Comp 204:
Computer Systems and Their Implementation

Dr. Terry Payne.
Department of Computer Science,
University of Liverpool.

Comp 204: Computer Systems and Their Implementation

Lecture 1: Introduction

Today

- Admin and module info
- Introduction to Operating Systems
  - Overview
  - OS managers

Admin

Lecturer:

Dr. Terry Payne
Room 205, Ashton Building
Email: T.R.Payne@liverpool.ac.uk

Course Notes:

http://www.csc.liv.ac.uk/people/trp/COMP204.html

Announcements (via RSS):

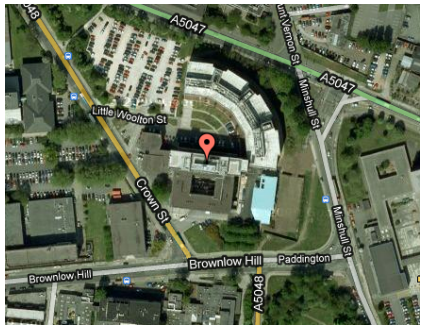
http://www.csc.liv.ac.uk/people/trp/Teaching/rss.xml

Acknowledgement:

thanks to both Katie Atkinson and Dave Jackson for supplying material related to the content of this module.

Module Delivery

- Lecture times and locations:
  - University Lecture Rooms Building
  - Tuesday 14:00
    - LIFS-LT2
  - Thursday 15:00
    - REN-LT6
  - Friday 12:00
    - NICH-LT
- Lab classes:
  - These will support the two assignments
  - People will be assigned to a lab slot when the assignments are announced
    - Check announcements during lectures and on the web site



Module Aims and Objectives

- To create an understanding of how the principal software components of modern computer systems perform their functions, and how they are constructed and interact with each other.

- At the end of the module, students should be able to construct programs which demonstrate in a simple form the operation of examples of systems programs, including simple compilers and programs that involve management of concurrent processes.

Syllabus Outline

- Operating Systems Concepts;
- Processes;
- Concurrent Programming;
- Memory Management;
- Input/Output and Files;
- Compilers;
- Revision.

Module Syllabus (Approximate)

- Operating Systems concepts:
  - communicating sequential processes;
  - process management and scheduling;
  - resource allocation, mutual exclusion, semaphores, deadlock.

- Concurrent programming in Java:
  - Java threads;
  - The Producer-Consumer problem.

- Memory Management:
  - storage management systems and their problems;
  - segmentation;
  - paging;
  - page replacement policies.

- Input/Output and Files:
  - filestore organisation;
  - file allocation policies;
  - buffering and caching;
  - device handling.

Module Syllabus (Approximate)

- Compilers:
  - a practical overview of compiler construction;
  - lexical analysis;
  - parsing;
  - code generation;
  - interpretation examined in the context of Java and available software tools.
- Run-time store organisation:
  - dynamic store allocation;
  - treatment of recursion;
  - organisation of the Java virtual machine.
- Revision.

Recommended Texts

- *Operating System Concepts (8th Edition).* Silberschatz, Galvin & Gagne (Wiley)
- *Understanding Operating Systems.* Flynn and McHoes (Thomson)
- *Compiler Construction: Principles and Practice.* Louden (Thomson)
- *Programming Language Processors in Java.* Watt & Brown (Prentice Hall)

Lecture notes include material based on examples from all of the above texts

Module Assessment

- There is a coursework component that counts for 20% of the final mark for Comp 204.
- The CA component consists of 2 practical (Java-based) exercises that each contribute 10% to the CA component. Details to follow as the module proceeds.
- There is also a 2 hour exam in May which is worth 80% of the final mark.

Notes

- **Course webpage**:

  http://www.csc.liv.ac.uk/people/trp/COMP204.html

- **Printouts** of the lecture notes will be available on a weekly basis from the computer science helpdesk (George Holt Building) as the module proceeds.

- **Office hours**: I will be available for people to come and see me during the following times, but **please email me first** to make an appointment:
  - Tuesdays 3pm – 4pm
  - Thursdays1pm – 3pm

Please…

- Switch off all mobile phones during lectures.
- Do not scan the register on behalf of other people.
- Attend lectures, but do not talk during them, and attempt the exercises set.
- Attend the practical classes and complete the coursework.
- Ask questions if there is anything that you do not understand.

Exercises

- The module will be interactive.
- The aim is to promote learning by making you *think* about the material presented.
- At frequent intervals I will ask questions about current and previous lectures' material and set small exercises for you to complete during the lectures. Requires collaboration and audience participation!

Operating Systems Concepts

Operating Systems

- Purpose
  - To turn base hardware into a usable machine
  - To make efficient use of available resources, particularly when they are shared

Examples

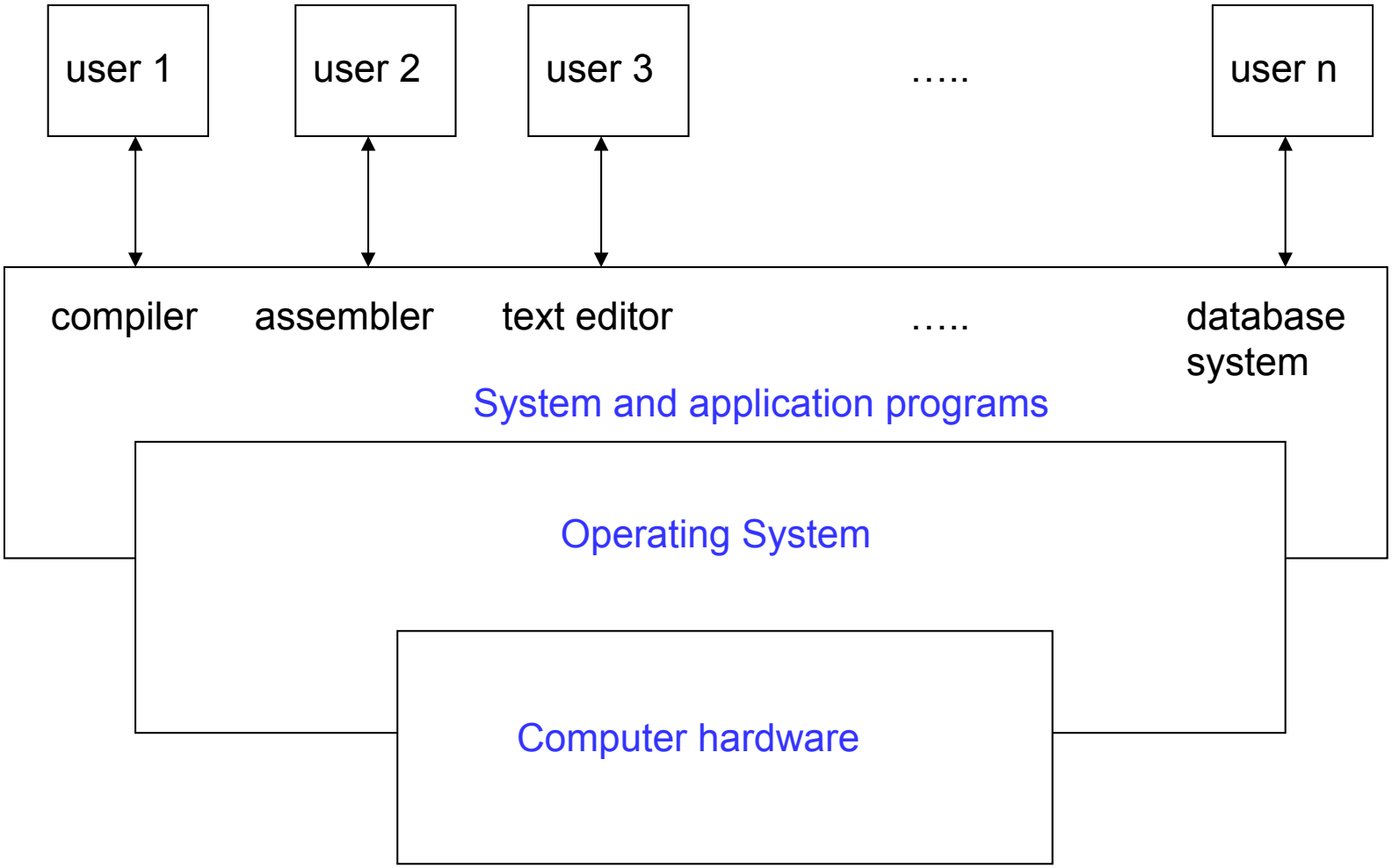
- Some operating systems are specific to certain types of computer, while others can run on a range of different designs:

- Windows (developed by Microsoft)
  - designed for Intel processors

- MAC-OS (developed by Macintosh)
  - designed for use only on Macintosh computers
    - In the past, running on Motorola's PowerPC chipset
    - Nowadays, mainly Intel

- UNIX and later LINUX (developed by AT&T)
  - designed for a range of computers, including PCs and MACs

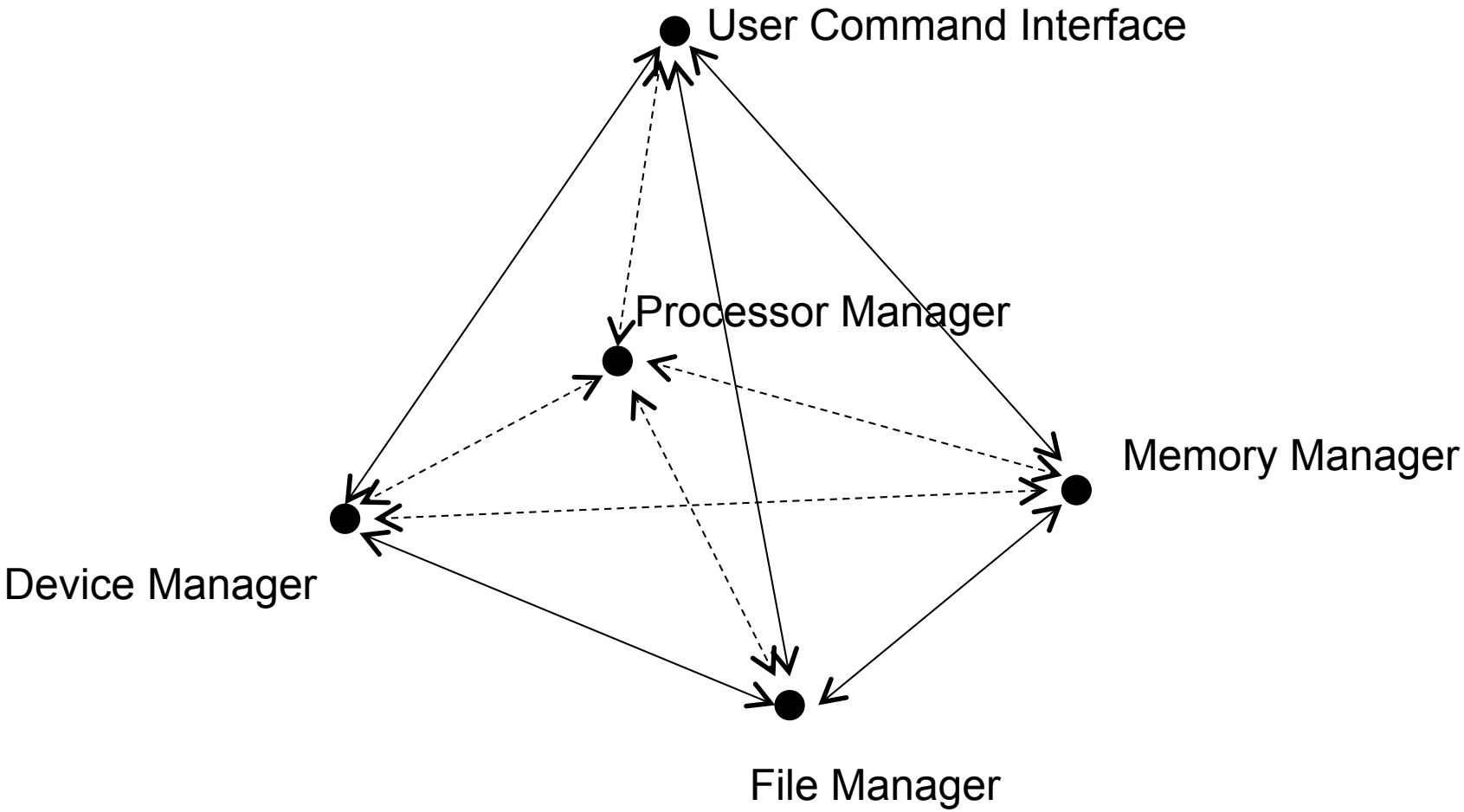
A Computer System



Machine Hardware

- OS role is to interact with the essential aspects of the computer system's hardware, the physical machine and its electronic components, which include:
  - Main memory: where data and instructions must reside in order to be processed
  - Input/Output (IO) devices: the peripheral units in the system, e.g., printers, keyboards, CD drives, modems etc.
  - The Central Processing Unit (CPU): contains the circuitry (the chips) that controls interpretation and execution of instructions

Operating System – An Abstract View



Operating System – An Abstract View

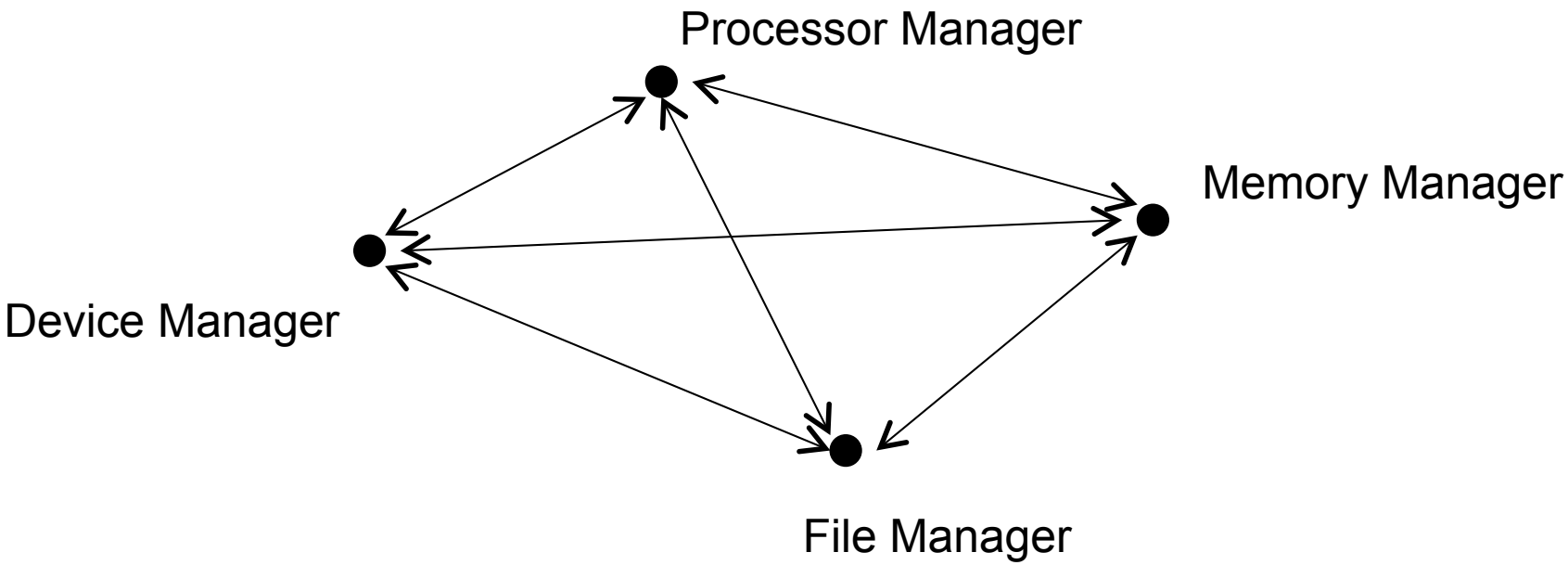
- The base of the pyramid shows the four essential managers of every OS, each working with the others to perform its task:
  - Memory Manager
  - Processor Manager
  - Device Manager
  - File Manager

- Network functions were not always an integral part of an OS
  - A Network Manager can be added to handle networking tasks

- User Command Interface: how users interact with the OS by issuing commands.
  - Varies from one OS to another

Operating Subsystem Managers

The base of the pyramid shows the four essential managers of every operating system:



Operating System Managers

- Each subsystem manager must perform the following tasks:
  - Continuous monitoring of resources
  - Enforcement of policies that determine who gets what resources, when they get them and how much
  - Allocation of resource when it is appropriate
  - De-allocation of resources when it is appropriate

Memory Manager

- Memory Manager: in charge of main memory
- Tasks:
  - Preserves and protects the space in main memory that is occupied by the OS itself
  - Checks validity of each request for memory space
  - For legal requests, allocates a portion of memory not already in use
  - In a multi-user system, must keep track of which users are using which section of memory
  - De-allocates sections of memory that are no longer needed

Processor Manager

- Processor Manager: decides how to allocate the central processing unit (CPU)

- Tasks:
  - Handles jobs as they enter the system
  - Manages each process within the jobs
  - Monitors whether CPU is executing a process or waiting for a ‘read’ or ‘write’ command to finish executing
  - Once the CPU has been allocated, sets up required registers and tables
  - Keeps track of the status of each process
  - Reclaims the CPU once the job is finished

Device Manager

- Device Manager: monitors every device and control unit

- Tasks:
  - Allocates the system's devices (e.g., printers, terminals, disk drives, etc.), in accordance with the system's scheduling policy
  - Must perform this allocation so as to allocate the devices in the most efficient manner possible
  - Once a device has been allocated the manager starts the device's operation and when required, de-allocates the device

File Manager

- File Manager: keeps track of every file in the system

- Tasks:
  - Monitors all files, including data files, compilers, application programs etc.
  - Enforces restrictions on who has access to which files (using a pre-determined access policy)
  - Controls what users are allowed to do with the files they can access
  - Allocates the resource by opening the file and de-allocates it by closing the file

Interaction Between OS Managers

- Each OS manager has specific, individual tasks to perform
- But, it is not enough for each to operate on its own: each manager must be able to work in harmony with the others
- Example:
  - Suppose a user types in a command at the keyboard to execute a program
- The following (simplified) steps must occur in sequence:

Interaction Between OS Managers

- **Device manager**: receives electronic signals from keyboard, decodes keystrokes, sends command to User Command Interface where Processor Manager validates command

- **Processor Manager**: sends acknowledgement message to monitor, determines whether program is already in memory or must be fetched from storage and notifies the appropriate manager

- **File Manager**: calculates program's exact location on disk, if not already in memory, and passes this info to the Device Manager

Interaction Between OS Managers cont'd

- **Device Manager**: retrieves the program and sends it on to the Memory Manager which must find space for it and records its exact location in memory

- **Memory Manager**: tracks program's location and progress as it is executed by the Processor Manager

- **Processor Manager**: receives a 'finished' message when the program has finished executing and forwards this message to the Device Manager which displays the message on the monitor

Network Manager

- For operating systems that have networking capability there is a fifth manager, the Network Manager, added to the model
- The Network Manager provides the facilities for users to share resources while controlling user access to them
- These resources include
  - Hardware, such as: CPUs, memory areas, printers, disk drives, etc
  - Software, such as: data files, application programs, compilers etc
- Adding this additional manager to our model, our system now looks like this…..

Operating System with Network Manager