

SCC.211 Part II

Introduction

Dr Andrew Scott  
a.scott@lancaster.ac.uk

---

---

---

---

---

---

---

Topics

- Wednesday
  - 6. Introduction
  - 7. Memory Allocation
  - 8. Multi-process Systems
  - 9. Memory Protection
  - 10. Virtual Memory
- Friday
  - 6. Processes
  - 7. Input and Output
  - 8. File-systems
  - 9. Paging
  - 10. Emerging Issues

---

---

---

---

---

---

---

Delivery: *Recorded Material*

- Majority of core material pre-recorded
- Slides and videos released around time of first session of week
  - Review material for next topic before next timetabled session
  - So you should have watched first videos by the next timetabled session  
...and videos for second topic by first session next week

---

---

---

---

---

---

---

Example Topic on Moodle

- Slides and Media
  - Video (with duration)
  - Slides
  - Reading
- Additional Reading

**1. Operating Systems**

We start by considering what we mean by an operating system, and the different ways an operating system can be structured.

**Slides and Media**

Topic	Video	Slides	Reading (Stilberschatt 10th ed.)
1. What is an OS?	Video 05:01	Slides	1.1 Initial Operating Systems (Ch. pages 4-7)
2. OS Design	Video 17:05	Slides	2.7 Operating System Design and implementation, 79-81 2.8 Operating System Structure, 81-91

**Additional Reading**

Textbook: 4th ed. 1.7 Operating System Structure, 82-88  
Video: Linux Tutorial in Arabic: Textbook Debate – Monolithic vs. Microkernel  
Paper: Library OS: version of Windows?  
 Paper: Linux/Windows

---

---

---

---

---

---

---

---

Delivery: *Timetabled Sessions*

- Run ½ week behind videos
- Some additional core content to get through
  - Otherwise, more responsive
  - Key slides available, only some/ parts recorded
    - Typically limited to key content
    - Look for sections in Moodle:

Timetabled Session: Thursday week 6

Topic	Video	Slides	Reading (Stilberschatt 10th ed.)

---

---

---

---

---

---

---

---

Delivery: *Timetabled Sessions*

- Typical structure
  - Brief summary of previous topic
  - Review key questions from Moodle forum
  - May expand on topic, add context, give examples, etc.
    - Possibly some *new* content
  - Brief overview of next topic

---

---

---

---

---

---

---

---

Reading

- You should/ need to read up on the topics covered
  - Give you a more in-depth understanding
    - There's a limit to how much we can cover in lectures
    - Important when explaining and justifying answers (in exams)
  - Helps identify links between topics
    - Actually makes things easier to follow/ understand
    - Makes subject more concrete and more interesting
- May raise questions you've not thought of
  - Have you misunderstood something?
  - Do you understand the implications of design choices?
- Some questions in papers may test you've done further reading
  - Strongly encouraged to have such questions by external examiners

---

---

---

---

---

---

---

Recommended Texts

- See Moodle page for details
  - Available from library
  - Recent editions okay



---

---

---

---

---

---

---

Main Support

- Taught material
  - Moodle forum
  - Review key forum questions in timetabled sessions
- Coursework
  - Labs 2 hrs per week – must attend your timetabled session
    - May be able to get support in other sessions if quiet – *but please check with TAs first*
  - Moodle forum

---

---

---

---

---

---

---

Lab Support

- Staff/ TAs there to help **you** design and debug
  - Not there to just give you answers
  - Remember: aim is to grow your confidence in these skills
- Please don't hog support, or try to ratchet a solution from multiple TAs

---

---

---

---

---

---

---

Summary

- Please try Moodle forum first
  - More eyes checking
  - Builds community
    - Feel free to reply... but please don't ask for, or post, solutions, key bits of code, etc.
  - Helps everyone

---

---

---

---

---

---

---

Coursework

Assessed exercise 2

---

---

---

---

---

---

---

Working on the Exercise

- (Constrained) real-world problem
  - Given requirements, design and implement a solution
- Multi-week exercise
  - Need to make a start
  - Expected to work on exercise outside of timetabled sessions
  - For you to make best use of help in labs
- If you've not kept up with your C, you'll need to review first-year material
  - Some additional pointers on Moodle
  - Not purpose of these labs, so support may be limited to SCC.211 specific questions

---

---

---

---

---

---

---

The Task

- Implementation of FAT16 filesystem
- Tasks:
  1. Reading specified parts of some file – *suggest a text file, known content*
  2. Reading filesystem metadata, from given 'disk' image
    - Boot Sector and BIOS Parameter Block
  3. Scanning the File Allocation Table
  4. Reading root directory
  5. Reading a file
  6. VFAT long filename support (i.e., not limited to old 8.3 format)
  7. Handling the directory tree



---

---

---

---

---

---

---

FAT16 Filesystem Structure

- Metadata
  - Boot Sector and BIOS Parameter Block
- File Allocation Table (duplicated for resilience, typically 2)
- Root Directory
- Data area
  - Files and directories

Boot Sector				
BIOS Parameter Block	Reserved Sectors	FAT[n]	Root Directory	Data

---

---

---

---

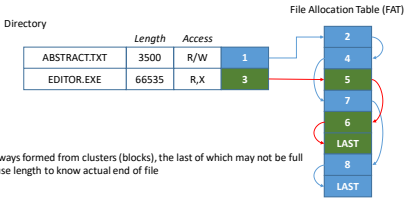
---

---

---

Finding and Accessing Files

- Directory and File Allocation Table
  - Given any cluster number, indicates next cluster in file



---

---

---

---

---

---

---

---

Introduction to Topic 1

Introduction to Operating Systems

---

---

---

---

---

---

---

---

Operating System

- Machine does little without one, OS...
  - Manages hardware and system resources
  - Shares out and accounts for resources
  - Offers secure environment for applications
  - Provides common device or I/O system
- Includes kernel and system library interface
  - Don't consider \*
    - Applications and services
    - Desktop environment

\* In practice distinction can be hard to maintain

---

---

---

---

---

---

---

---

Operating Systems

- Embedded/ Real-Time, emphasis on
    - Proven long term reliability and strict timing guarantees
    - Small footprint, no unnecessary code
    - Certification process for safety critical systems
  - Server, emphasis on
    - Fairly sharing resources
    - Reliability
  - Desktop, emphasis on
    - Interactivity
    - GUI and graphics/ media
- Could be same OS code running with different parameters

---

---

---

---

---

---

---

Operating Systems

- There are many, but...
- Unix and Windows
  - Dominate market and have huge application and support base
    - Achieved 'critical mass'
  - User view driven by graphical interface
    - Arguably not part of OS
    - Can be quite different to underlying OS
  - Not necessarily where we'd start from today

---

---

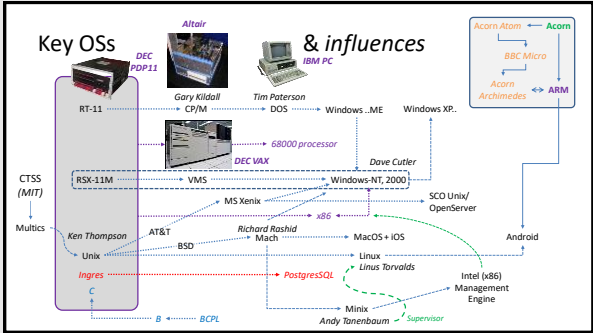
---

---

---

---

---



---

---

---

---

---

---

---