Course Aims

This course aims to:

- provide you with a general understanding of how a computer works,
- explain the structure and functions of an operating system,
- explain the need for concurrency control and how to implement it,
- illustrate key operating system aspects by concrete example, and
- prepare you for future courses. . .

Course Objectives

At the end of the course you should be able to:

- describe the fetch-execute cycle of a computer,
- understand the different types of information which may be stored within a computer memory,
- compare and contrast CPU scheduling algorithms,
- explain the following: process, address space, kernel and user thread,
- distinguish paged and segmented virtual memory,
- outline how files are managed,
- explain with examples why concurrency control is needed,
- understand how concurrency control can be implemented,
- discuss the relative merits of Unix and NT. . .

Course Outline

- Part I: Computer Organisation
  - Computer Foundations
  - Operation of a Simple Computer.
  - Input/Output.
- Part II: Operating System Functions
  - Introduction to Operating Systems.
  - Processes & Scheduling.
  - Memory Management.
  - Filing Systems.
- Part III: Concurrency Control
  - Mutual exclusion and condition synchronisation.
  - Multi-threading: user and kernel threads.
  - Implementation and use of semaphores.
  - Inter-process Communication (IPC).
- Part IV: Case Studies
  - Unix and Windows NT.

Recommended Reading

- books for your hardware/architecture courses, e.g.
  Patterson D and Hennessy J
  Computer Organization & Design (2nd Ed)
  Morgan Kaufmann 1998.
- Silberschatz A, (Peterson J), Galvin P, (Gagne G.)
  Operating Systems Concepts
  (recent editions, 5th, 6th as available)
  Addison Wesley 1998, 2001.
- Bacon J M
  Concurrent Systems (2nd Ed)
  Addison Wesley 1997
  (Bacon and Harris due approx Dec/Jan.)
- OS books contain case studies on UNIX and NT.
  There are specialist books on both
  - not required reading.