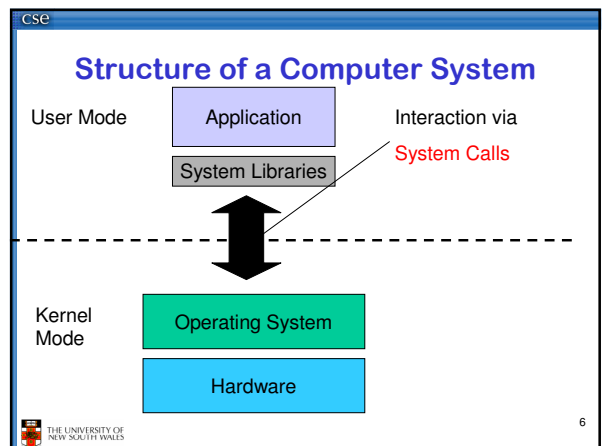cse

Operating System Overview
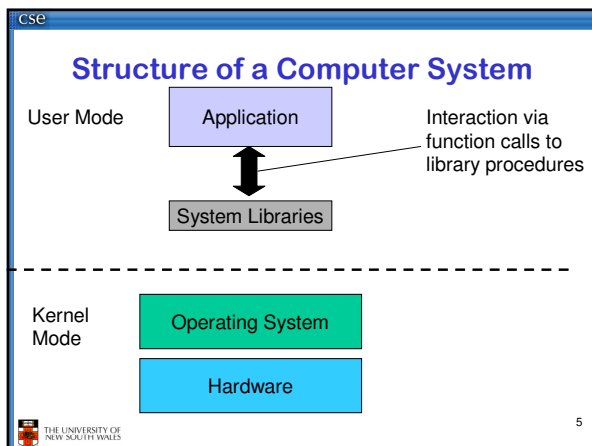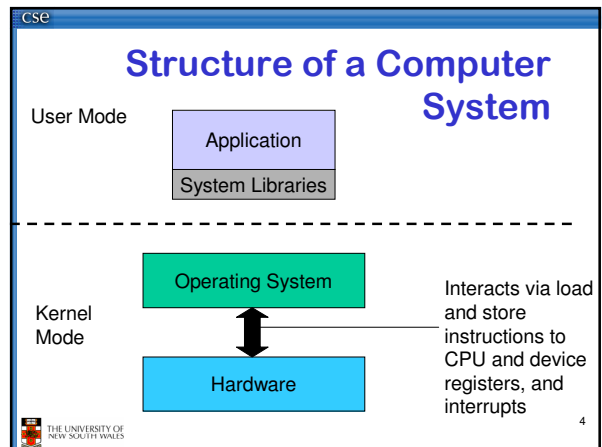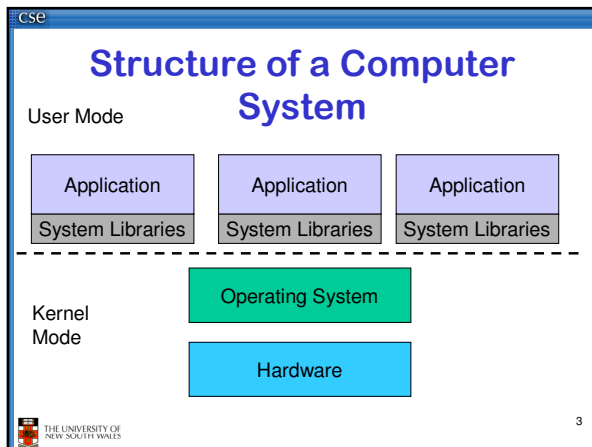
Chapter 1.5 – 1.9

1

---

cse

Operating System

- A program that controls execution of applications
  - The resource manager
- An interface between applications and hardware
  - The extended machine

2

---

cse

Structure of a Computer System

User Mode

Application	Application	Application
System Libraries	System Libraries	System Libraries

Kernel Mode

Operating System

Hardware

3

---

cse

Structure of a Computer System

User Mode

Application

System Libraries

Kernel Mode

Operating System

↕

Hardware

Interacts via load and store instructions to CPU and device registers, and interrupts

4

---

cse

Structure of a Computer System

User Mode

Application

↕

System Libraries

Interaction via function calls to library procedures

Kernel Mode

Operating System

Hardware

5

---

cse

Structure of a Computer System

User Mode

Application

System Libraries

Interaction via

System Calls

↕

Kernel Mode

Operating System

Hardware

6

A note on System Libraries

- System libraries are just that, libraries of support functions (procedures, subroutines)
  - Only a subset of library functions are actually systems calls
    - strcmp(), memcpy(), are pure library functions
    - open(), close(), read(), write() are system calls
  - System call functions are in the library for convenience

THE UNIVERSITY OF
NEW SOUTH WALES

7

Operating System

- Convenience **Objectives**
  - Make the computer more convenient to use
- Abstraction
  - Hardware-independent programming model
- Efficiency
  - Allows the computer system to be used in an efficient manner
- Ability to evolve
  - Permit effective development, testing, and introduction of new system functions without interfering with existing services
- Protection

THE UNIVERSITY OF
NEW SOUTH WALES

8

Services Provided by the Operating System

- Program development
  - Editors, compilers, debuggers
    - Not so much these days
- Program execution
  - Load a program and its data
- Access to I/O devices
- Controlled access to files
  - Access protection
- System access
  - User authentication

THE UNIVERSITY OF
NEW SOUTH WALES

9

Services Provided by the Operating System

- Error detection and response
  - internal and external hardware errors
    - memory error
    - device failure
  - software errors
    - arithmetic overflow
    - access forbidden memory locations
  - operating system cannot grant request of application

THE UNIVERSITY OF
NEW SOUTH WALES

10

Services Provided by the Operating System

- Accounting
  - collect statistics
  - monitor performance
  - used to anticipate future enhancements
  - used for billing users

THE UNIVERSITY OF
NEW SOUTH WALES

11

Operating System Software

- Fundamentally, OS functions the same way as ordinary computer software
  - It is a program that is executed (just like apps)
  - It has more privileges
- Operating system relinquishes control of the processor to execute other programs
  - Reestablishes control after
    - System calls
    - Interrupts (especially timer interrupts)

THE UNIVERSITY OF
NEW SOUTH WALES

12

Kernel

- Portion of the operating system that is running in *privileged mode*
- Usually resident in main memory
- Contains fundamental functionality
  - Whatever is required to implement other services
  - Whatever is required to provide security
- Contains most-frequently used functions
- Also called the nucleus or supervisor

13

Major OS Concepts

- Processes
- Concurrency and deadlocks
- Memory management
- Files
- Information Security and Protection
- Scheduling and resource management

14

Processes

- A program in execution
- An instance of a program running on a computer
- The entity that can be assigned to and executed on a processor
- A unit of resource ownership
- A unit of activity characterized by a single sequential thread of execution, a current state, and an associated set of system resources
  - Nowadays the execution abstraction is separated out: *Thread*
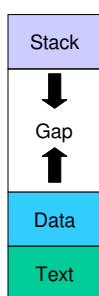 - Single process can contain many threads

15

Process

Memory

- Consist of three segments
  - Text
    - contains the code (instructions)
  - Data
    - Global variables
  - Stack
    - Activation records of procedure
    - Local variables
- Note:
  - data can dynamically grow up
  - The stack can dynamically grow down
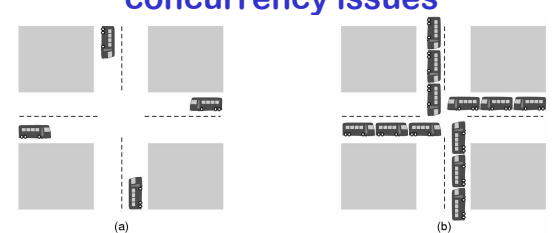
Stack

Gap

Data

Text

16

Process

- Consists of three components
  - An executable program
    - text
  - Associated data needed by the program
    - Data and stack
  - Execution context of the program
    - All information the operating system needs to manage the process
      - Registers, program counter, stack pointer, etc…
    - A multithread program has a stack and execution context for each thread

17

Multiple processes creates concurrency issues

(a) (b)

(a) A potential deadlock. (b) an actual deadlock.

18

3

Memory Management

- The view from thirty thousand feet
  - Process isolation
    - Prevent processes from accessing each others data
  - Automatic allocation and management
    - Don't want users to deal with physical memory directly
  - Support for modular programming
  - Protection and access control
    - Still want controlled sharing
  - Long-term storage
  - OS services
    - Virtual memory
    - File system

THE UNIVERSITY OF NEW SOUTH WALES

19

Virtual Memory

- Allows programmers to address memory from a logical point of view
  - Gives apps the illusion of having RAM to themselves
  - Logical addresses are independent of other processes
  - Provides isolation of processes from each other
- Can overlap execution of one process while swapping in/out others.

THE UNIVERSITY OF NEW SOUTH WALES

20

Virtual Memory Addressing

Figure 2.10 Virtual Memory Addressing

THE UNIVERSITY OF NEW SOUTH WALES

21

Paging

- Allows process to be comprised of a number of fixed-size blocks, called pages
- Virtual address is a page number and an offset within the page
- Each page may be located any where in main memory
- A page may actually exist only on disk

THE UNIVERSITY OF NEW SOUTH WALES

22

Main Memory

Disk

Main memory consists of a number of fixed-length frames, equal to the size of a page. For a program to execute, some or all of its pages must be in main memory.

Secondary memory (disk) can hold many fixed-length pages. A user program consists of some number of pages. Pages for all programs plus the operating system are on disk, as are files.
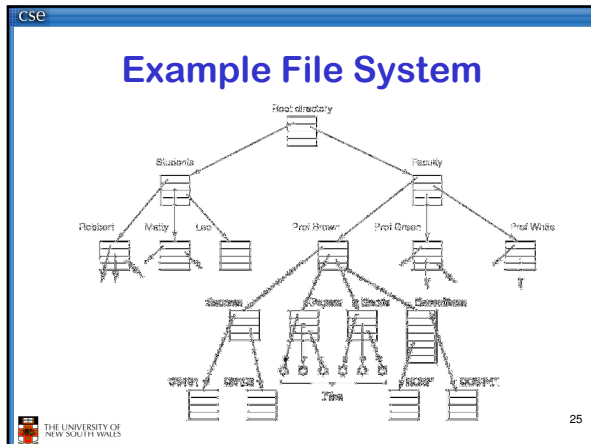
Figure 2.9 Virtual Memory Concepts

File System
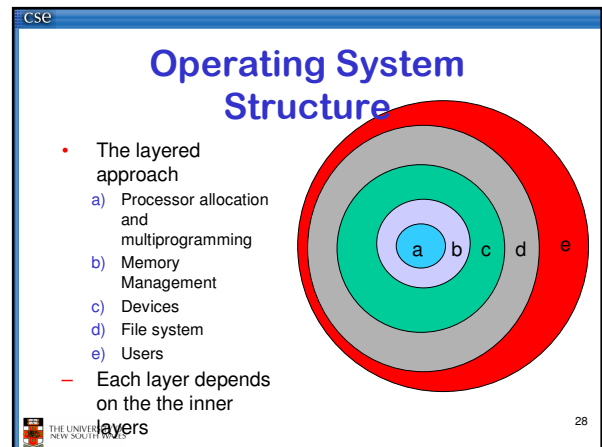
- Implements long-term store
- Information stored in named objects called files

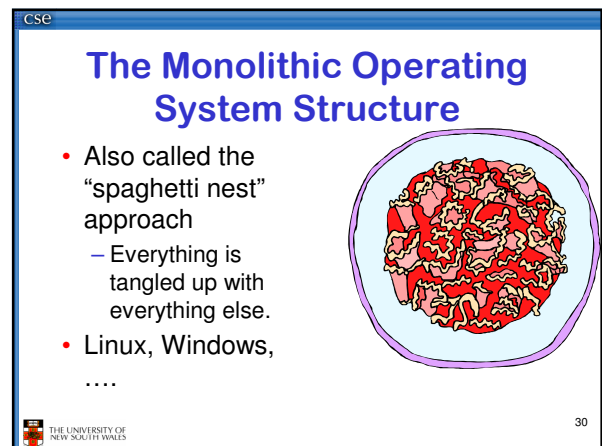THE UNIVERSITY OF NEW SOUTH WALES

24

4

Example File System

25

Information Protection and Security

- Access control
  - regulate user access to the system
  - Involves authentication
- Information flow control
  - regulate flow of data within the system and its delivery to users

26

Scheduling and Resource Management

- Fairness
  - give equal and fair access to all processes
- Differential responsiveness
  - discriminate between different classes of jobs
- Efficiency
  - maximize throughput, minimize response time, and accommodate as many uses as possible

27

Operating System Structure

- The layered approach
  a) Processor allocation and multiprogramming
  b) Memory Management
  c) Devices
  d) File system
  e) Users
- Each layer depends on the the inner layers

28

Operating System Structure

- In practice, layering is only a guide
  - Operating Systems have many interdependencies
    - Scheduling on virtual memory
    - Virtual memory on I/O to disk
    - VM on files (page to file)
    - Files on VM (memory mapped files)
    - And many more…

29

The Monolithic Operating System Structure

- Also called the "spaghetti nest" approach
  - Everything is tangled up with everything else.
- Linux, Windows, ….

30

5

The Monolithic Operating System Structure

- However, some reasonable structure usually prevails

Main procedure

Service procedures

Utility procedures

31

UNIX

- Provides a good hardware abstraction
  - Everything is a file (mostly)
- Runs on most hardware
- Comes with a number of user services and interfaces
  - shell
  - C compiler

32

Traditional UNIX Structure

UNIX Commands and Libraries

System Call Interface

Kernel

Hardware

User-written Applications

Figure 2.15 General UNIX Architecture

33

Traditional UNIX Kernel

User Programs

Trap

Libraries

User Level

Kernel Level

System Call Interface

File Subsystem

Inter-process communication

Process Control Subsystem

Scheduler

Buffer Cache

Memory management

character   block

Device Drivers

Hardware Control

Kernel Level

Hardware Level

Hardware

Microkernel-based Systems

- Assigns only a few essential functions to the kernel
  - Address space
  - Interprocess Communication (IPC)
  - Basic scheduling
  - Minimal hardware abstraction
- Other services implemented by user-level servers
- Traditional "system calls" become IPC requests to servers
- Extreme view of a microkernel
  - A feature is only allowed in the kernel if required for security

35

---

documents
windows
symbols
blocks & heaps
arrays & structures
variables

Application

threads
coroutines
modules
procedures
statements

Monolithic Kernel

File   Address Space   Semaphore   Priority   IPC   Pipe

Socket   Process   Monitor   Event   Segment   ACL   Page

Mutex   Thread   Task   Schedule

Bit Byte Word Register
Instructions

HW

36

6