UNIVERSITY *of* STIRLING

www.cs.stir.ac.uk

Concurrent and Distributed Systems

Introduction to Concurrent Systems

Introduction

- Some Logistics
- Course Contents
- Course Layout
- Resources for private study
- Introduction / Motivation to the subject

Course Contents

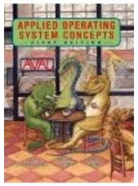
- **Lectures**
  - Previously :
    - programming Java (CSCU9P1/P2/A3/A5)
    - systems (CSCU9V4)
  - Now :
    - Processes and Threads and their management
    - Concurrency
    - Distributed systems
- **Laboratories**
  - Focus not so much Java, more Concurrency & Distribution
  - Practical experience on taught (lecture) material

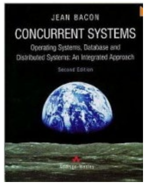
Course Layout

- Introduction to lecture series
- Process and Thread Management
  - Basic process model
  - Introduction of the problem
- Concurrency
  - Means of communication
  - Critical sections, Synchronisation
  - Scheduling
- Distributed Systems
  - Client – Server model
  - Java for distributed systems

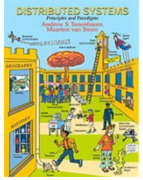
Resources for Private Study

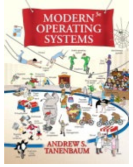
- Internet: lectures & lab sheets will be available in pdf format
- Books




JEAN BACON
CONCURRENT SYSTEMS
Operating Systems, Database and Distributed Systems: An Integrated Approach
Second Edition

  - Applied Operating Systems Concepts, A. Silberschatz, P. Galvin, G. Gagne, John Wiley & Sons
  - Concurrent Systems, J. Bacon, Addison-Wesley
  - Modern Operating Systems, A.S. Tanenbaum, Prentice-Hall,
  - Distributed Systems, A.S. Tanenbaum, M. v. Steen, Prentice-Hall





- Emails or Teams chat
  - Mario Kolberg
    mario.kolberg@stir.ac.uk

Concurrent & Distributed Systems

- Concurrent System
  - Several activities are happening at the same time

    Two activities are concurrent if, at any given time, each is between its starting and finishing point. (Bacon)
- Distributed Systems
  - Special case of concurrent systems

    A distributed system is a collection of independent computers that appear to the user of the system as a single computer. (Tanenbaum)
  - Two aspects:
    - Hardware: autonomous machines
    - Software: hardware appears as single machine to the user

Classification of Concurrent Systems

- Inherently Concurrent Systems
  - Real-time Systems
  - Operating Systems

- Potentially Concurrent Systems
  - Large amount of computing (graphics applications)
  - Large amount of data to be processed (simulations)
  - Real-time requirement for the result (voice processing)
  - Hardware is available to run applications in parallel

Example: Real-time Systems

- Timing constraints dictated by the environment of a system
- System has to respond to external events within a certain time
- Examples: - aircraft systems
  - hospital patient monitoring
- Software monitors & controls aspects of the environment of the system

Example: Operating Systems

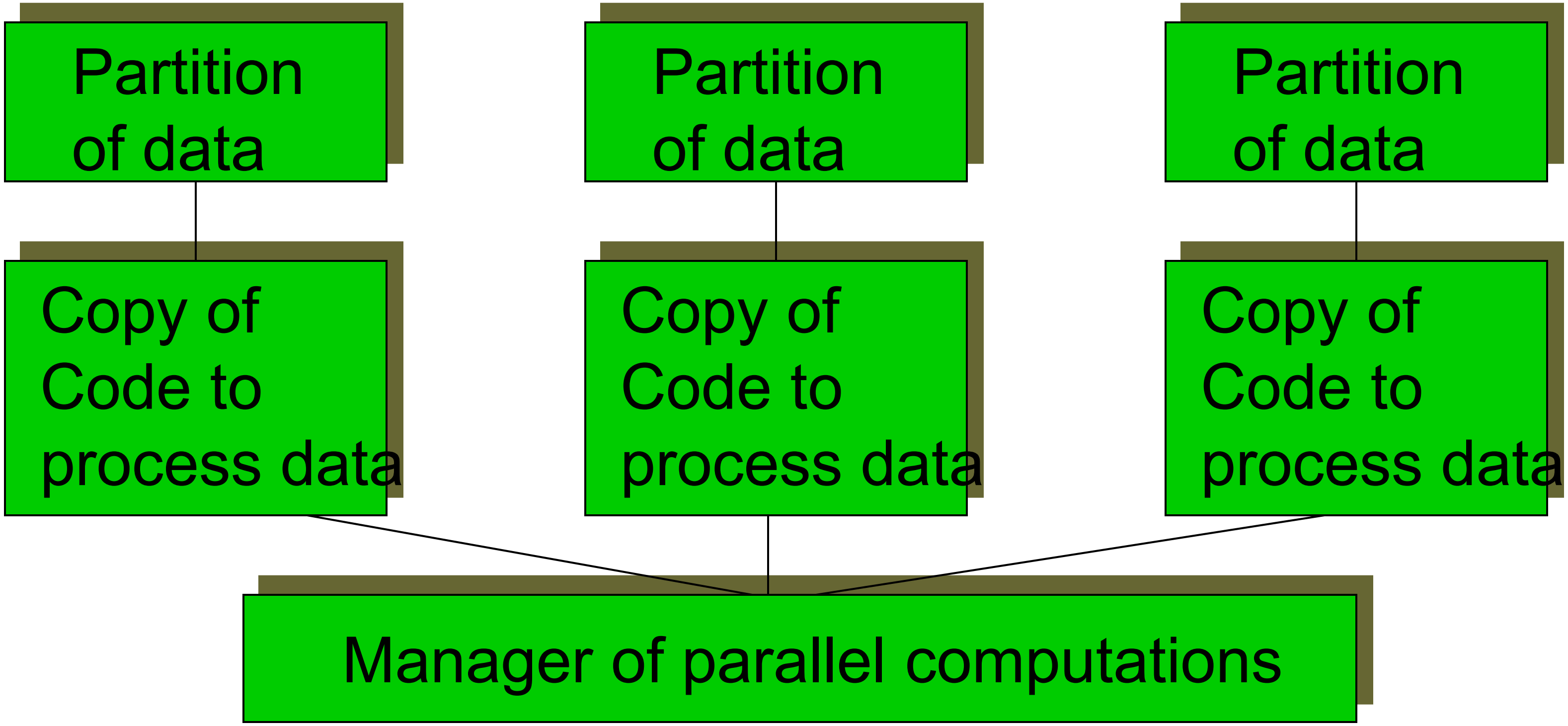
- Single user and multi-user systems
- Devices are slow compared to the main processor
  - OS attends to the devices when necessary
  - Other tasks are performed while devices are busy
- Users run a number of applications in parallel
- Multi-core CPU
- Running programs of different users in parallel
- Preemptive and non-preemptive scheduling
- Distributed Operating Systems

Examples of Potential Concurrency

- Examples in this category would benefit from concurrency
- Example cooking recipe
  - Can be ‘executed’ sequentially
  - Some steps can be carried out simultaneously
  - Ingredients for the next step need to be ready

- How is concurrency introduced
  - Partition data, replicate code
  - pipeline

Partition data, replicate code (Bacon)


Partition of data

Partition of data

Partition of data

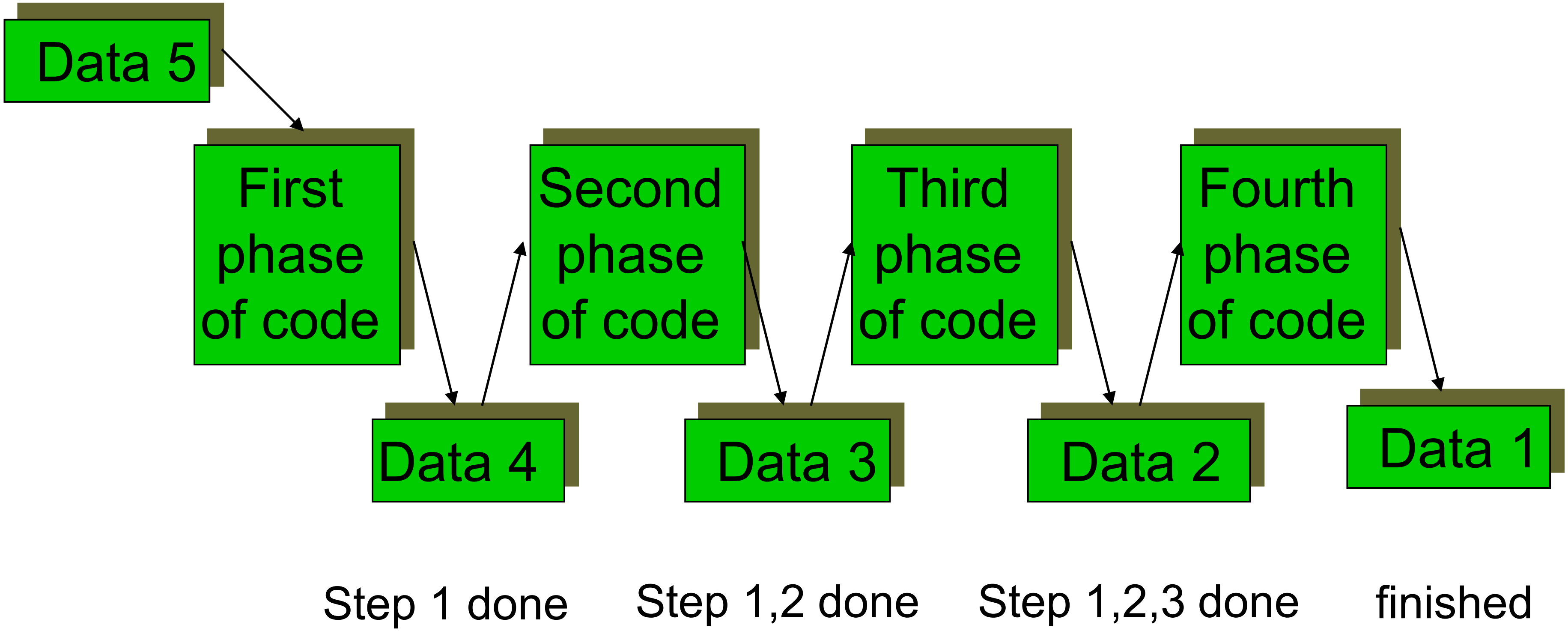
Copy of Code to process data

Copy of Code to process data

Copy of Code to process data

Manager of parallel computations

Pipeline (Bacon)



Benefits of Distributed Systems

Economics	Data Sharing
Speed	Device Sharing
Inherent Distribution	Communication
Reliability	Flexibility
Incremental Growth	Transparency

Challenges of Distributed Systems

Expensive Software	Communications Delay
Scalability/Overhead	Inconsistent State
Security	No Global Time
Independent Failure Nodes	Heterogeneity