concurrency summary

Multithreading is when a program makes use of multiple threads in execution

Benefits of multithreading

Multithreading speeds up execution by taking advantage of parallel hardware.

Multithreading increases responsiveness as threads can still run while others are waiting for I/O.

Multithreading allows for scalability since it breaks down execution into chunks that can be run on multiple cores or even machines.

Multithreading can be a convenient solution to a problem, like running different aspects of a program on different cores.

Concepts in Multithreading

Parallelisation

Parallelisation is running code on parallel hardware, actually running multiple things simultaneously.

Concurrency

Concurrency is having multiple things on the go at the same time, even if some of them are paused. For instance run program 1 until it needs I/O, then while its waiting for that run program 2. Only one program was ever running at a time, but their execution was interleaved.

Kernel Threads

Kernel threads are threads managed by the OS. Multiple kernel threads can be run on different cores at the same time.

Kernel threads incur some overhead in being managed by the OS

Green Threads

Green threads are threads which are managed by a programming language. Any interleaving of green threads is done by the programming language.

They are seen by the OS as one process and therefore cannot take advantage of parallel hardware.

Since they are managed within one process, the system doesn't concern itself with interleaving and this incurs less overhead.

Processes

Processes are more expensive to create than threads. The benefits of using them is they can be more convenient as their concurrency is managed by the OS, they limit information sharing - don't share variables, and don't effect other processes when they crash.

Spawning processes

Processes are spawned by a parent process. They are effectively a copy of the parent process at the start.

Lifecycle

- new
- ready
- waiting
- running
- terminated

Process Control Block (PCB)

The PCB stores information associated with each process. This information includes:

- state
- number
- program counter(s)
- registers
- memory limits
    - limits of the memory the program is allowed to access
- list of open files