Australian National University

C01

Introduction to Concurrency

What is Concurrency?

Sources of Concurrency

What is Concurrency?

Negative definition based on absence of sequence:

If there is no external observer
who can identify two events
as being in strict temporal sequence
(that is, one event has fully terminated before the other one started)
then these two events are considered *concurrent*.

What is Concurrency?

In the context of programming and logic:

> *"Concurrent programming abstraction is the study of interleaved execution sequences of the atomic instructions of sequential processes."*

- Ben-Ari 2006

M. Ben-Ari *Principles of Concurrent and Distributed Programming*, Addison-Wesley, second edition, 2006

Why Do We Need Concurrency?

- Physics, engineering, electronics, biology, …
  … basically **every real world system is concurrent!**
- Sequential processing is suggested by most computer architectures
  … yet (almost) all current architectures have concurrent elements
  … and most computer systems are part of a concurrent network
- Widely used programming languages assume sequential processing
- Sequential programming delivers some fundamental components for concurrent programming
  … but additional concepts are required

Why Do Computer Scientists Care?

- Computer scientists must understand concurrency to be able to:
  - connect computer systems with the real world
  - employ / design concurrent parts of computer architectures
  - construct complex software packages (operating systems, compilers, databases, …)
  - understand when sequential and/or concurrent programming is required enhance the reactivity of a system
  - enhance the performance of a system
  - design embedded systems

Sources of Concurrency in Computer Systems

- Overlapped I/O and computation
  - Employ interrupt programming to handle I/O
- Multi-programming
  - Allow multiple independent programs to be executed on one CPU
- Multi-tasking
  - Allow multiple interacting processes on one CPU
- Multi-processor systems
  - Add physical/real concurrency
- Parallel machines & distributed operating systems
  - Add (non-deterministic) communication channels
- General network architectures
  - Allow any form of communicating, distributed entities

Types of Concurrent Computer Architectures

Flynn's [1966] terminology for physically concurrent machine architectures:

- SISD (single instruction, single data)
  - Sequential processors
- SIMD (single instruction, multiple data)
  - Vector processors
- MISD (multiple instruction, single data)
  - Pipelined processors
- MIMD (multiple instruction, multiple data)
  - Multi-processors or computer networks

Michael Flynn (1966). *Very high-speed computing systems*. Proceedings of the IEEE (Volume: 54, Issue: 12, Dec. 1966)

An Engineer's View of Concurrency

- Multiple physical, coupled, dynamical systems form the actual environment and/or task at hand
  - In order to model and control such a system, its inherent concurrency needs to be considered
  - Multiple less powerful processors are often preferred over a single high-performance CPU
  - The system design is usually strictly based on the structure of the given physical system

Does Concurrency Lead to Chaos?

- Concurrency often leads to the following features / issues / problems:
  - non-deterministic phenomena
  - non-observable system states
  - non-reproducible behaviour
    - results may depend on more than just input parameters and starting state (e.g. timing, throughput, load, available resources, signals … *throughout* the execution)
    - what are consequences for debugging?

Does Concurrency Lead to Chaos?

- Use non-determinism where underlying system is non-deterministic
- Use non-determinism where precise execution sequence is irrelevant
- Use synchronization only where necessary