Operating System

Henock Mulugeta (Ph.D)

Outline

- What is an Operating System?
- Operating System Functions
- Operating System features
- Operating System variants

Operating System

- A modern computer system consists of:
  - one or more processors,
  - main memory, disks, printers,
  - a keyboard, a display,
  - network interfaces, and
  - other input output devices,
  - All in all, a complex system.
- In order to:
  - Manage all these devices for granting proper function and interaction with each other,
  - To create user friendly environment, and
  - User programs with a simpler interface to the hardware,
- there is a program known as Operating system.

What is an Operating System?

- A program that acts as an intermediary between a user of a computer and the computer hardware.
- Operating system goals:
  - Manage computer system resources.
  - Make the computer system convenient to use.
- Use the computer hardware in an efficient manner.
  - To manage and share/multiplex resources in time and space (resource manager).
    - Time multiplexing – E.g. sharing CPU, printer…
    - Resource multiplexing – E.g. sharing main memory

What is an Operating System?...

An operating system is:

A collection of software components that

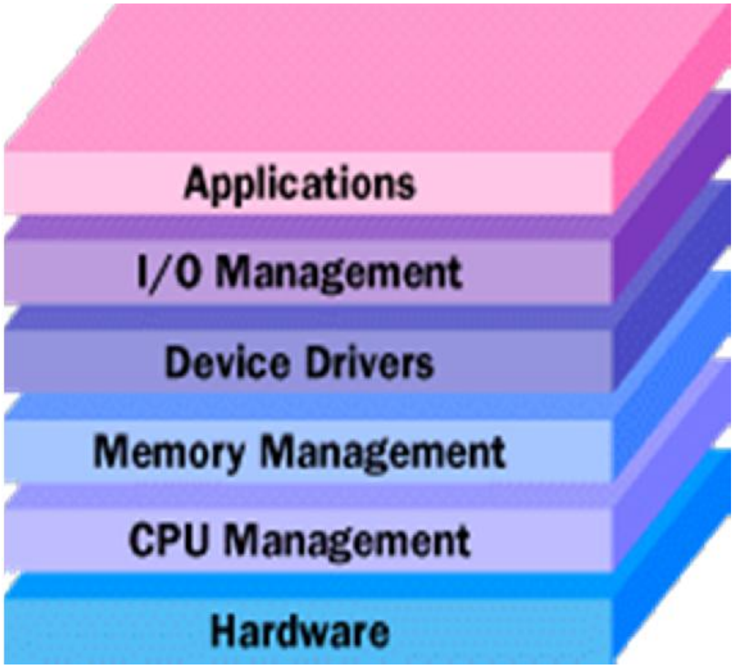
- Provides useful abstractions and
- Manages resources to
- Support application programs, and
- Provide an interface for users and programs

Resource allocator – manages and allocates resources.

Control program – controls the execution of user programs and operations of I/O devices.

What does Operating System do?

- Manages all the resources in a computer (including processor, memory, i/o devices)
- Provides an interface between the hardware and application software.
- Three layers:
  - Inner layer, computer hardware
  - Middle layer, operating system
  - Outer layer, different software



Operating System Functions

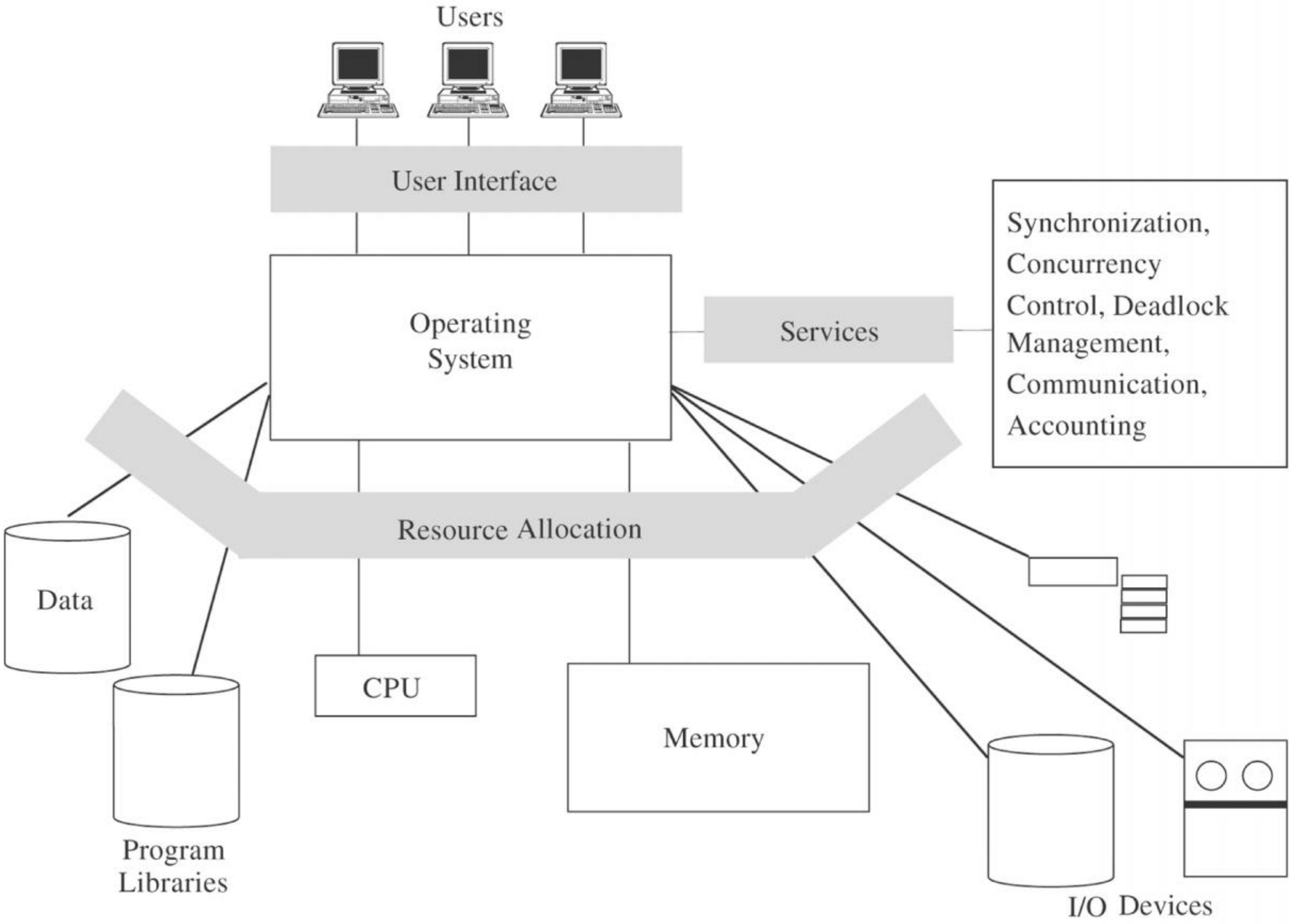
- An operating system’s main functions are:
  - Multiprogramming, multiprocessor
  - Computer resource management
  - Provides a user interface
  - Runs software utilities and programs
  - Schedule jobs
  - Provide tools to configure the operating system and hardware
  - Administers user actions and accounts
  - Enforce security measures

Operating System Functions...

An operating system's main functions are:...

- Schedule processes & multiplex CPU
- Provide mechanisms for IPC and synchronization
- Manage main memory
- Manage other resources (E.g, Input/Output)
- Provide convenient persistent storage (files)
- Maintain system integrity, handle failures
- Enforce security policies (e.g., access control)
- Give users and processes an interface

Operating System functions



Operating System features

- Authentication of users
  - password, passphrase comparison, biometrics, digital authentication (SSL, CA, PKI, Kerberos, DS)
- Mandatory (enforce multilevel security by classifying the data and users into various security classes) and
- Discretionary Access Control (grant privileges to users)
- Protection of memory
  - user space, paging, segmentations
- File and I/O device access control
  - access control matrix
- Enforcement of sharing resources
  - To preserve integrity, consistency (critical section)

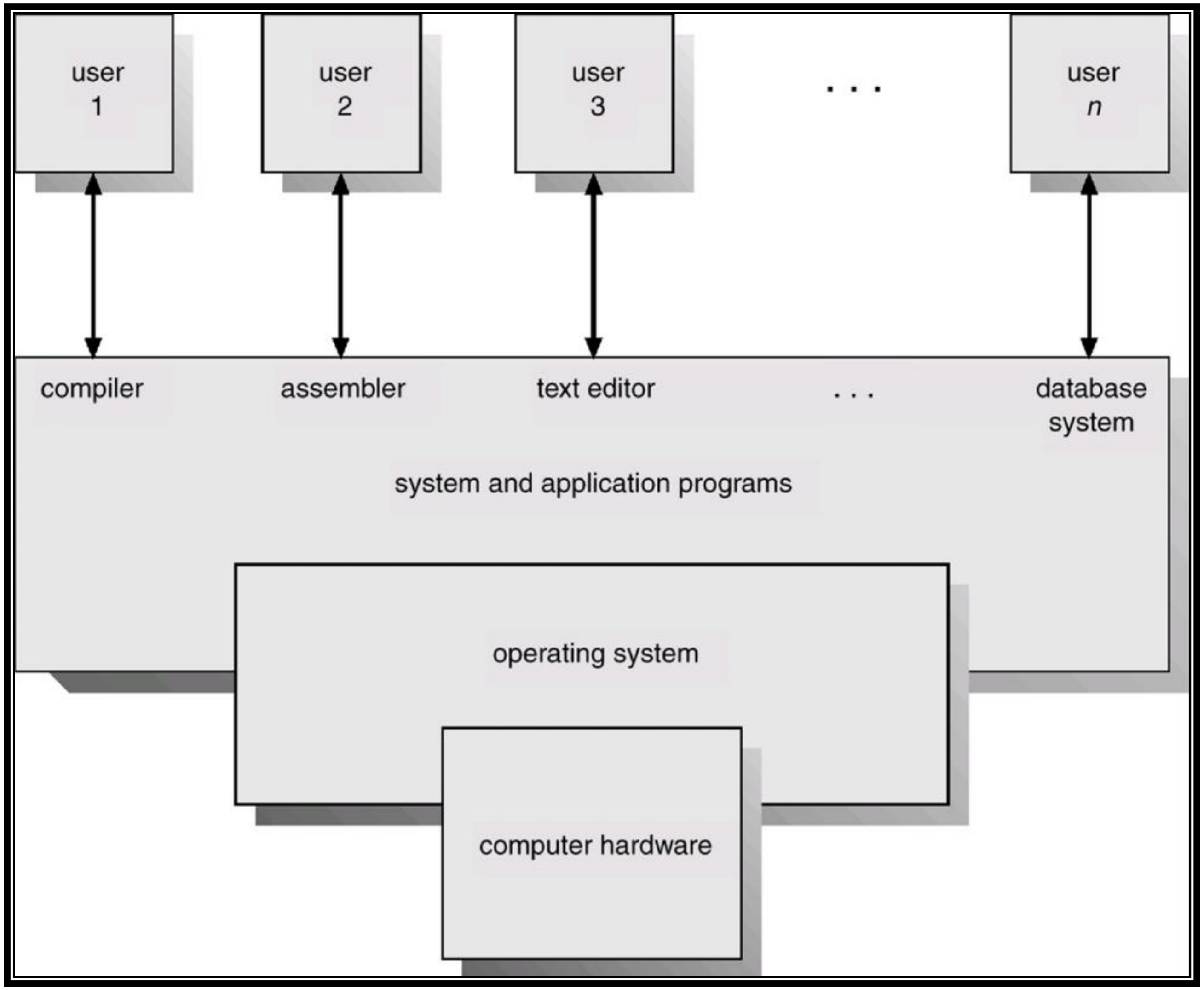
Operating System features...

- Fair service
  - no starvation and deadlock
- Inter-process communication & synchronization
  - Shared variable (e.g, using semaphores)
- Protection of data
  - encryption, isolation
  - ...

Computer System Components

1. Hardware – provides basic computing resources (CPU, memory, I/O devices).
2. Operating system – controls and coordinates the use of the hardware among the various application programs for the various users.
3. Applications programs – define the ways in which the system resources are used to solve the computing problems of the users (compilers, database systems, video games, business programs).
4. Users (people, machines, other computers).

Abstract View of System Components



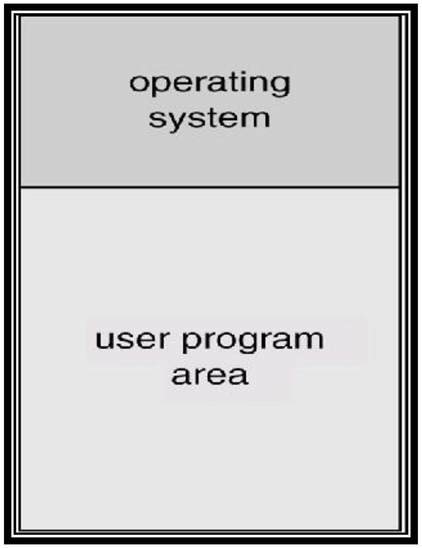
Operating System variants

Mainframe Systems

- The Operating systems for main frames are heavily oriented towards processing of many jobs at once (many I/O devices).
- Reduce setup time by batching similar jobs.
- Automatic job sequencing – automatically transfers control from one job to another.
- They typically offer three kinds of services:
  - Batch System
  - Transaction processing system
  - Time sharing system

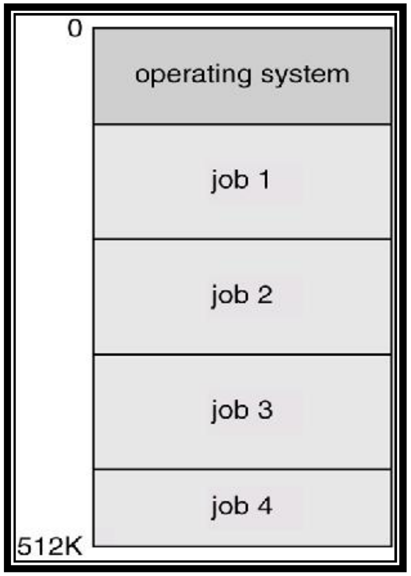
Memory Layout for a Simple Batch System

Batch system - is one that processes routine jobs in the absence of interactive user.



E.g: Multiprogrammed Batch Systems

Several jobs are kept in main memory at the same time, and the CPU is multiplexed among them.



OS Features Needed for Multiprogramming

- I/O routine supplied by the system.
- Memory management – the system must allocate the memory to several jobs.
- CPU scheduling – the system must choose among several jobs ready to run.
- Allocation of resources.

Transaction processing system

- handles large number of small requests.
- Each unit of work is small but the system must handle thousands per second.
- E.g. check processing at bank, airline reservation.

Time-Sharing Systems–Interactive Computing

- The CPU is multiplexed among several jobs that are kept in memory and on disk (the CPU is allocated to a job only if the job is in memory).
  - A job swapped in and out of memory to the disk.
  - On-line communication between the user and the system is provided;
    - when the operating system finishes the execution of one command, it seeks the next "control statement" from the user's keyboard.
  - On-line system must be available for users to access data and code.

Desktop Operating Systems

- Personal computers – computer system dedicated to a single user.
  - I/O devices – keyboards, mice, display screens, small printers.
  - User convenience and responsiveness.

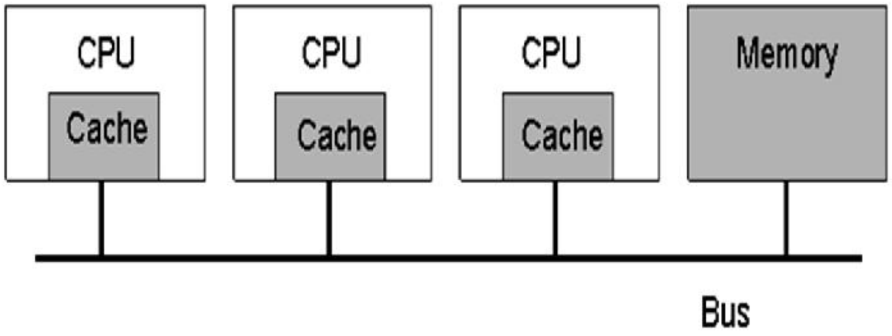
- Often individuals have sole use of computer and do not need advanced CPU utilization of protection features.
- May run several different types of operating systems (Windows, MacOS, UNIX,...)
- Linux (RedHat, OpenSuSe, Fedora, Obuntu,..)

Parallel Systems (Multiprocessor OS)

- Connecting multiples CPU’s into a single system.
- Multiprocessor systems with more than on CPU in close communication.
- Tightly coupled system – processors share bus, memory, clock and peripheral device.
- communication usually takes place through the shared memory.
- Advantages of parallel system:
  - Increased throughput
  - Economical - they can share mass storage, peripherals,...
  - Increased reliability
    - If one fail, the other will take responsibility

Multiprocessors OS...

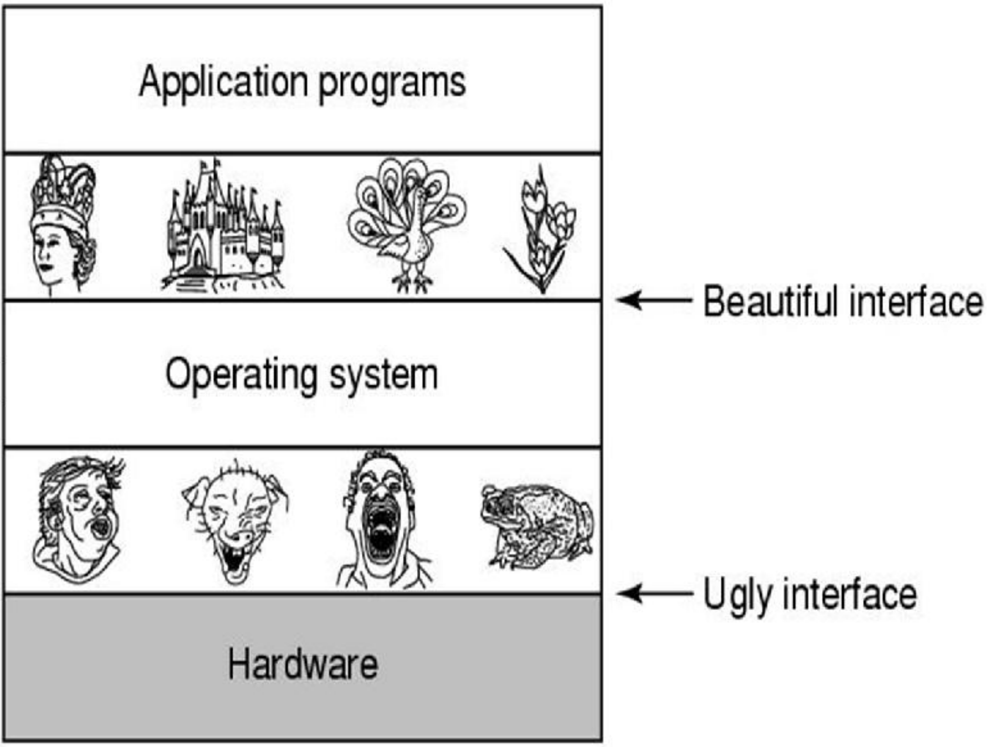
- Like a uniprocessor operating system
- Manage multiple CPUs transparently to the user
- Each processor has its own hardware cache
  - Maintain consistency of cached data
- Shared variable versus message passing



A bus-based multiprocessor

Multiprocessor OS...

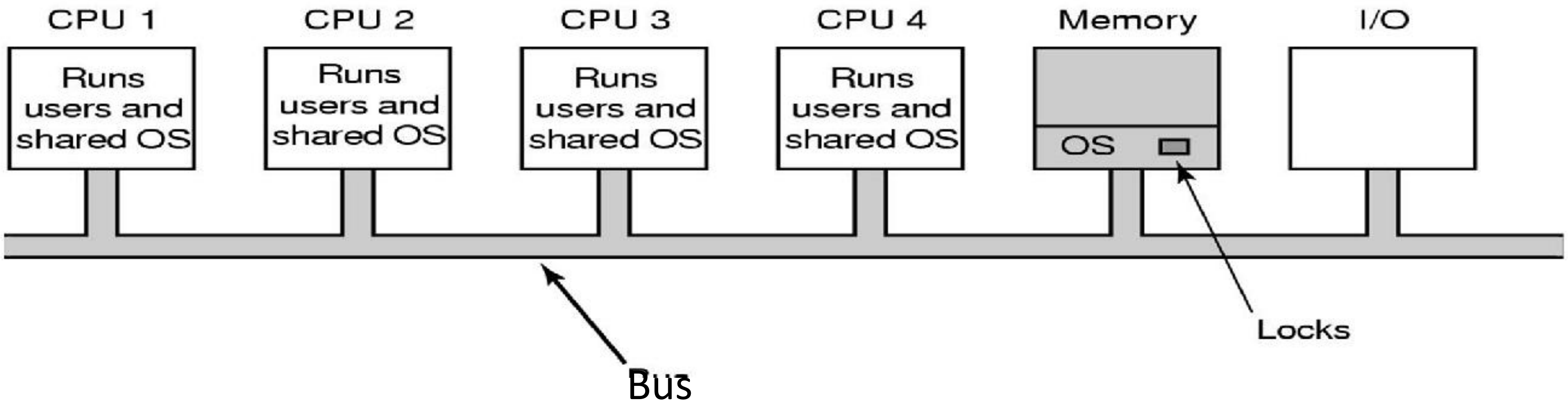
- Are similar to multiprogrammed uniprocessor operating systems in many respects and
  - they perform resource management and
  - hide unpleasant features of the hardware to provide a high-level machine abstraction to the users.



- Are more complex because multiple processors execute tasks concurrently
  - with physical concurrency as opposed to virtual concurrency in multiprogrammed uniprocessors.

Parallel Systems (Multiprocessor OS)...

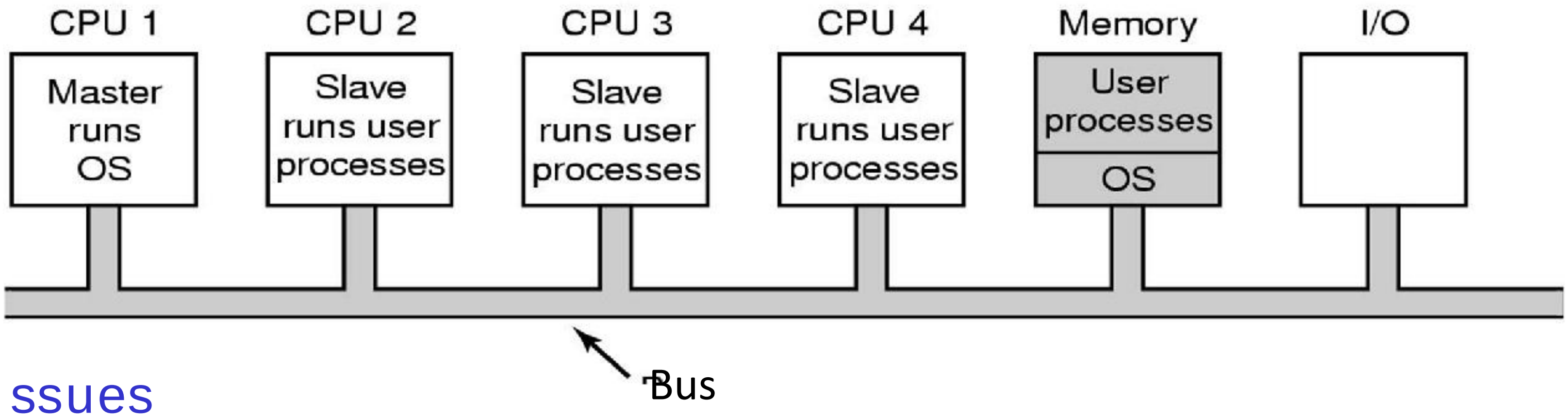
- Symmetric multiprocessing (SMP)
  - Each processor runs an identical copy of the operating system.
  - There is one copy of the supervisor or kernel that can be executed by all processors concurrently.
  - Many processes can run at once without performance deterioration.



- Most modern operating systems support SMP
- It permits the parallel execution of a single task.
- Examples : Hydra OS

Parallel Systems (Multiprocessor OS)...

- Asymmetric multiprocessing
  - Master-slave. Operating system in master processor.
  - Each processor except master is assigned a specific task
  - master processor schedules and allocated work to slave processors.
  - More common in extremely large systems


CPU 1
CPU 2
CPU 3
CPU 4
Memory
I/O
Master runs OS
Slave runs user processes
Slave runs user processes
Slave runs user processes
User processes
OS
Bus

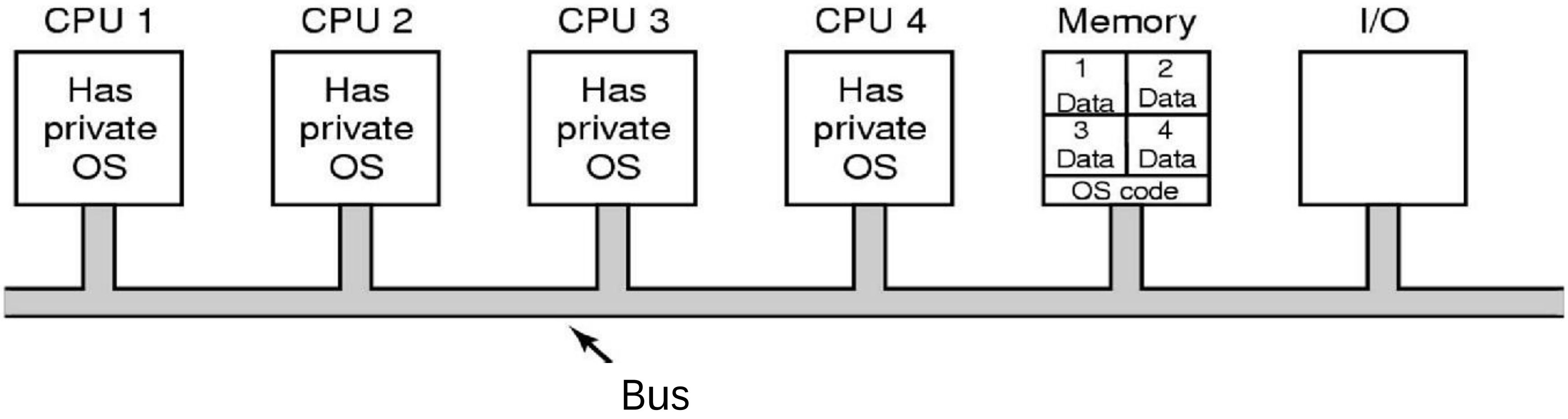
Issues

- Failure of the master processor.
- The master can become a bottleneck
- Examples : Cyber 170 and DEC(Digital Equipment corporation)

Multiprocessor OS...

Separate supervisor Configuration

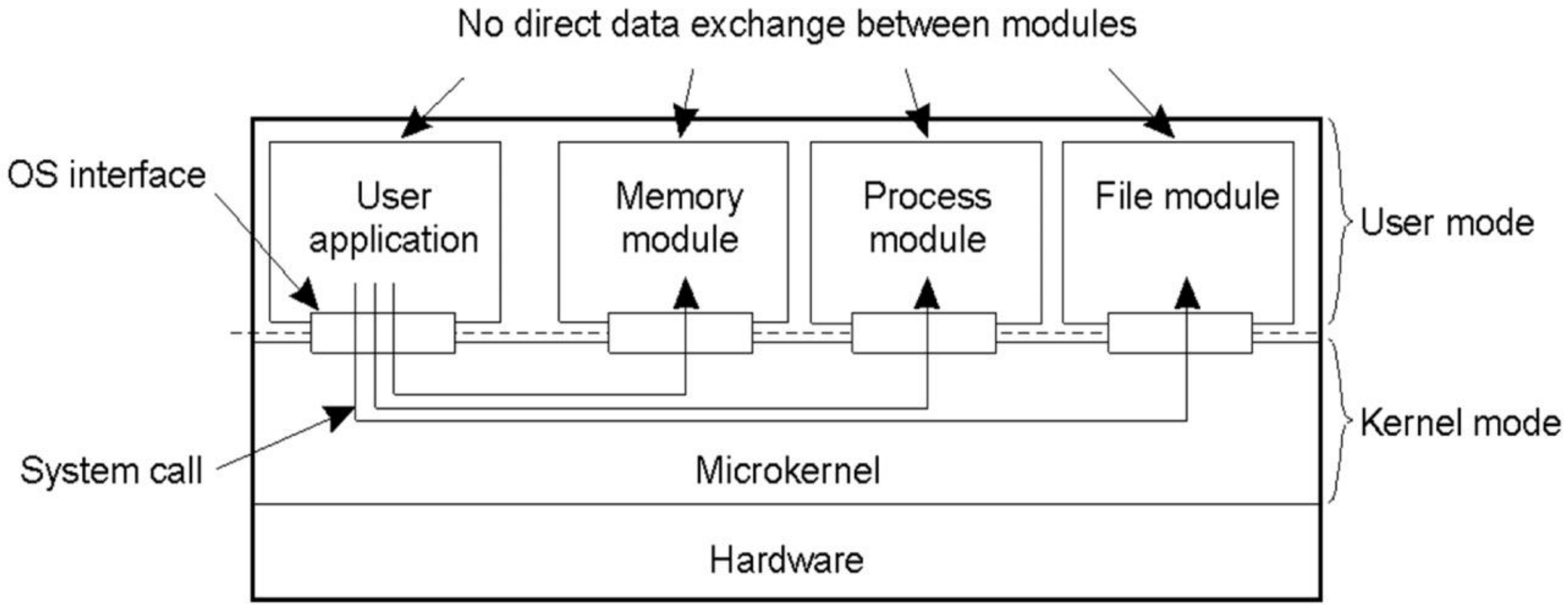
- Each CPU has its own operating system
- There is very little coupling among processors and
- each processor acts as an autonomous, independent system.



- There are some common data structures for the interaction among processors

The access to which is protected by using some synchronization mechanism (such as semaphores).

Uniprocessor Operating Systems

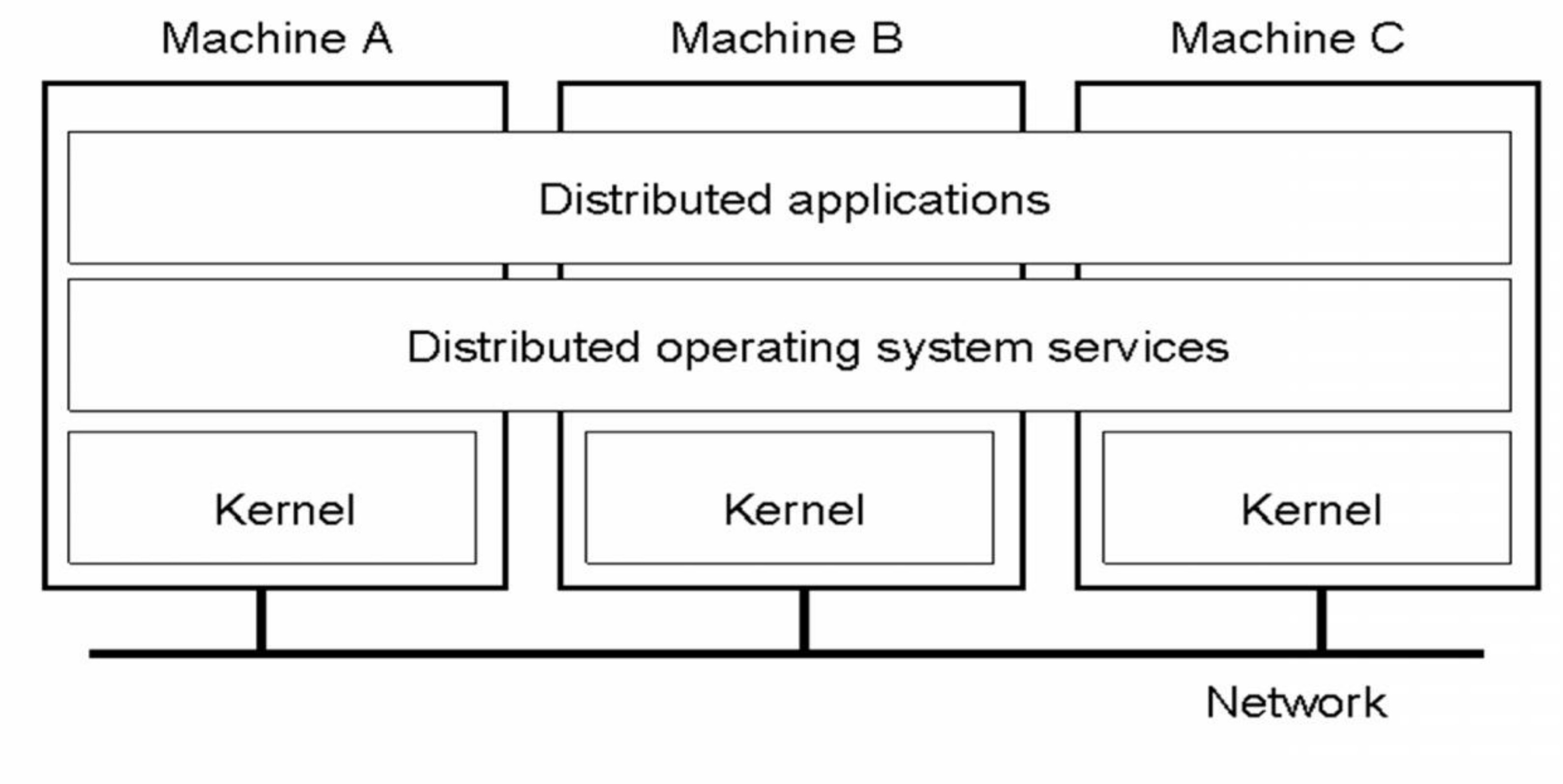


Separating applications from OS code through a microkernel

Multicomputer (DS) Operating Systems

More complex than multiprocessor OS

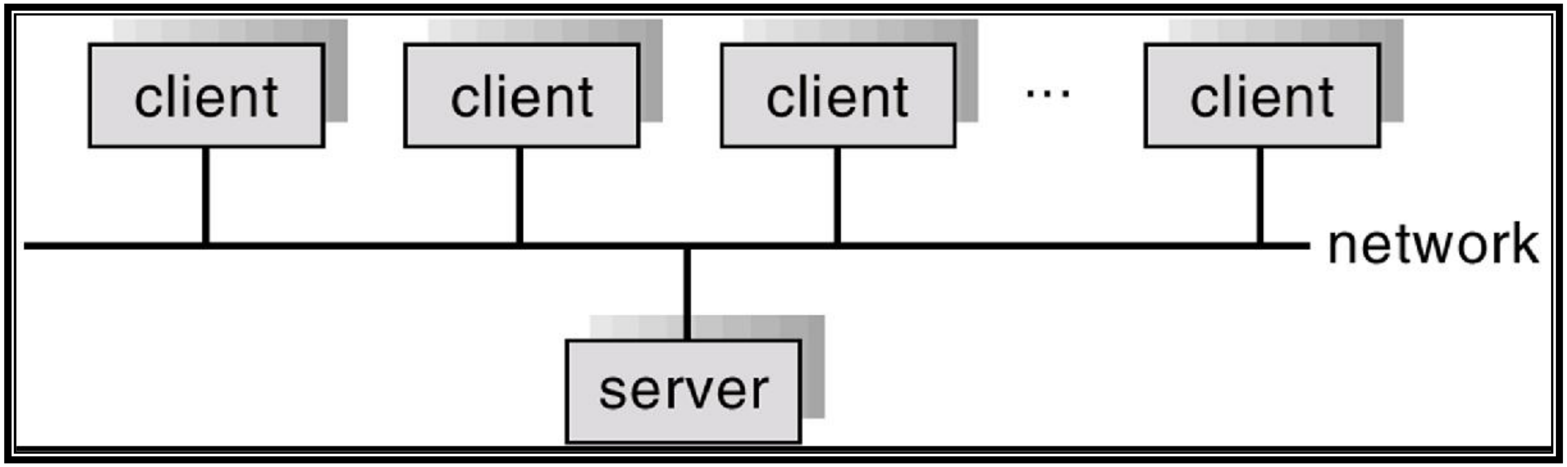
– Because communication has to be done through explicit message passing



General structure of a multicomputer operating system

Server Operating System

- They run on servers, which are very large personal computers, workstations, or even mainframes.
- They serve multiple users at once over a network.
- They allow the users to share HW and SW resources.
- E.g. Server can provide print service, file service, or web service.
- Typical server operating systems are UNIX, Window 2003, and Linux...

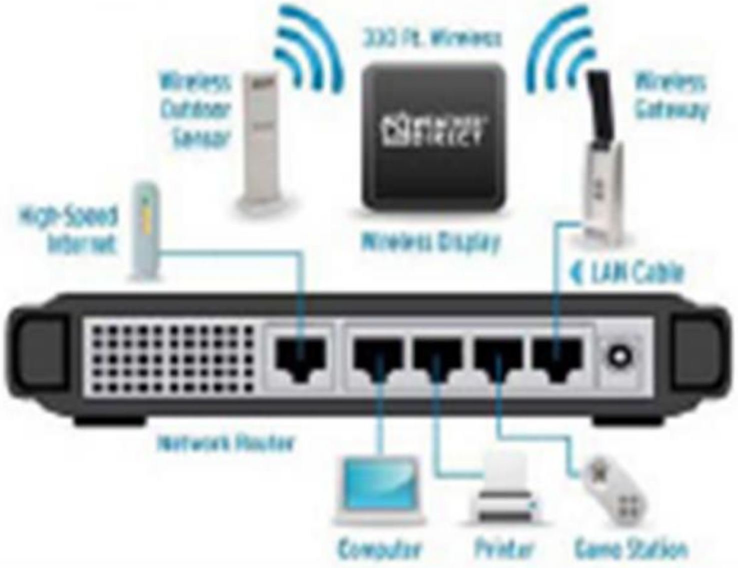


Network Operating System

- Bridges, Routers, Wireless access points

Bridge

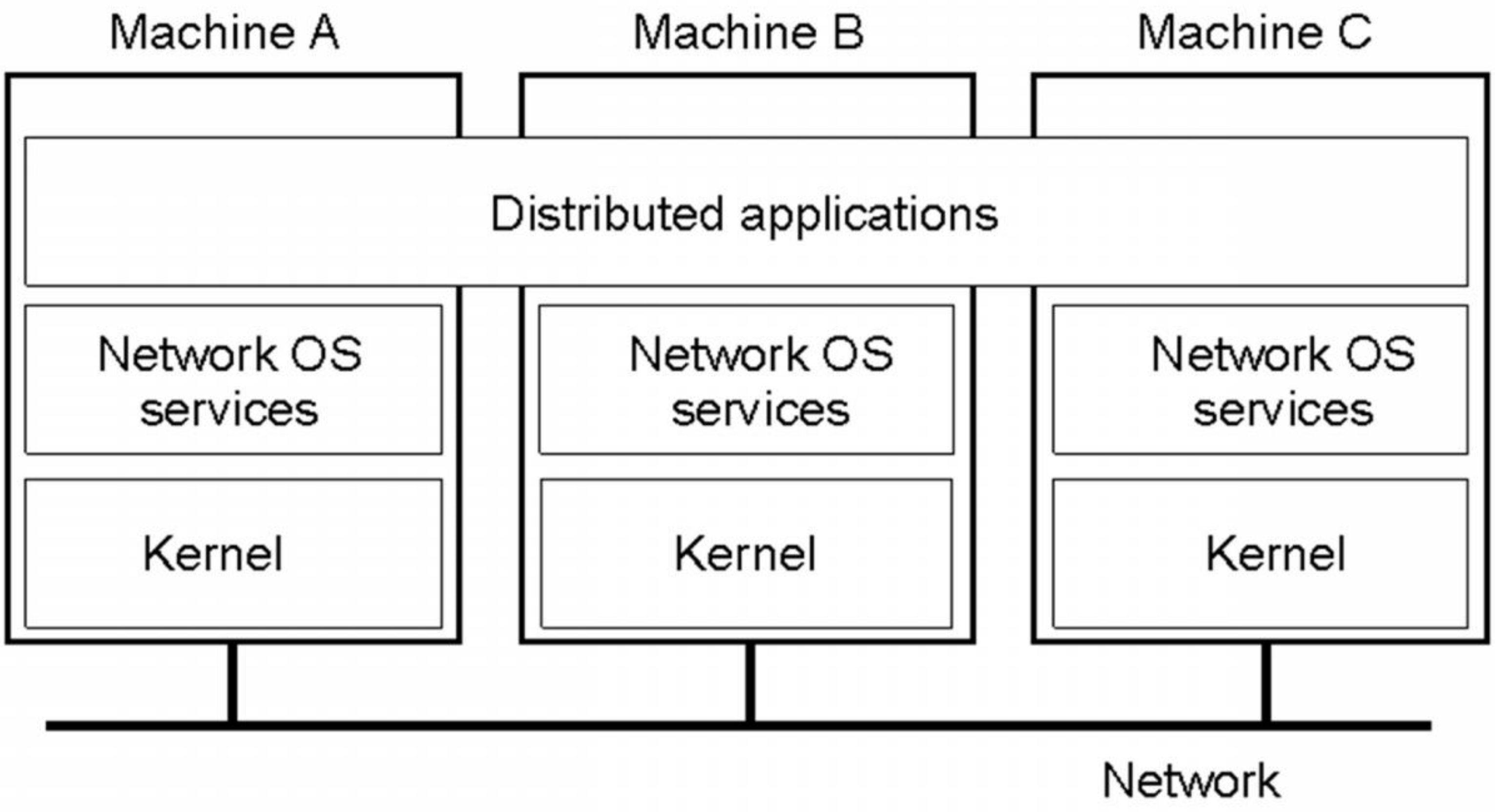
Wireless access point



Router

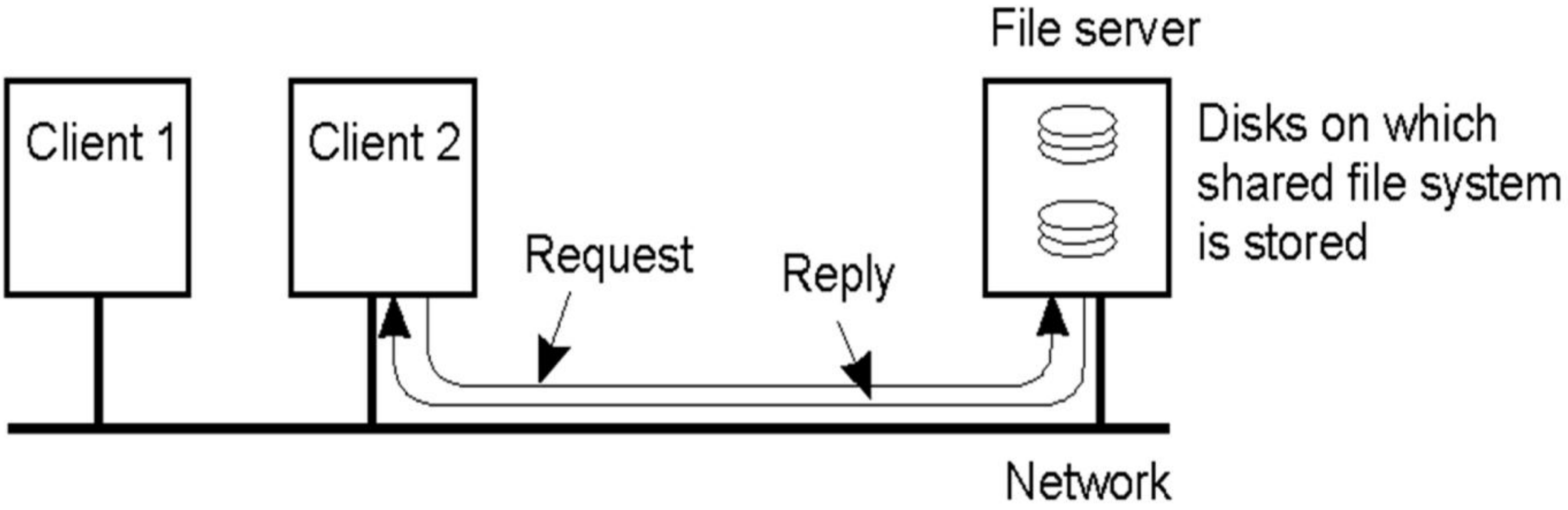
Network Operating System...

General structure of a network operating system



Network Operating System...

Employs a client-server model



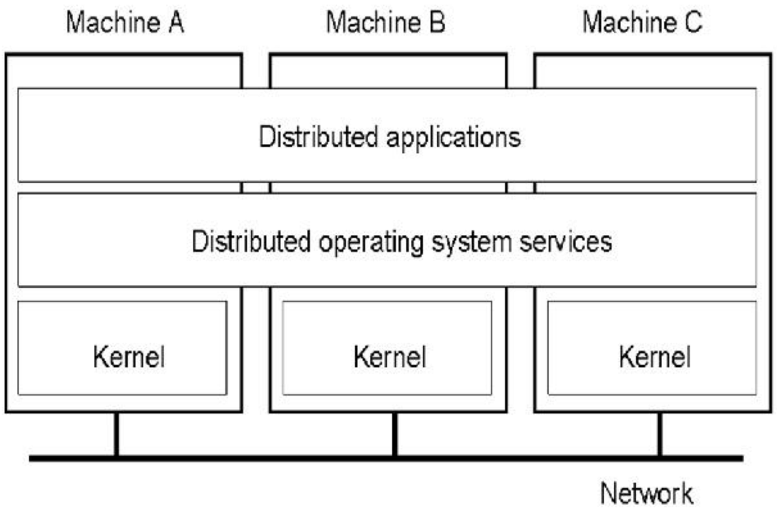
Two clients and a server in a network operating system

Network Operating System...

- Users are aware of multiplicity of machines.
- Access to resources of various machines is done explicitly by
  - Remote logging into the appropriate remote machine.
  - Transferring data from remote machines to local machines, via the File Transfer Protocol (FTP) mechanism.

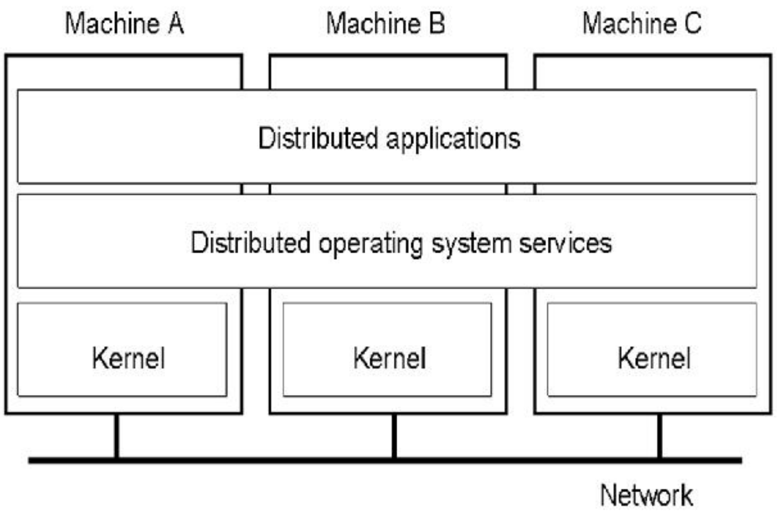
Distributed Operating System

- Distribute the computation among several physical processors.
- Loosely coupled system – each processor has its own local memory, clock, peripheral devices,…
- processors communicate with one another through various communications lines, such as high-speed buses or telephone lines.
- Advantages of distributed systems.
  - Resources Sharing
  - Computation speed up – load sharing
  - Reliability
  - Communications



Distributed Operating System...

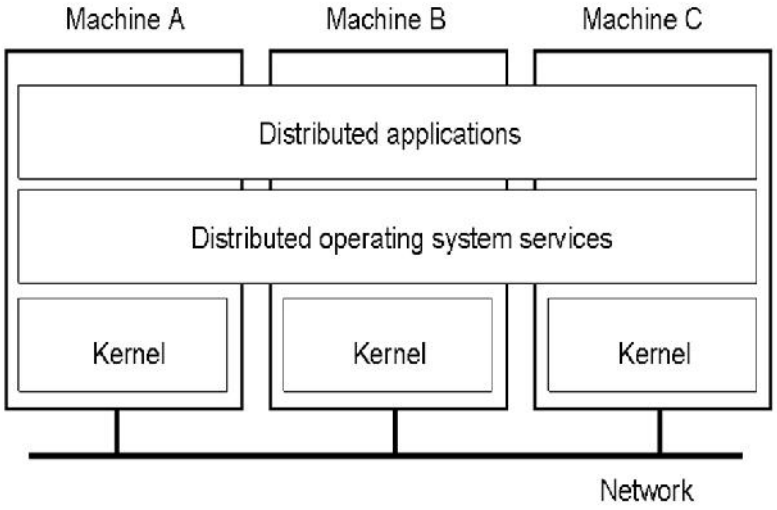
- Users not aware of multiplicity of machines.
- Manages resources in a distributed system
  - Seamlessly and transparently to the user
- Looks to the user like a centralized OS
  - But operates on multiple independent CPUs
- Provides transparency
  - Location, migration, concurrency, replication,…
- Presents users with a virtual uniprocessor



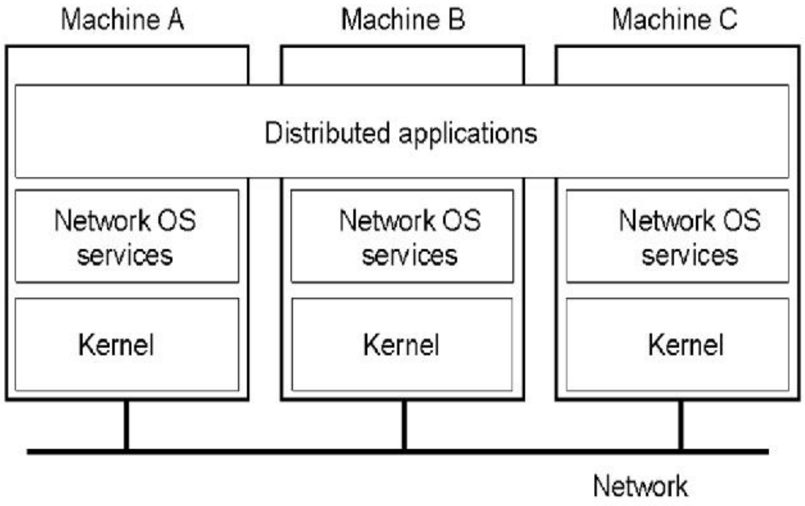
Distributed Operating Systems...

- Requires networking infrastructure.
- Local area networks (LAN) or Wide area networks (WAN)
- May be either client-server or peer-to-peer systems.

Distributed OS vs. Network OS.



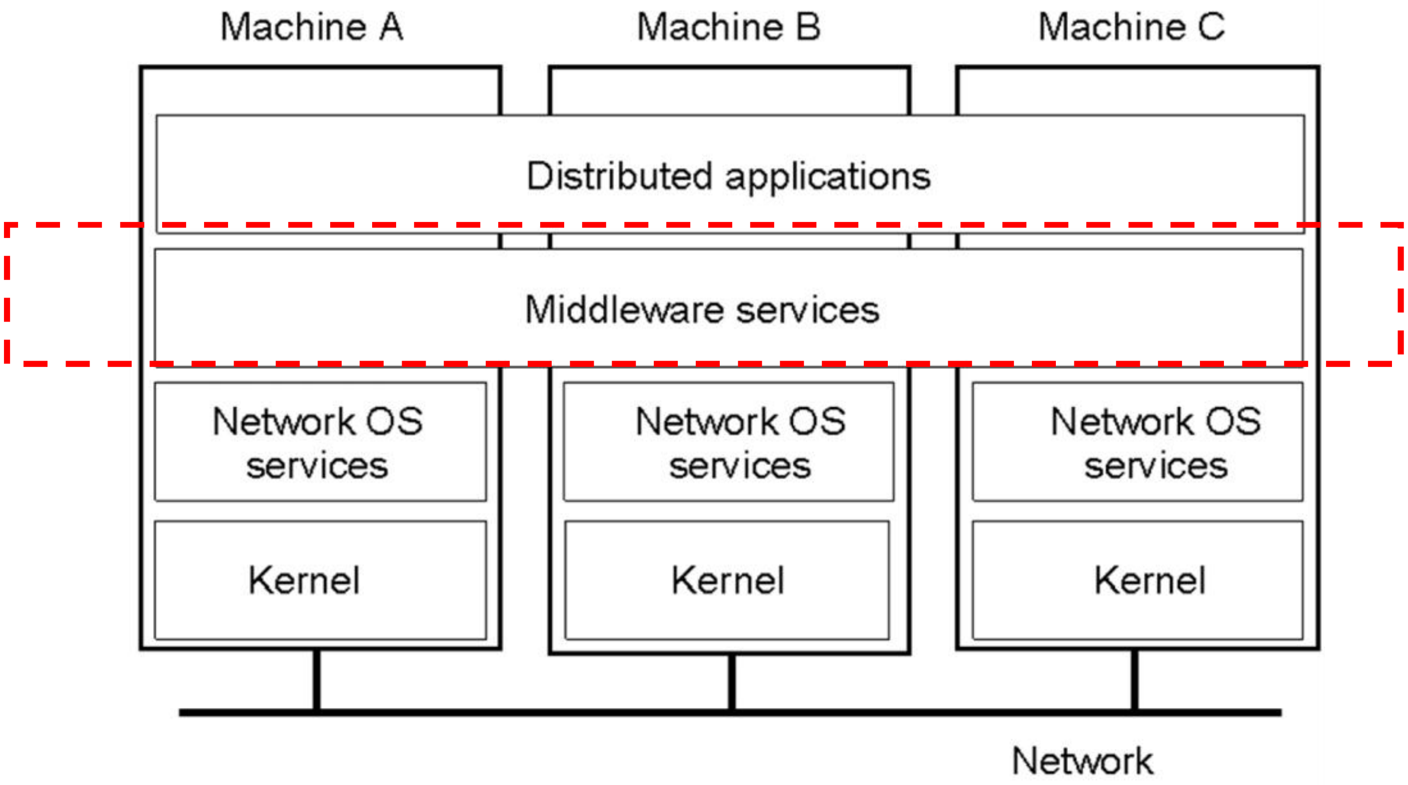
- ➢ User is not aware of the multiple CPUs.
- ➢ Each machine runs a part of the Distributed Operating System.
- ➢The system is fault-tolerant.



- ➢ User is aware of the existence of multiple CPUs.
- ➢ Each machine has its own private Operating System.
- ➢ The system is not fault-tolerant.

Positioning Middleware

General structure of a distributed system as middleware



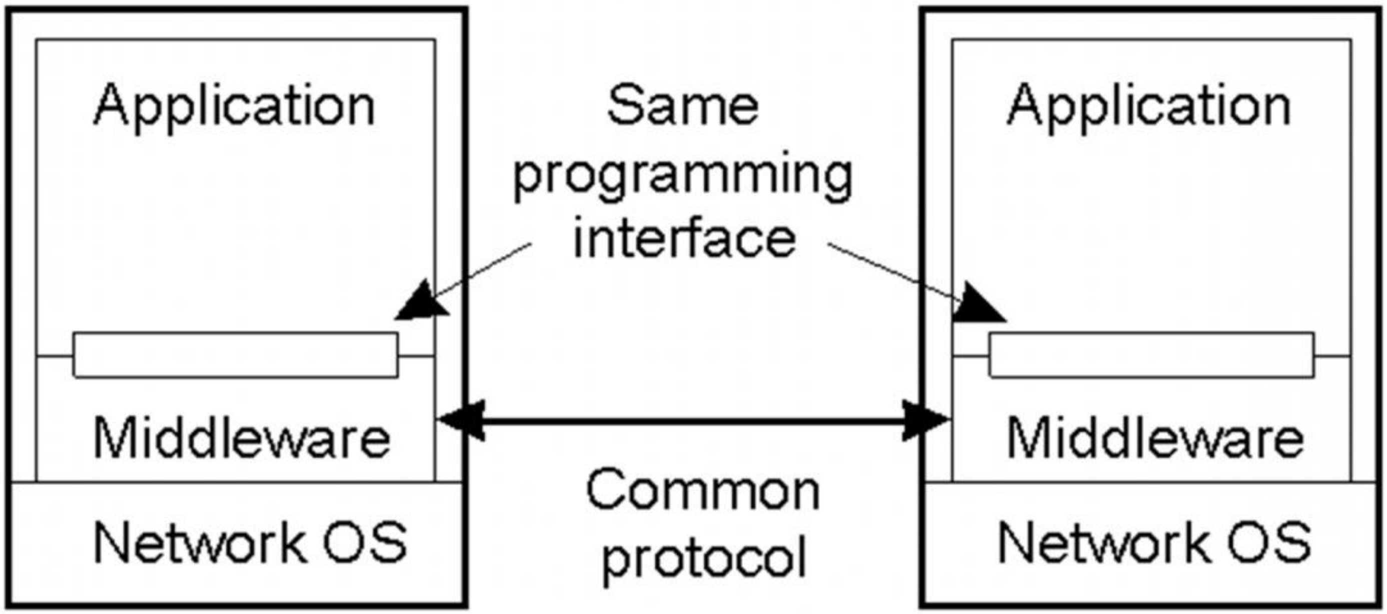
Types of Operating Systems

System	Description	Main Goal
DOS	Tightly-coupled OS for multi-processors and homogeneous multicomputers	Hide and manage hardware resources
NOS	Loosely-coupled OS for heterogeneous multicomputers (LAN and WAN)	Offer local services to remote clients
Middleware	Additional layer a top of NOS implementing general-purpose services	Provide distribution transparency

An overview of

- DOS (Distributed Operating Systems)
- NOS (Network Operating Systems)
- Middleware

Middleware and Openness



- In an open middleware-based distributed system,
  - the protocols used by each middleware layer should be the same,
  - the interfaces they offer to applications should also be the same

Role of Middleware (MW)

- MW tried to provide the illusion that a collection of separate machines was a single computer.
- MW also supports seamless access to remote services, doesn't try to look like a general-purpose OS
- Examples of Middleware
  - CORBA (Common Object Request Broker Architecture)
  - DCOM (Distributed Component Object Management)
  - RPC (Remote Procedure Call)
  - RMI (Remote Method Invocation)
  - Socket (TCP,UDP)

Middleware Examples...

- All of the previous examples support communication across a network:
- They provide:
  - protocols that allow a program running on one kind of computer,
  - using one kind of operating system,
  - to call a program running on another computer
  - with a different operating system
- The communicating programs must be running the same middleware.

Comparison between Operating Systems

Item	Distributed OS		Network OS	Middleware-based OS
	Multiproc.	Multicomp.		
Degree of transparency	Very High	High	Low	High
Same OS on all nodes	Yes	Yes	No	No
Number of copies of OS	1	N	N	N
Basis for communication	Shared memory	Messages	Files	Model specific
Resource management	Global, central	Global, distributed	Per node	Per node
Scalability	No	Moderately	Yes	Varies
Openness	Closed	Closed	Open	Open

A comparison between multiprocessor OS, multicomputer OS, network OS, and middleware based distributed systems

Real-Time Operating Systems

- Often used as a control device in a dedicated application such as controlling scientific experiments, industrial control systems,...
- These systems are characterized by having time as a key parameter.
  - E.g. Governor, which is used to control the flow of water in the production of power, Industrial process controls system.
- If the action absolutely must occur at a certain moment (or within a certain range), we have a hard real time OS.
  - E.g. Governor, flight control system, air bag in a car...
- Another kind of real time system is a soft real time operating system, in which missing an occasional deadline is acceptable.
  - E.g. Digital audio or multimedia systems.

Multimedia Applications

Classes of MM applications:

1) Streaming stored audio and video

   - lectures, songs, movies
   - user interactive (pause/resume)

2) Streaming live audio and video

   - broadcast of radio/TV over internet
   - non-interactive

3) Real-time interactive audio and video

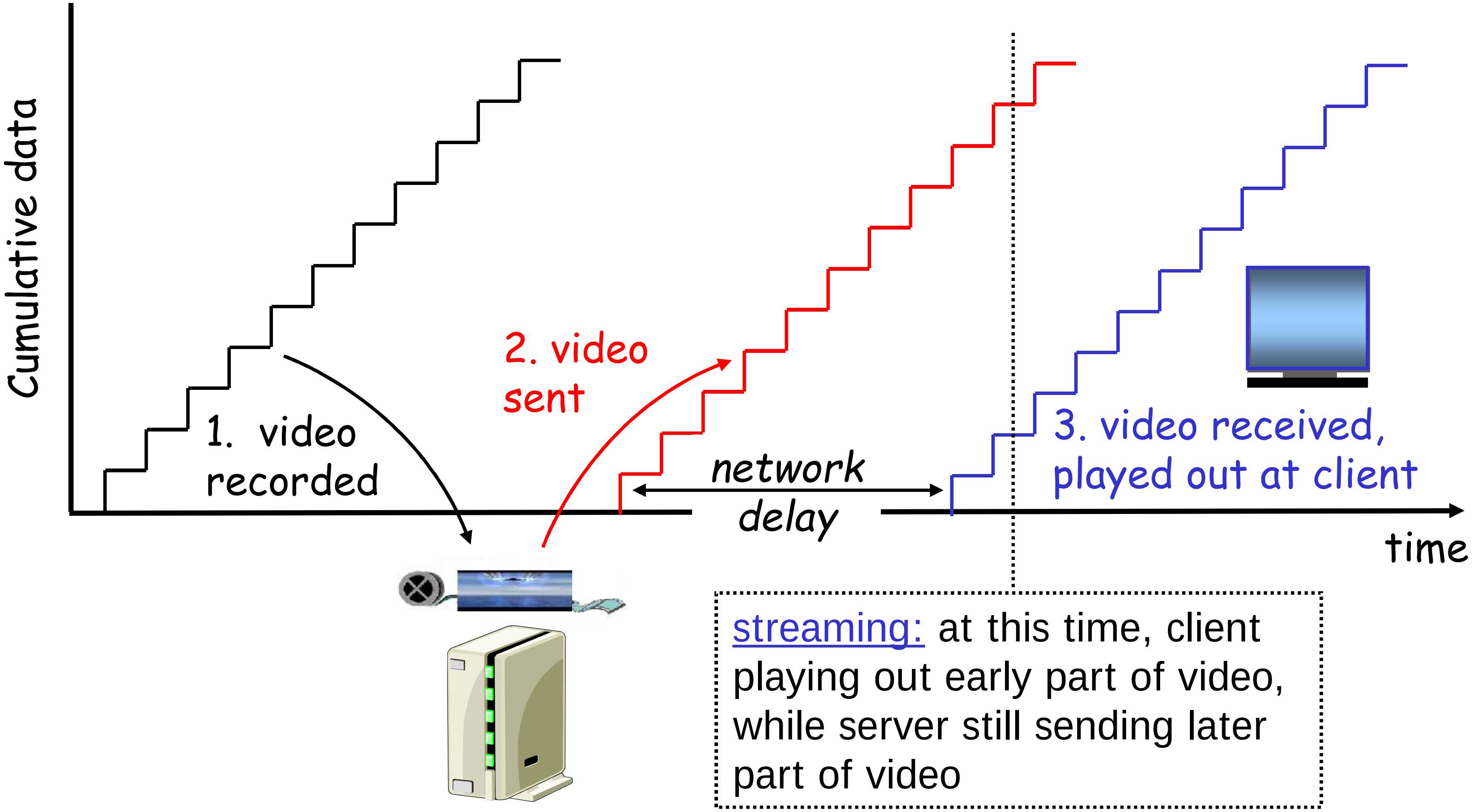
   - internet phone, audio/video conferencing

Fundamental characteristics:

- Typically delay sensitive
  - end-to-end delay
  - delay jitter
- But loss tolerant: infrequent losses cause minor malfunctions
- Opposites of data, which are loss intolerant but delay tolerant.

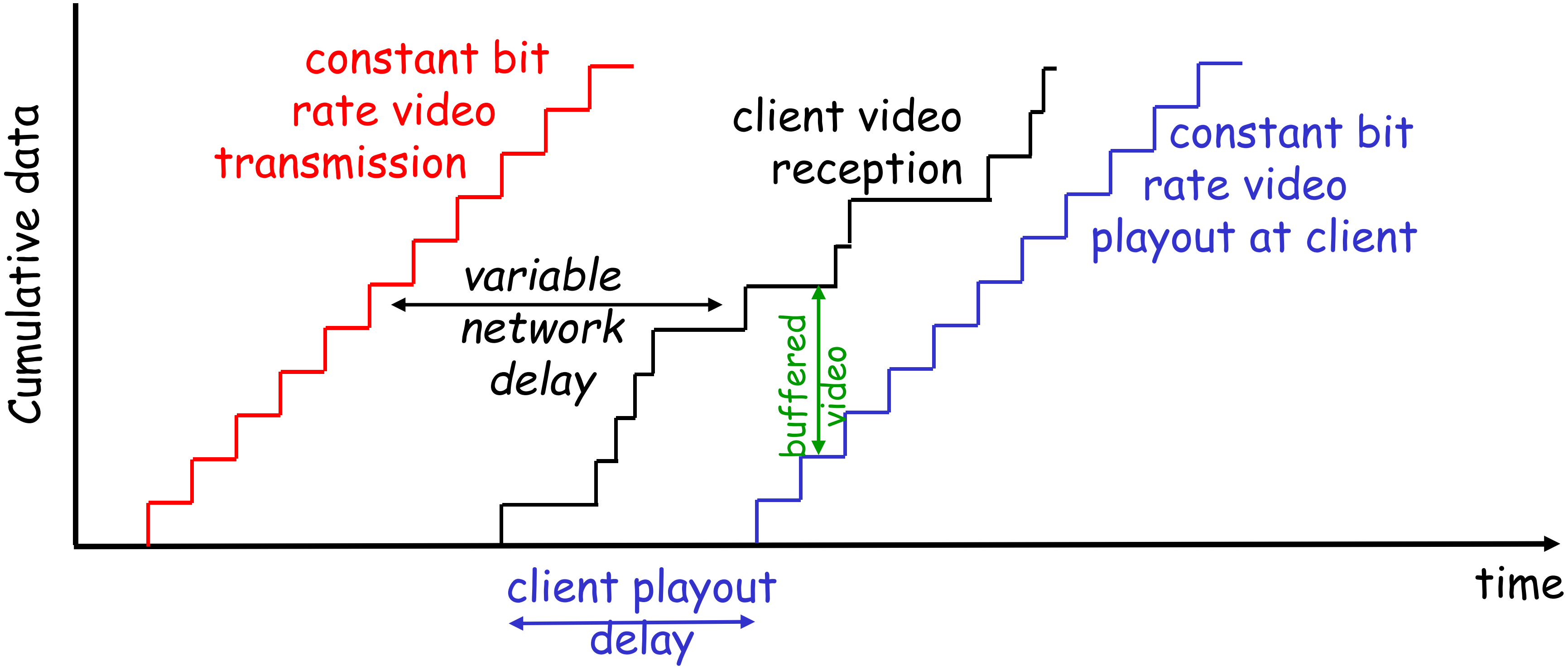
Jitter is the variability of packet delays within the same packet stream

Streaming Stored Multimedia



streaming: at this time, client playing out early part of video, while server still sending later part of video

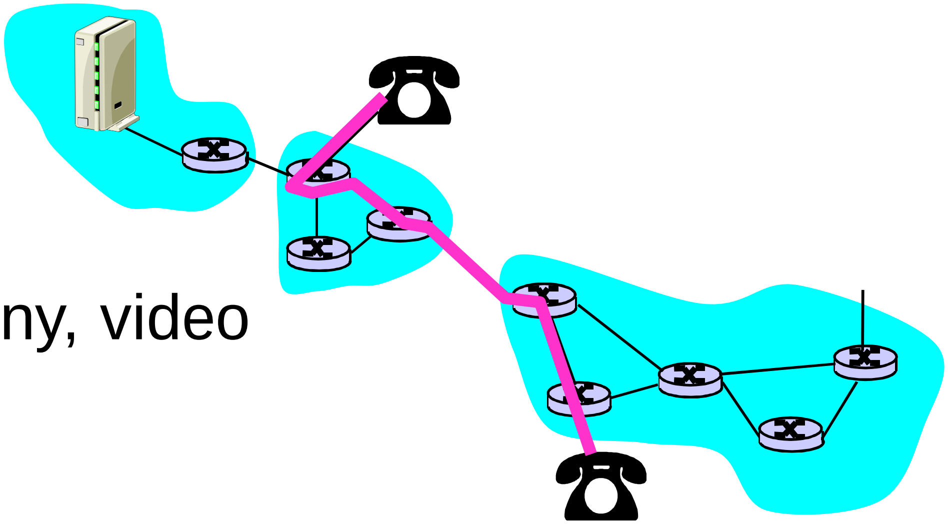
Streaming Multimedia: Client Buffering



- Client-side buffering - playout delay compensate for network-added delay, delay jitter

Interactive, Real-Time Multimedia

Example of soft real time system



- ❒ applications: IP telephony, video conference, distributed interactive worlds

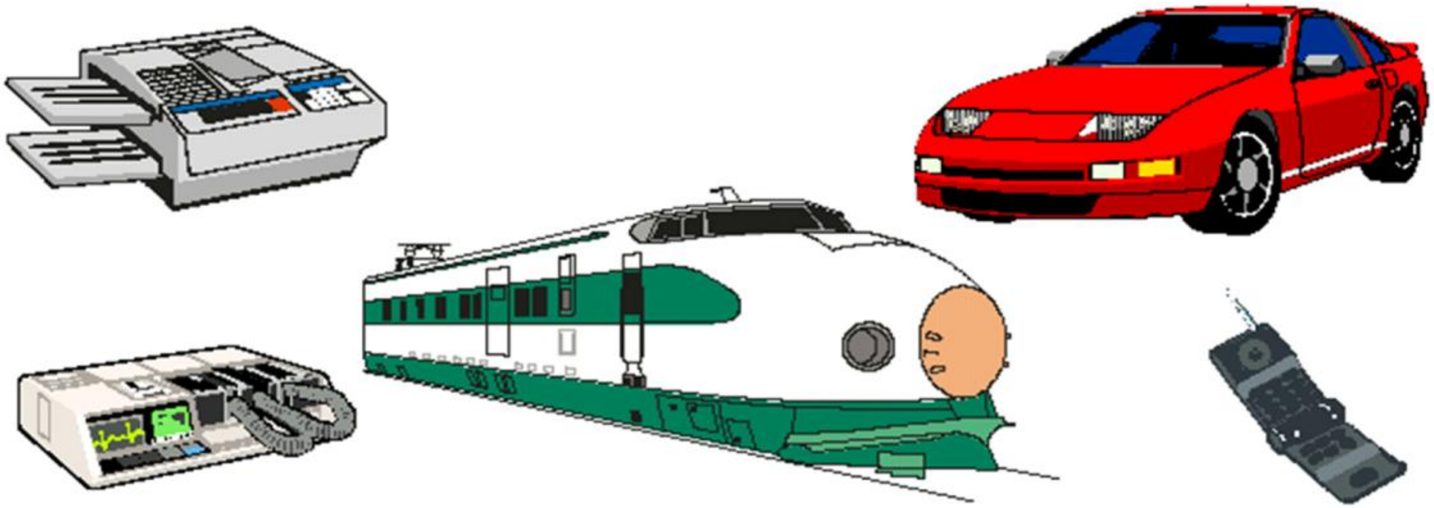
- ❒ end-end delay requirements:
  - ❍ audio: < 150 msec good, < 400 msec OK
    - includes application-level (packetization) and network delays
    - higher delays noticeable, damage interactivity

Embedded Operating Systems

- Personal Digital Assistants (PDAs)
- Cellular telephones, smart phones,...
- E.g, Windows CE (Consumer Electronics)
- PalmOS, Android OS,...
- Issues:
  - Limited memory
  - Slow processors
  - Small display screens.

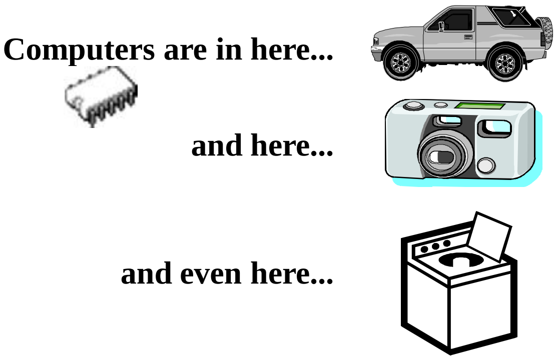
What is an embedded system?

Embedded System = Computer Inside a Product.



Embedded systems...

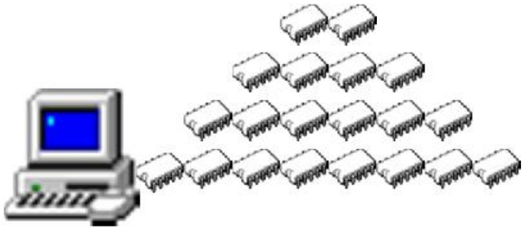
- Embedded computing systems
  - Computing systems embedded within electronic devices
  - Billions of units produced yearly, versus millions of desktop units
  - Perhaps >50 per household and per automobile
  - A lot more programming is done for embedded systems than desktop computers or servers



Computers are in here...

and here...

and even here...



Lots more of these, though they cost a lot less each.

Embedded Operating Systems…

- Many different platforms:
  - J2ME
  - Android
  - Apple iPhone
  - Microsoft Windows Mobile
  - Blackberry
  - PalmWebOS
  - Nokia (C/C++, Python)
    - Symbian (S60, S80)

Product: Pavion Portable GPS Navigation & Multimedia System

Microprocessor: ARM , DSP

OS: Windows CE

Also plays MP3s and Videos

Product: Cannon EOS 3D Digital Camera

Microprocessor: DIGIC II Image Processor

Media players are embedded systems.
Microsoft’s Zune Multimedia player uses an ARM processor and the Windows CE Operating System.

Product: Microsoft’s Zune Portable Media Device

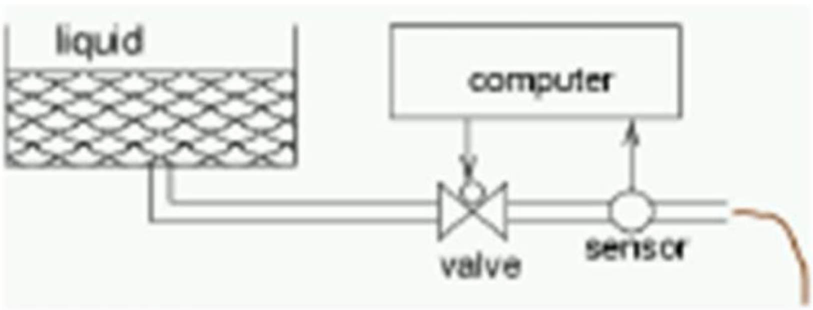
Microprocessor: ARM

OS: Windows CE

Embedded OS...

Industrial Automation

- Process and plant control systems in nuclear power plants, Hydro power plants, industries.



Embedded OS...

Automotive Electronics

Product: S class Mercedes
Microprocessors: around
100 embedded processors !

- Dashboard electronics such as the radio, air conditioning, and satellite navigation system, Airbags,...

Efficient automatic gearboxes, media, safety ...

Embedded OS...

Aircrafts

- Flight control systems,
- Pilot information systems,
- Power supply system,
- Entertainment system,

Embedded OS...

Product: Samsung BlackJack II Smartphone

Microprocessor: TI OMAP (ARM + DSP)

OS: Windows Mobile 6 (CE)

Embedded Mobile Technologies

- **Technologies:**
  - SMS
    - Communication layer for local apps
    - SMS applications
  - Local Applications:
    - Java 2 Micro Edition (J2ME)
    - Python (Nokia)
    - Android
    - Apple
    - Etc, etc, etc…
  - Mobile Web
    - Internet access over 2G/3G/4G
    - Communication layer for local apps
  - Telephony Apps
    - Phone menus
    - Voice recognition

Local Embedded Mobile Applications

- Fast, rich user interfaces
  - Forms, menus, alerts, buttons, pictures, videos, textboxes, touch screen, orientation
- Access to device features
  - Location (GPS, Google maps, compass, …)
  - Voice / speaker
  - Storage
  - Camera
  - Wi-fi (local networking)
  - Bluetooth,IR, RFID, NFC
  - Mobile network (SMS, data)

Embedded device apps dev’t Technology Tradeoffs

	SMS Application	Mobile Web	Local Application
Installed Base	Everyone	Mostly Everyone	Many and growing
Portability	Best	Different Flavors	Phone Specific /Platform specific
Bandwidth Req.	Low	High (nothing local)	Variable (local interaction /cache locally)
User Interface / User Experience	Simple	Adequate	Rich & Responsive
Advanced Features	None	Few	Yes! (GPS, orientation, local networking)

WSN Operating systems

- TinyOS
- FreeRTOS
- RETOS
- μC/OS II
- AMBIENT RT
- Nano-Qplus
- Android
- Windows CE
- Contiki

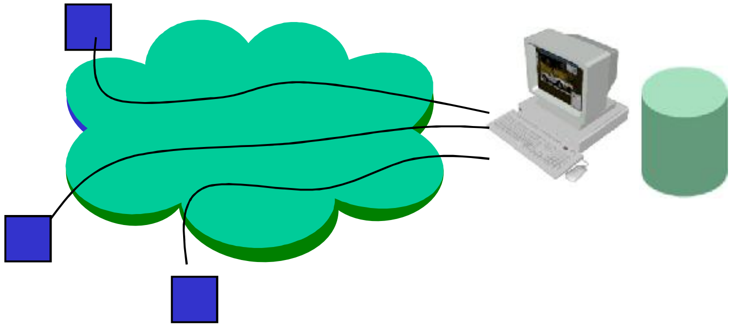
The “Linux” of sensor node OS

Developed by ETRI

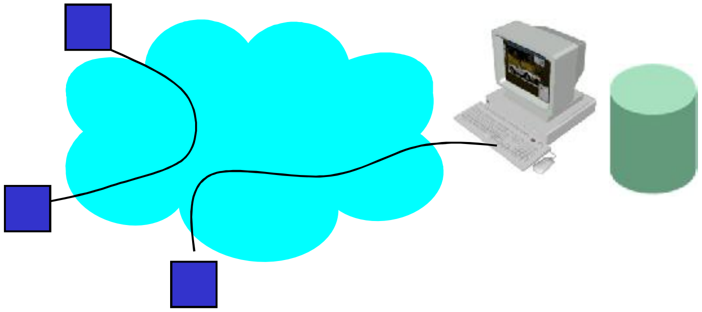
Developed for mobile phone

Wireless Sensor Networks

- Control (actuator) and monitor (sensor) networks
- Initial approach
  - Data collecting networks
  - Wired



- Present approach
  - Ad-hoc capacity → Local processing
  - Wireless → Easy deployment



Embedded OS

Android Overview

- Android was originally created by Andy Rubin as an operating system for mobile phones,
    - around the dawn of this twenty-first century.
- In 2005, Google acquired Android Inc., and made Andy Rubin the Director of Mobile Platforms for Google.
- Over the past decade, Android has matured and evolved into an extremely reliable, embedded operating system platform.
- Having gone from version 1.0 to stable versions at 1.5, 1.6, 2.0, 2.1, 2.2, 2.3, and, recently, 7.1

Android Overview…

- Android has the power of a complete computer operating system.
- It is based on:
  - Linux open source platform and
  - Oracle’s (formerly Sun Microsystems’s) Java, one of the world’s most popular programming languages.
- Android is used as the primary operating system for a rapidly expanding range of consumer electronics, including:
  - Smartphones
  - Netbooks
  - MP4 players
  - Tablets
  - Internet TVs
  - Some desktop systems

Android Phones

HTC G1

Samsung i7500

HTC Hero

Motorola Cliq

Sony X10

HTC Magic

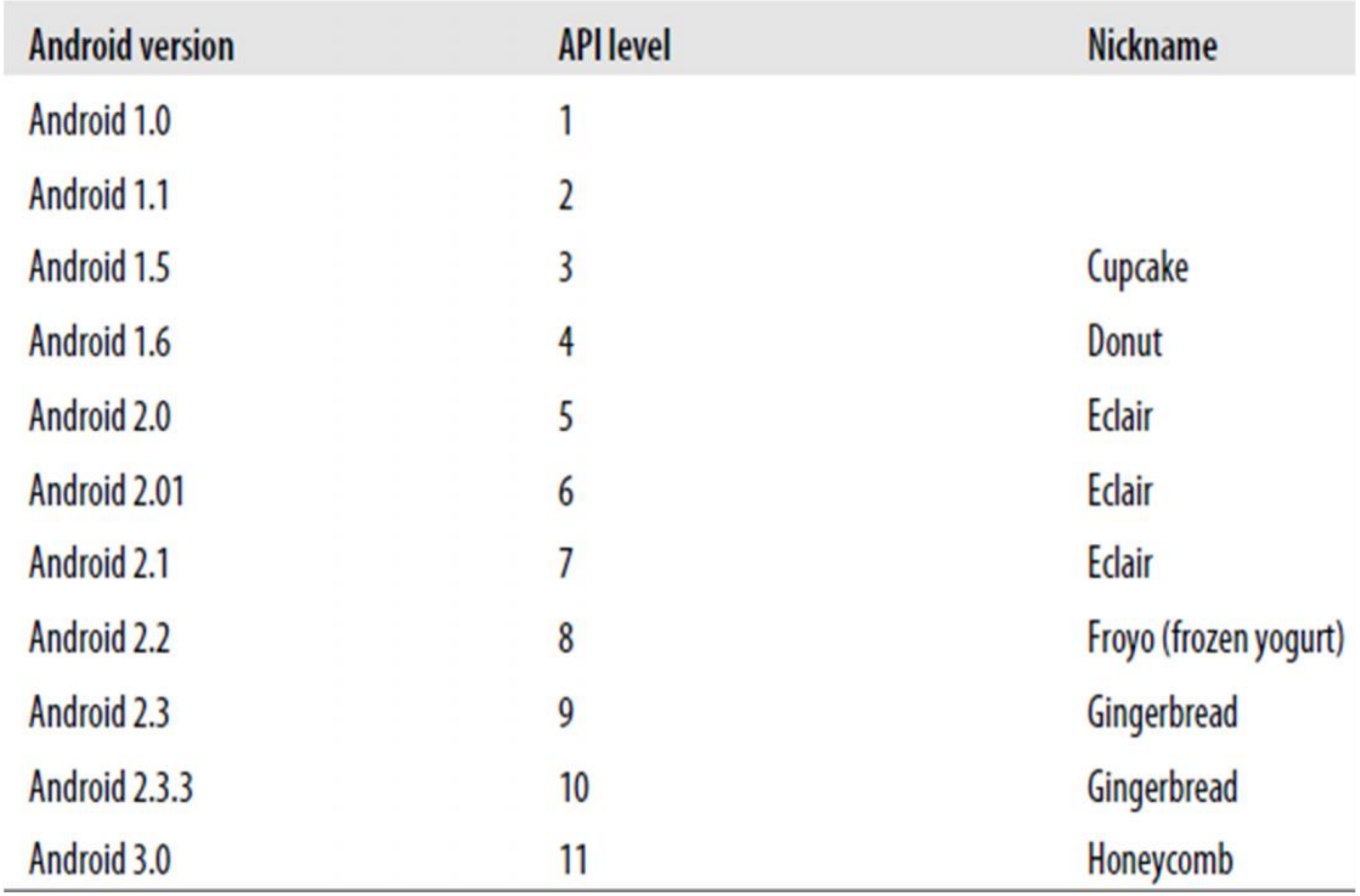
Samsung Moment

Motorola Droid

HTC Tattoo

nexus one

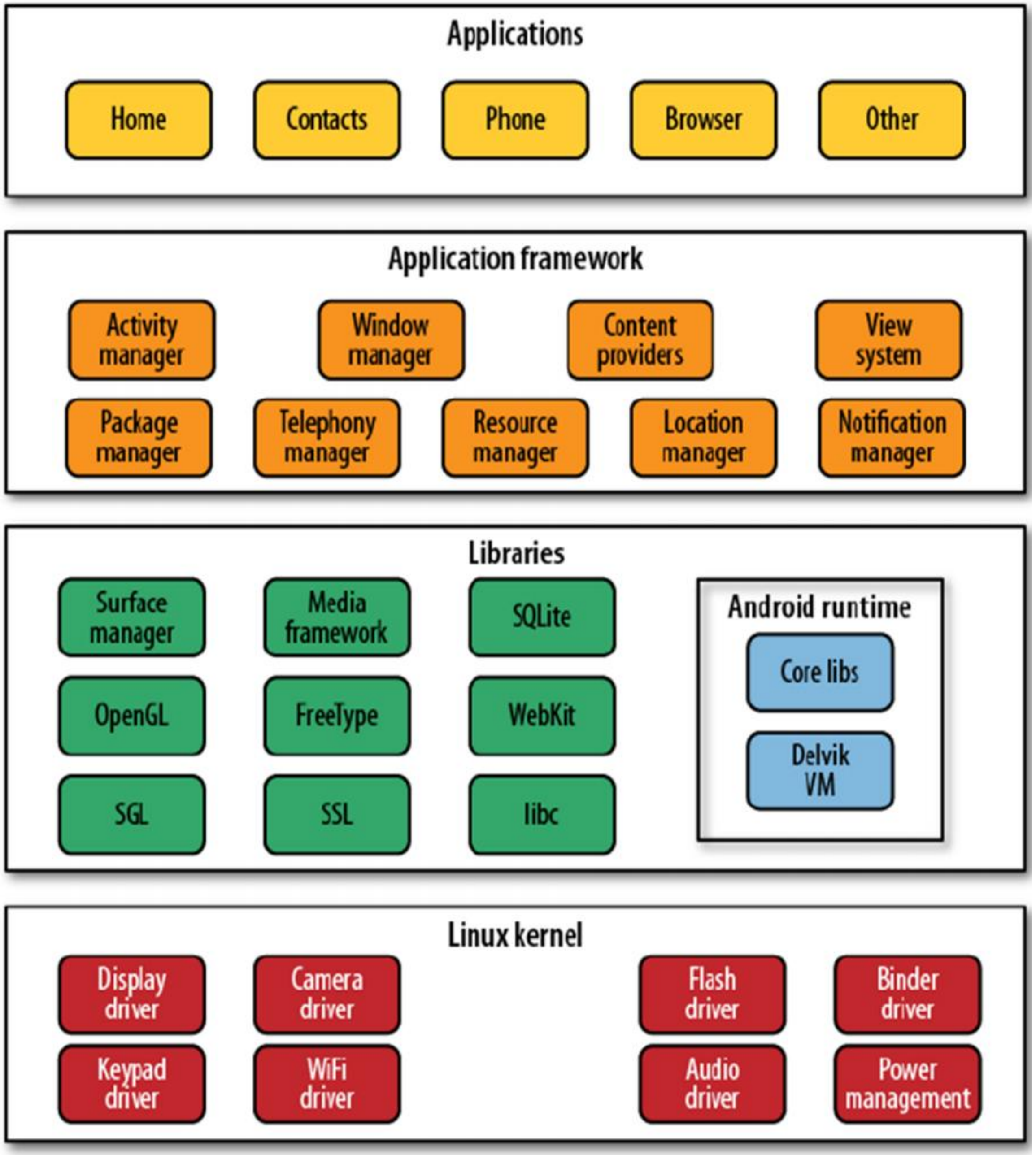
Android Versions



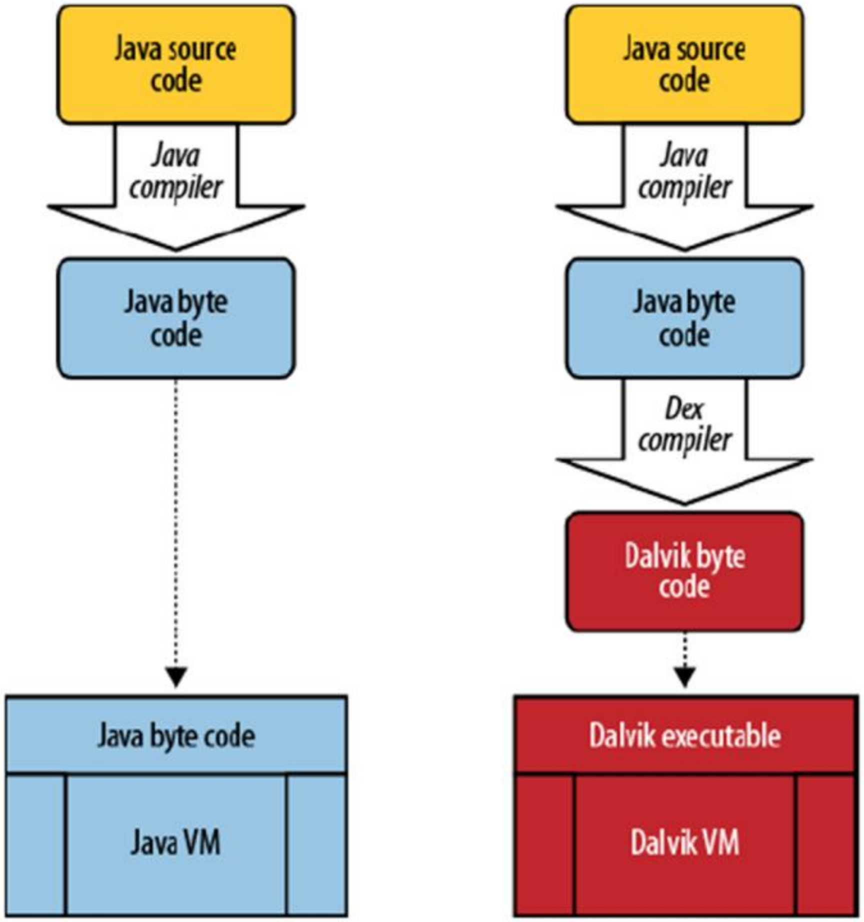
Android version	API level	Nickname
Android 1.0	1	
Android 1.1	2	
Android 1.5	3	Cupcake
Android 1.6	4	Donut
Android 2.0	5	Eclair
Android 2.01	6	Eclair
Android 2.1	7	Eclair
Android 2.2	8	Froyo (frozen yogurt)
Android 2.3	9	Gingerbread
Android 2.3.3	10	Gingerbread
Android 3.0	11	Honeycomb

Android stack

- *Webkit* - A fast web-rendering engine used by Safari, Chrome, and other browsers
- *SQLite* - A full-featured SQL database
- *OpenGL* - 3D graphics libraries
- *OpenSSL* - The secure locket layer
- *Delvik* – A virtual machine designed specifically for Android
- *Apache Harmony* - An open source implementation of Java



Android and Java



Android Features

- Linux OS kernel
- Java programming
- Open source libraries: SQLite, OpenSSL,WebKit, OpenGL
- A simple and powerful SDK
- No licensing, distribution, or development fees
- Development over many platform
  - Linux, Mac OS, windows
- Excellent documentation

Android Application dev’t

- Java
- Android SDK
- XML
- Android VM

Smart card operating system

- The smallest operating system runs on smart card.
- Contains CPU chip.
- Processing power and memory constraints.
- They handle a single function like electronic payment
- Some of them are java oriented.

The Trends in OS Technology

- Mainframe computing (60's-70's)

– massive computers to execute big data processing applications

– very few computers in the world

- Desktop computing (80's-90's)

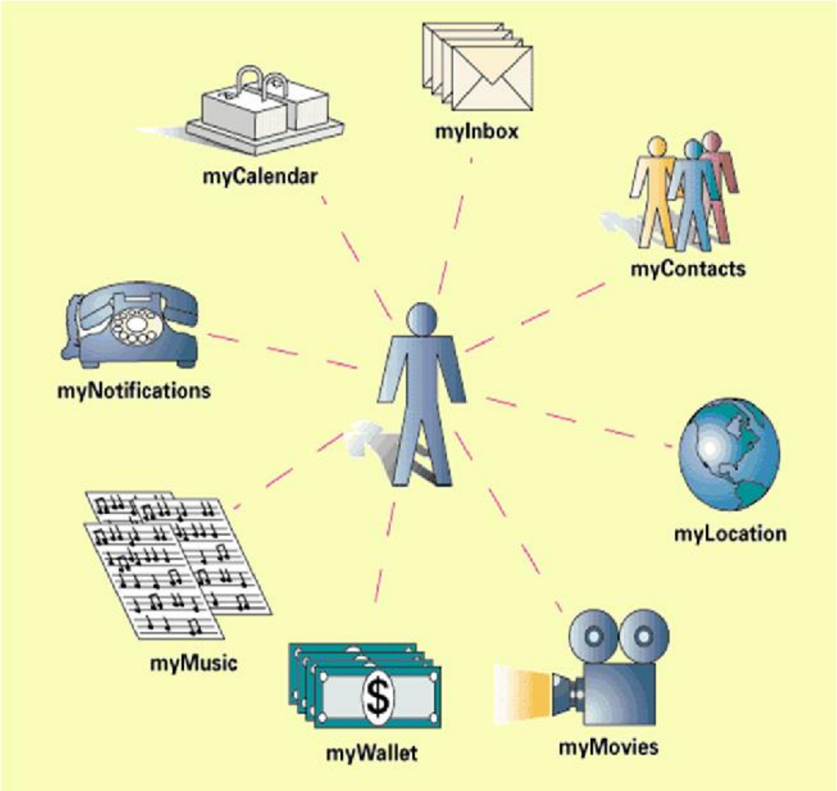
– one computer at every desk to help in business related activities

– computers connected in intranets to a massive global network (internet), all wired

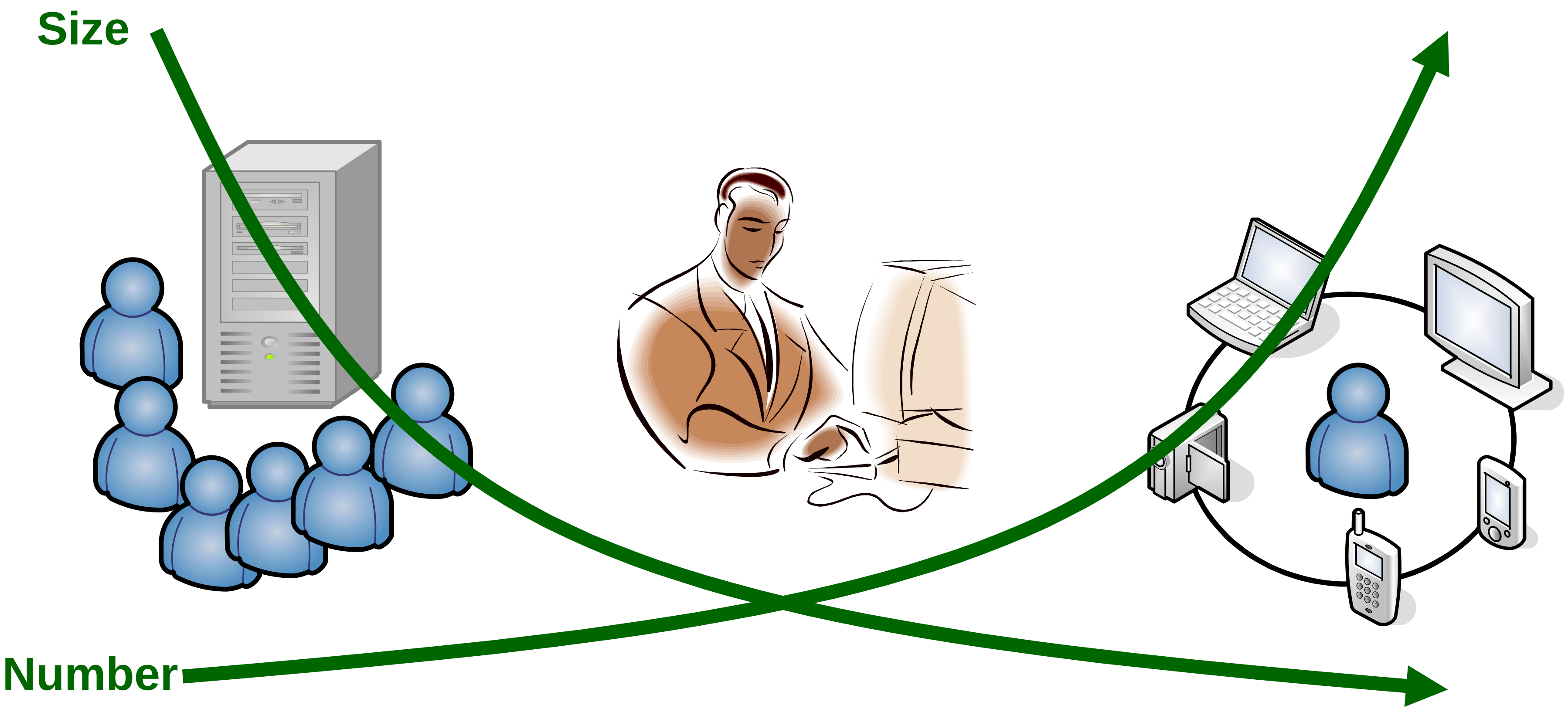
- Ubiquitous computing (00's?)

– tens/hundreds of computing devices in every room/person,

- becoming "invisible" and part of the environment



OS computing: Trend



One Computer for Many People

(Mainframe Computing)

One Computer for One Person

(PC Computing)

Many Computers for One Person

(Ubiquitous/Pervasive Computing)