Operating Systems

Overview

Dr Alun Moon

Computing, Engineering and Information Sciences

27th September 2011

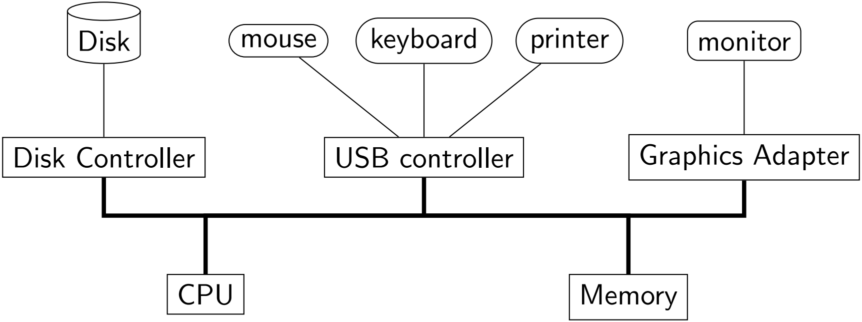
Chapters 1 & 2

Galvin Silberschatz.
Operating System Concepts.
Wiley, 6 edition, 2003.

What does an OS do?

User
- maximise use of resources
- share resources fairly

System
- allocate and manage resources
- control system



What does it do?

- The OS is intermediate between a user of a computer and the computer hardware
- It manages *efficiently* the CPU, memory, secondary storage, communucations (network, etc), peripherals
- It manages the Execution, running, synchronisation of application *processes/jobs* (running programs)
- It provides user interfaces to the computer system convenient to use - one or more *shells*

Why have OSs?

In early computing systems (and more recent embedded computing systems) a program running on the CPU read data from memory or from input peripherals, computed, wrote data to memory or output peripherals.

- Not scaleable! We want our system to manage several tasks concurrently
- Not efficient! Reading from the disk or network is 1000 times slower than performing a computation: we want the CPU to be using time efficiently.
- Our system may have several cooperating CPUs, some with dedicated functions like memory controllers, disk controllers, video controller, network hardware.
- We need a "lower layer" of software to run these tasks and to manage/coordinate them, and pprovide "services" to an upper layer of application software.
- This lower layer comprises the *device drivers* and the *operating system kernel*

Users		
system applications shells and commands compilers and interpreters system libraries		
System call interface		
signals handling character I/O System terminal drivers	file system block I/O system disk drivers	CPU Scheduling paging virtual memory
kernel interface		
Terminal controllers terminals	disk controllers disks	memory controllers physical memory

Good Operating System Features

A good OS -

- maximises machine utilization
- maximises system throughput
- minimises response time
- reduce the need for operator action, intervention
- allows the user to specify requirements easily (and meet those requirements as far as possible)
- allocates resources (file handles, video, ...) to jobs
- record usage of resources

Some of these requirements be conflict!

Good Operating System Features - Exercise

List some essential characteristics of an OS for:

- a washing machine controller
- an automatic pilot for an aeroplane
- a general purpose office PC
- a computer running an accountancy package
- a web server

Consider the following desirable OS characteristics -

- Fair allocation of system resources to tasks
- Maximising system throughput
- Reliability
- Meet user deadlines
- Ease of use
- Efficient use of storage

For each of the systems, *prioritise* these characteristics.

Process Management

- Process = running program - a unit of work within the system.
- A process needs resources to accomplish its task, eg CPU, memory, I/O
- On process termination, OS reclaims reusable resources
- A *single-threaded* process has one *program counter* specifying location of next instruction to execute. Process executes instructions sequentially, one at a time, until completion
- Multi-threaded process has one program counter per thread
- A typical system has many processes, some user, some operating system running concurrently on one or more CPUs
- *Concurrency* achieved by multiplexing the CPUs among the processe /threads

Memory Management

- All data in memory before and after processing
- All instructions in memory in order to execute
- Memory management determines what is in memory when, to optimise CPU utilisation and responsiveness to users
- Memory management activities include -
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes, data to move into and out of memory
  - Allocating and deallocating memory space as needed

Storage Management

Main memory only large storage medium that the CPU can access directly

Secondary storage provides large nonvolatile storage

- **Magnetic disks** Disk surface is logically divided into *tracks*, subdivided into *sectors*
  The disk controller determines the logical interaction between the device and the computer
- **Solid-state drives** flash memory devices that behave like disk drives

Storage Management

Caching copying information into faster storage system; main memory can be viewed as a "last cache" for secondary storage

- Important principle, performed at many levels in a computer
- Data in use copied from slower to faster storage temporarily
- Faster storage (cache) checked first to determine if information is there; if it is, data used directly from the cache (fast); if not, data copied to cache and used there
- Cache smaller than storage being cached; so cache management important design problem. How big? What data refresh policy?

Storage Management

OS provides uniform, *logical* view of information storage - files, devices

- File-System management
  - Files usually organised into directories
  - *Access control* determines who has what level of access to what
  - OS activities include
    - Creating, deleting, manipulating files and directories
    - Mapping files onto secondary storage
    - Backup of files onto stable (non-volatile) storage media
- OS activities
  - Free-space management
  - Storage allocation
  - Disk scheduling

Input, Output

- OS hides peculiarities of hardware devices from the user
- I/O subsystem responsible for
  - Memory management of I/O:
    - buffering: storing data temporarily while it is being transferred
    - caching: storing parts of data in faster storage for performance
    - spooling: the overlapping of output of one job with input of other jobs
  - General device-driver interface
  - Drivers for specific hardware devices

Protection, Security

Protection
: controlling access of processes or users to resources of the OS

Security
: defence of the system against internal and external attacks
  - denial-of-service, worms, viruses, identity theft, theft of service

Systems generally first distinguish among users, to determine who can do what

- User IDs (with password): one per user
- User ID associated with all files, processes of that user to determine access control
- Group identifier (group ID) allows a set of users to be defined and controls managed
- *Privilege escalation* allows user to change to effective ID with more rights

Computing Environments

- stand-alone v networked
- client-server computing
    - client, server computers communicating over a network
    - a client sends a request to a server
    - server responds to request, sends response to client
    - examples: file server, application server, DB server
- peer-to-peer computing
    - node broadcasts request for service, listens globally for requests it can fulfil
- web-based computing