Basics

Concurrent Computing

- several operations are performed during overlapping time periods
- a feature of a program that can do multiple things at the same time
- program is concurrent if it consists of units that can be executed independently
- **Purpose of Concurrency**
  - makes program run faster
    Running multiple activities at once can use the machine more effectively because there are multiple hardware components
  - make programs more responsive
    Structuring a program as concurrent activities can make it more responsive because while one activity blocks waiting for some event, another can do something

Concurrency with Process

- Processes run concurrently on the computer
- OS virtualizes memory processes don't share memory naturally
- This can make it difficult to program processes that have complicated cooperative behavior

Threads

- thread
  basic unit of CPU utilization within a process
- multi-threaded process
  concurrent execution of different parts of the same running program
- Thread comes with its own
  - Thread ID
  - Program Counter
  - Register Set
  - Stack
- shares with other threads
  - code/text section
  - data segment
  - list of open file descriptors
  - heap
  - signal behaviors
- Advantage of Threads vs. Process
  - Resource Sharing
    threads naturally share memory having concurrent activities in the same address
  - Economy
    Creating a thread is cheap
    Context-switching between threads is cheaper than between processes
    So if you can do with threads what you can do with processes, then you likely can do it a bit faster
- Drawback of Threads vs Process
  - one thread fails with an error/exception which is not managed
  - Threads may be more memory-constrained than processes
  - Threads do not benefit from memory protection

User Threads vs Kernel Threads

- User Threads
  - threads can be supported solely in User Space
  - main advantage: low overhead (e.g. no system calls)

  - Drawbacks
    If one thread blocks, all other threads block
    All threads run on the same core

- Kernel Threads
  - The kernel provides data structures and system calls to handle threads

- Thread Libraries
  provide users with ways to create threads in their own programs

Java Threads

- Thread Class
  Implementing a subclass that extends Thread

-

Java threads takeaway

- To launch or spawn a Thread/Runnable it is necessary to call the start() method (instead of run())