concurrency

Algo lab 4 Notes

Critical

Critical Utilization -

- Percentage of time the CPU is busy
- too saturated w/ tasks
- Struggles to handle new tasks EFFICIENTLY
- Performance degregation, lag, instability
- there is no fixed value / percentage
- common warning load is 80% - 90%
- but in-practice depends on system, application type & desired user experience
- A sustained load above crit would suggest A Bottleneck
- CPU cant keep up
- Optimization of software may be needed
- or upgrade hardware

Default Behaviour of exceeding load is a slowing down of the system.

- but it can also lead to system instability & compramised data integrity
  → unreliable or incorrect results

① Resource Starvation & Unpredictable Scheduling
caused when pushing past critical threshold

OS scheduler may struggle to give tasks (processes & threads) the CPU time they need

Some critical operations have timeouts. May end before CPU has completed = WRONG State / Data

Race conditions is an underlying flaw in the code.

the outcome of a programme relies on the 2 or more independ threads accessing & modifying A share resource - i.e. variable or file in memory

under normal load the system by chance will run in a favourable order most of the time

Exceeding threshold will lead to increased chance of Execution orders
→ often through interruptions & resource switching w/o completion on the same share resource

Race conditions are exist in the code

critical load creates an unstable environment

Memory & I/O contention

CPU doesnt work in isolation

Puts demand on memory & I/O (input, output)

which intros a new opp for error

IF runs out of RAM will use disk

as "virutal memory" (swapping)

this state is known as thrashing

Puts enormous stress of SOS, making all operations slow

if a critical data structure is swapped out of the virtual memory the program may retrieve stale or incons data