ETEC3702 – Concurrency
Midterm Study Guide

Topics

You are responsible for **everything** covered in class through Tuesday, 25 February 2020.

Some of the major (but not necessarily all) topics included will be:

- ❍ Introduction to Concurrency
    - ❒ Definition
    - ❒ Concurrent Versus Sequential
    - ❒ Formal notation
        - "For all"
        - "There exists"
        - "It is true that"
        - "Concurrent with" (||)
        - "Precedes" (→)
        - Formal definition of sequential
        - Formal definition of concurrent
    - ❒ Non-deterministic execution
- ❍ Dangers of Concurrency
    - ❒ Effective orderings – how to compute
    - ❒ Problematic orderings
    - ❒ Safety Property Violations
        - Concurrent update problem (turnstiles problem, bank account problem)
        - Critical Sections
        - Mutual Exclusion
        - Software Solutions
        - Dekker's Algorithm
        - Peterson's Algorithm
    - ❒ Liveness Properties Violations
        - Deadlock
        - Livelock
        - Starvation

    - ❒ Difficulty in Testing
- ❍ Motivation for Concurrent Solutions
    - ❒ Organization
        - Natural Solutions
        - Relax over-specification
        - Multiple separate tasks
    - ❒ Speed-up
        - Moore’s Law and the Historic Trend of CPU Clock Speeds
        - Multiple Core CPUs
        - Distributed Computing / “Cloud” Computing
        - Specialty Parallel Processors / GPU Computing
        - Computing Speedup to compare sequential versus concurrent solutions.
        - Computing Maximum speedup of a program with sequential and parallelizable parts.
    - ❒ Unavoidable / Inherent
        - Network / Communication Systems
        - Database Systems
        - Distributed Systems
        - Event-driven Systems
        - Physical Systems.
- ❍ Synchronization mechanisms
    - ❒ Locks
        - acquire( )
        - release ( )
        - with lock syntax.
        - Optional arguments (blocking, timeout)
        - The “Monitor” object design pattern concept.
    - ❒ Rlocks
        - motivation for / versus normal locks.
        - acquire( )
        - release( )
        - with rlock syntax
        - Optional arguments (blocking, timeout)
    - ❒ Semaphores

- As a data-type
- Set of values
- Set of permissible operations
  - Definition of P( ) or acquire( )
  - Definition of V( ) or release( )
- Types of Semaphores
  - Binary
  - Counting
- In Python
  - Semaphore objects
  - BoundedSemaphore objects

❐ Conditional Sections

- Producer / Consumer design patterns.
- The need for conditional synchronization
- Condition objects
  - acquire( )
  - release( )
  - wait( )
  - notify( )
  - notify_all( )
  - Use to implement a “Monitor” design pattern

❐ Events

- Concept and function
- As a synchronization mechanism
- As a communication mechanism
- Event objects
  - set( )
  - clear( )
  - is_set( )
  - wait( )

❐ Barriers

- Concept and function
- Uses
- Barrier objects
  - wait( )

        - abort( )
        - reset( )
        - broken
        - parties
        - n_waiting

❍ Communication Mechanisms

- ❒ Global variables
- ❒ Events
- ❒ Queues
    - Queue(), LifoQueue( ), PriorityQueue( )
    - Optional args: maxsize
    - put( )
    - get( )
    - empty( )
    - full( )
    - task_done( )
    - join( )

❍ Multiprocessing

- ❒ Differences from threading
- ❒ Advantages over threading
- ❒ Disadvantages related to threading
- ❒ Potential for speedup
- ❒ Need for the __name__==”__main__” check
- ❒ cpu_count( )
- ❒ Inter-process communication using queues.
- ❒ Using queues with processes to “return” values.
- ❒ Support for other synchronization and communication mechanisms
    - Lock
    - Rlock
    - Condition
    - Semaphore / BoundedSemaphore
    - BarriersEvent
    - Queue

Reminders/Advice:

- The entire exam will be open-notes.
- You cannot use the internet or any other electronics for the duration of the written portion of the exam. There may be a programming portion of the exam. You can use a computer for that portion, but you may not communicate with others during that portion. (no discord, no stack-exchange, no reddit, no facebook chat, no email, no communicating whatsoever.)
- You should complete and understand all of the assigned labs.
- You may have to write ad submit code. Come prepared to do that.
- Open notes, but don’t assume open-book == no preparation/study.