OS Types

- Mainframe
- Server
- Multiprocessor
- Embedded
- Handheld
- Distributed
- Network
- Real-time

OS: Mainframe

- Mainframes: mainly used by large organisations for critical applications (e.g. bulk data processing such census, industry and consumer statistics)
  - Originally: enormous computers
  - Now: high-end commercial machines
- Features:
  - Redundant internal engineering
  - high reliability and security
  - extensive I/O facilities
  - strict backward compatibility
  - high utilisation rates

OS: Mainframe

- Characteristics:
  - Run (or host) multiple OS (~ virtual machines)
    - A single mainframe can replace many servers
  - Handle very high volume I/O and emphasise throughput computing
  - Massive database and files:
    - Massive data storage online with fast access
- Examples:
  - IBM OS/360 keeps track of all the system resources
- Uses:
  - E-business
  - Batch processing
  - Difficult computing problems: e.g. on-line transactions in China

OS: Server

- Designed from the ground up to provide platforms for multi-user, frequently business-critical, networked applications
- Servers are computers or groups of computers used for
  - internet serving: a hypertext transfer protocol (HTTP) server is included with many server OSs
  - intranet serving: internal company communication
  - print serving: allows multiple computers to use a single printer
  - file serving: providing a common storage point for a company's documents
  - application serving: e.g. databases, shared environments for collaborative apps
  - caching: speeding up network access by storing previously downloaded files
  - terminal services: allow a client to run a productive app on a server, while seeing the visual results on their screen

OS: Server

- Focus:
  - Security
  - Stability and collaboration
  - Not user interface
- Checklist
  - Administration – what tools are available
  - Security – very important feature
  - Stability – want to avoid downtime
  - Features – what specific services are provided?
  - Performance – does it match requirements?
  - H/W requirements – is OS tied to h/w?
  - Scalability – how many clients can it handle? How wll it scale?
  - Costs – productivity, admin, downtime
  - Third-party applications – what products are available?

OS: Server

- Issues to consider:
  - business size
  - cost
  - interface
  - support
  - licensing
    - No issues: Linux, FreeBSD, other open source OSs
    - User licences: 1 licence/client access (e.g. Windows)
    - Per-CPU server licensing model: e.g. Sun
  - maintenance
- Directory services
  - Keep track of an organisation's resources and user access permissions

OS: Multiprocessor

- Multiprocessor h/w
  - A computer system in which two or more CPUs share full access to the main memory
- Multiprocessing OS
  - An OS that enables several programmes to run concurrently (e.g. Unix)
  - More complex than single process systems as the OS must allocate resources to competing processes in a reasonable manner
- Applications
  - Multiprogramming, concurrent servers (e.g. web servers), parallel programmes
- Issues:
  - Synchronisation
  - Scheduling
    - Processor load balancing
    - Static vs dynamic algorithms

OS: Multiprocessor

- Multiprocessor OS: each CPU has its own OS
  - Quick to port from a single-processor OS
  - Difficult to share resources
- Master/Slave: all OS functionality goes to one CPU
  - No multiprocessor concurrency in kernel
  - OS CPU consumption may be large => bottleneck
- Shared OS: all CPUs run a single OS instance
  - The OS must handle multiprocessor synchronisation
    - Only 1 processor can execute in kernel at a time
    - Support fine-grain synchronisation

OS: Embedded

- Embedded systems are special-purpose systems in which the computer is completely encapsulated by the device it controls
- Perform pre-defined tasks that have very specific requirements
- In the simplest embedded systems, there is not distinction between the OS and the application
- Design:
  - Very compact
  - Efficient
  - Leave out unnecessary functions
- Examples:
  - Symbian OS (smartphones)
  - Cisco IOS (routers and switches)
  - iPodLinux (ipods)

OS: Embedded

- Characteristics
  - Designed to do some specific task
  - Not always standalone devices (e.g. Gibson Robot Guitar)
  - Programme instructions written for embedded systems are stored in read-only memory or flash memory chips. Run with limited h/w resources (little memory, small (if any) keyboard and/or screen)
- Software architecture
  - Simple control loop
  - Interrupt controlled system
  - Cooperative multitasking (nonpreemptive)

OS: Handheld

- Handheld computer:
  - Is a pocket-sized computing device with a display screen with touch input or miniature keyboard
  - E.g. PDA, smartphones, Nintendo DS
- OS Examples:
  - Palm
  - Windows mobile (multimedia)
  - BlackBerry
  - Symbian
- Features:
  - Pre-emptive multitasking (quick response)
  - Memory protection (Integrity and security of user data)
  - Conserve resources

OS: Distributed

- An OS which manages a collection of independent computers and makes them appear to the users of the system as a single computer
  - Create, maintain and transfer files in a network
  - High configuration (RAM, high-speed processor)
- Characteristics:
  - Looks like a virtual uni-processor
  - contains n copies of the OS
- Issues:
  - Transparency (location, replication (files), concurrency)
  - Performance
  - Scalability
  - Reliability
  - flexibility

OS: Distributed

- Micro Kernel provides:
  - Inter-process communication mechanism
  - Low-level I/O
  - Some memory management
  - Process management and scheduling
  - All the rest is done at user level
- Tasks:
  - File and directory naming
  - How to share files
  - Load balancing
  - Migration architecture

Network OS (NOS)

- Used to manage networked computer systems
- Create, maintain and transfer files in the network
- An NOS:
  - Controls a network and its message traffic and queues
  - Controls access by multiple users
  - Provides admin functions and security
- Features:
  - Basic support for h/w ports
  - Security (authentication, authorisation, access control)
  - Name services and directory services
  - File services
  - Systems management
  - Network administration
- Note: an NOS is an OS that has been specifically written to implement and maintain networks (e.g. BDS)

Real-Time OS (RTOS)

- Multitasking OS intended for real-time applications
- E.g. industrial robots, spacecraft, industrial control, scientific research equipment
- An RTOS facilitates the creation of a real-time system, but does not guarantee the final result will be real-time
  - Generally (soft real-time)
  - Deterministically (hard real-time)
- Use specialised scheduling algorithms
- Quick and/or predictable response is more important than the amount of work it can perform over a given period of time
- Key factors:
  - Minimal interrupt latency: the time between the generation of an interrupt by a device and the servicing of the device
  - Minimal thread switching latency: the time needed by the OS to switch the CPU to another thread (concurrent task)

Real-Time OS (RTOS)

- 2 basic designs
  - Event-driven (priority scheduling)
    - Switch tasks only when an event of higher priority needs service (pre-emptive priority)
  - Time-sharing designs:
    - Switch tasks on a clock interrupt and on events
  - Newer RTOSs implement time-sharing scheduling with priority driven pre-emptive scheduling
- Some RTOSs allow the application to run in kernel mode for greater system call efficiency
- Memory allocation is critical
  - Speed (must occur in fixed time)
  - Fragmentation
  - Fixed-size block algorithms
- Examples: QNX, RTLinux, VxWorks, Windows CE