starts (compulsory misses)
  - Capacity misses
- FIFO: replace the one in the memory the longest
- LRU: the least recently used is replaced
- Random: just randomly replace a page
- Optimal: the furthest away used is replaced

#### Scheduling (Part of CPU)

- Workloads, schedulers, metrics
- Turnaround time: complete - arrive
- Response time: first run - arrive
- FIFO
  - Convoy effect - small tasks build up while long task run
  - Negative effect for FIFO
- SJF
  - We don't know how long future jobs will take
  - Needs pre-emption to interrupt long jobs when shorter jobs arrive
- STCF
  - Check textbook for definition - considers shortest time overall instead of shortest time left according to Olaf
  - Cruz calls this SJF preemptive, if she calls schedulers by different names and you are not sure what it is, write your assumption and its definition.
- RR
  - Horrible turnaround time, good response time
- I/O aware
  - While a process is using I/O, other processes can use the CPU
- MLFQ
  - Mac
  - Rules
    - Priorities: higher priority runs, same priority RR
    - Set priority:
      - Long jobs tend to drop in priority (starvation)
- CFS (mentioned, no slide)

- Linux

## Concurrency

- To achieve concurrency we need threads
- Locks - mutex
  - Synchronising
    - Condition variables: process checks variable, if condition not fulfilled, release lock and go to sleep
      - Wait and signal (signal wakes process up from sleep)
      - producer/consumer
        - While loop is needed to check condition over and over again because the condition can change
        - Consumer can consume in the order the product is produced, but not guaranteed
  - Semaphores
    - Integer values initiated to a specific value
    - Wait decrements value
    - Value is non-negative (can be a negative in text-book, but shouldn't be in exam)
    - readers/writers problem
      - Consider shared database
      - Writers can read and modify
      - Readers can only read
      - Critical section should be locked when a writer is writing
      - Approaches
        - Single mutex (lock when anyone access)
          - Not efficient
        - While reader in section, no writers can enter
          - Writer never gets there
        - Process readers and writers according to their arrival
  - Equivalence claim
    - Exam: don't need to know proofs but need to understand that they are equivalent
  - Deadlock
    - Circular dependency
    - Mutual exclusion
    - Hold and wait: holding a mutex and not releasing it while waiting for another mutex
    - No pre-emption: cannot interrupt

## Persistence

- Protocol variants
  - Status checks
    - Polling

- Interrupts
- Data:
  - PIO
  - DMA
  - Adv and Disadvantage
  - Kernel needs to manage/protect it
  - Know how DMA work
- Control
- Disk
  - The Amazing Magnetic Disk
    - Explain terminology such as platters, etc.
  - Solid state disks
    - Basic idea: electric charge trapped between insulation layers, write to different locations to avoid wear and tear
    - Write:
    - Erase:
  - RAIDs
    - Bunch of disks
    - RAID-0
      - Striping, using many disks as one disk
    - RAID-5
      - Parity bits rotated: calculated by XOR
  - Inodes
    - Stores metadata
    - Has pointers to blocks
  - Superblocks
  - Bitmap blocks
  - Multilevel indexed files
    - (e.g. 12 direct blocks)
    - Direct blocks goes straight to a data file
    - Pros and cons
  - Aging: free list makes contiguous chunks hard to find - fragmented over time
  - Disk aware
    - Cylinder route
      - Similar stuff similar side
    - Smart policy
  - Creating a file
    - Find free blocks
    - Find free inode entry
    - Find directory to insert
    - Mark as used in many places (map, inode, dirent)
  - Journaling
    - Write ahead
    - Log the writes first

- 2 variants
  - Writes metadata only
  - Writes all data
- Crash recovery
  - Use commits
- NEED TO KNOW UNITS!!! (Time, memory)
- Be aware of MCQ that asks what is "FALSE"? (This means read the question correctly)