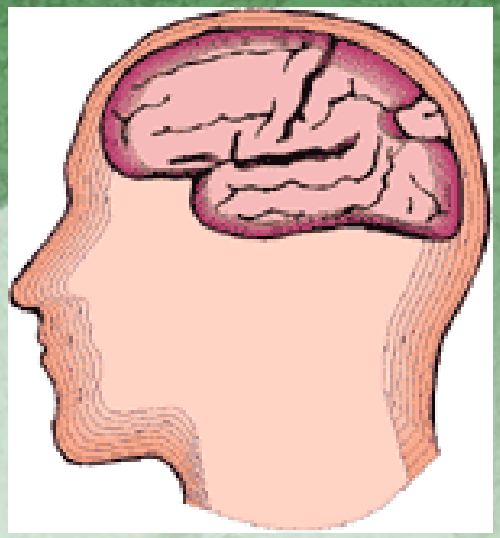
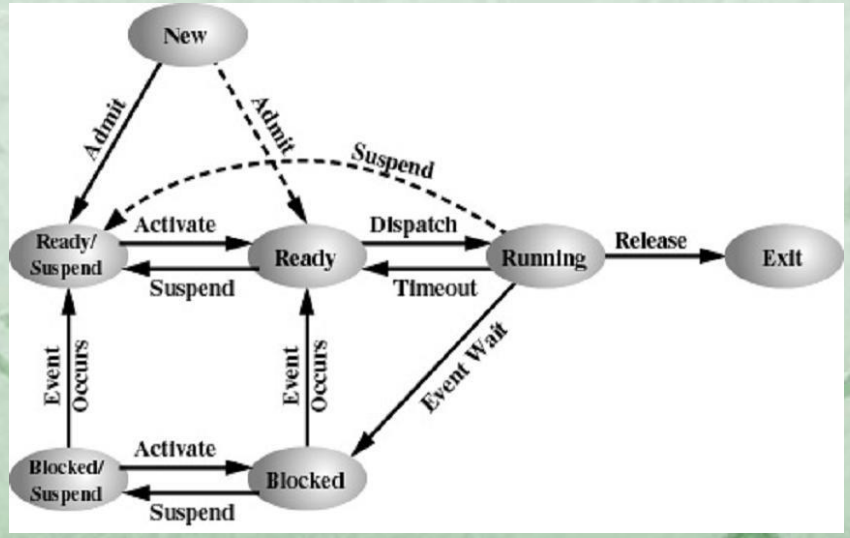
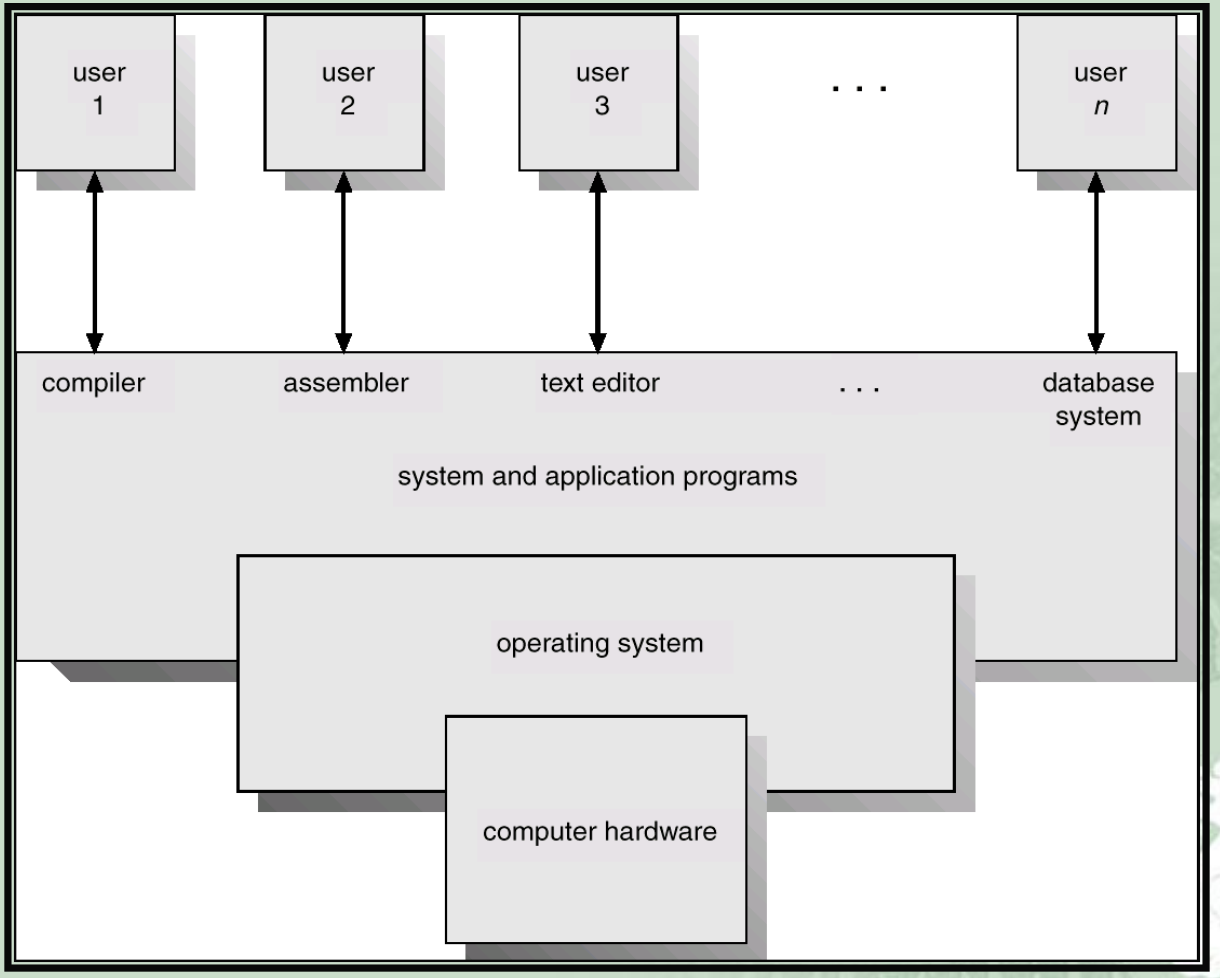


Operating Systems



Abstract View of System Components



Operating Systems

- What is an operating system?
  - Hard to define precisely
  - How about…
    - "Software that makes computing power available to users by controlling the hardware."
    - "Software executes when nothing else is happening."
    - "A collection of software modules including device drivers, libraries, and access routines."

What does a modern operating system do?

- **Provides Abstractions:**
  - Hardware has low-level physical resources with complicated, special-purpose interfaces.
  - OS provides abstractions that present clean interfaces.
  - Goal: make computer easier to use.
  - Examples: Processes, Unbounded Memory, Communication Mechanisms.
- **Provides Standard Interface:**
  - Goal: portability.
  - Example: Unix runs on many very different computer systems.

What does a modern operating system do?

- **Manages Resource Usage:**
  - Goal: allow multiple users to share resources fairly, efficiently, safely and securely.
  - Examples:
    - Multiple processes share one processor.
    - Multiple processes share multi-core processors.
    - Multiple programs share one physical memory.
    - Multiple users and files share one disk.
    - Multiple programs share a given amount of disk and network bandwidth.
- **Consumes Resources:**
  - Solaris takes up about 8 Mbytes physical memory.
  - Android has about 12 million lines of code
  - Windows 10 has about 50 million lines of code.

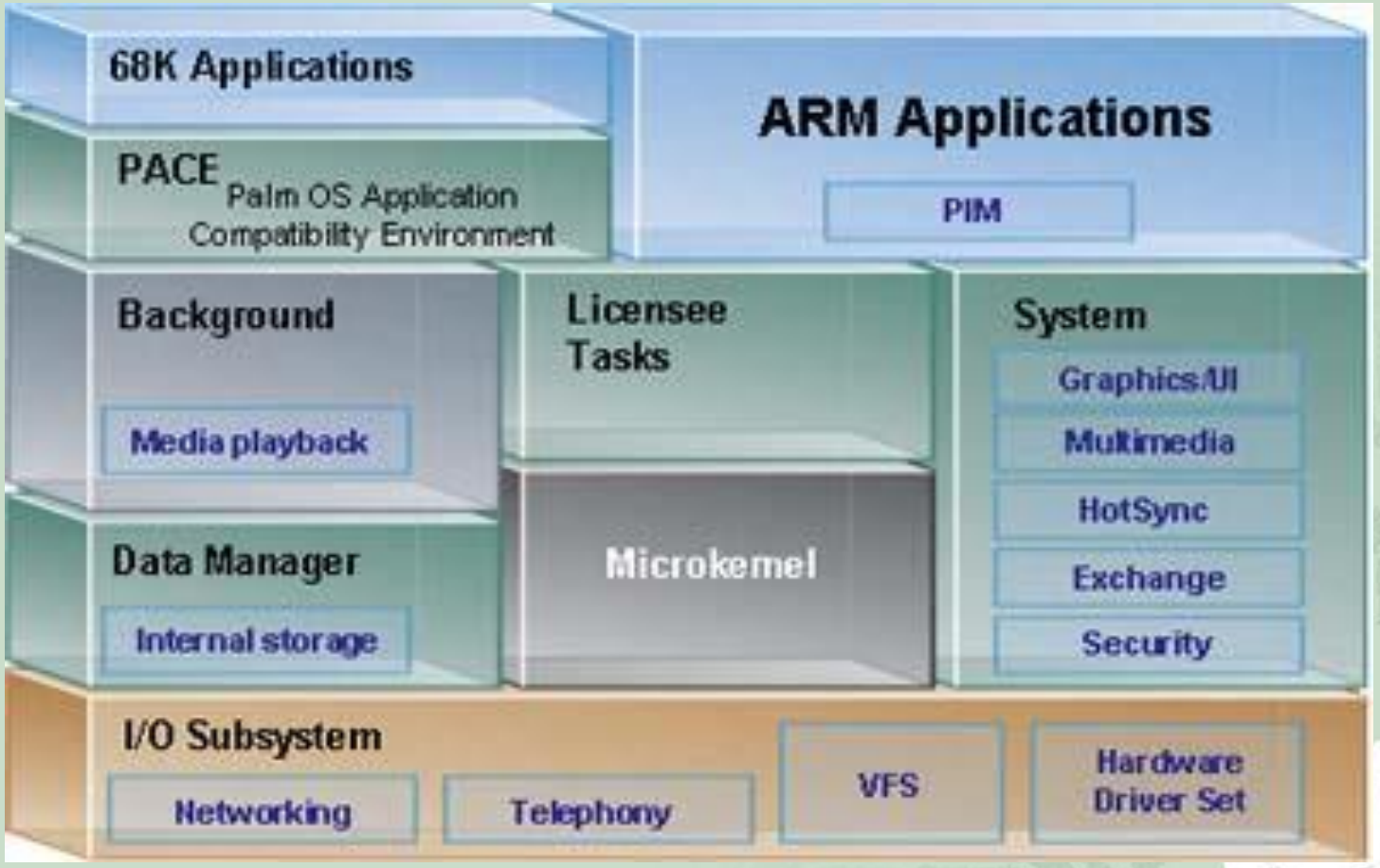
Where are OS's Used?

- In more and more places!
  - Desktops and Server Computers
  - Laptops and Tablets
  - Smart phones
  - Cars
  - Game boxes
  - Smart appliances
  - ….

Example OS: PalmOS

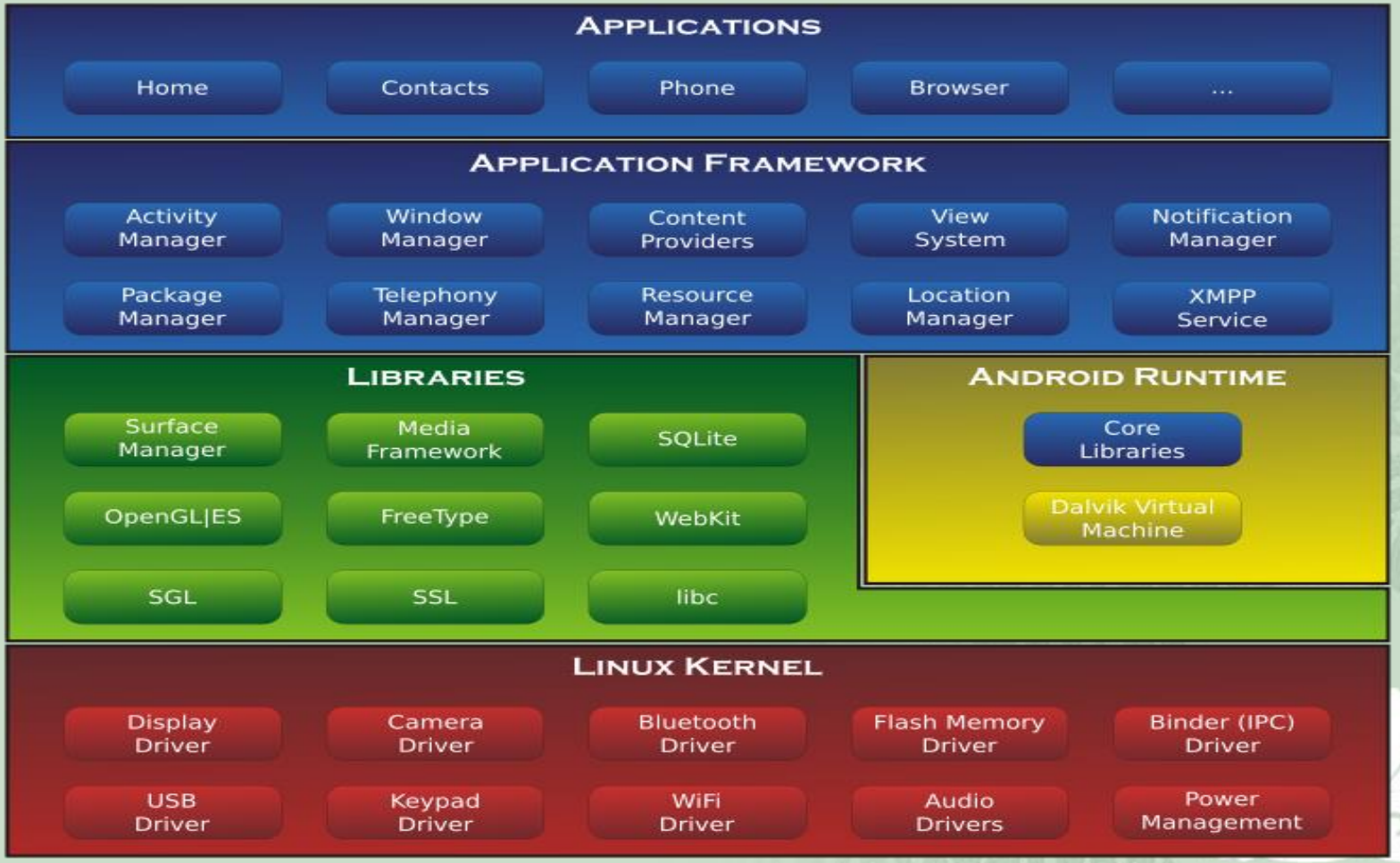
- Used for PalmPilot PDAs and successors
- Multitasking since PalmOS 5

Example OS: PalmOS

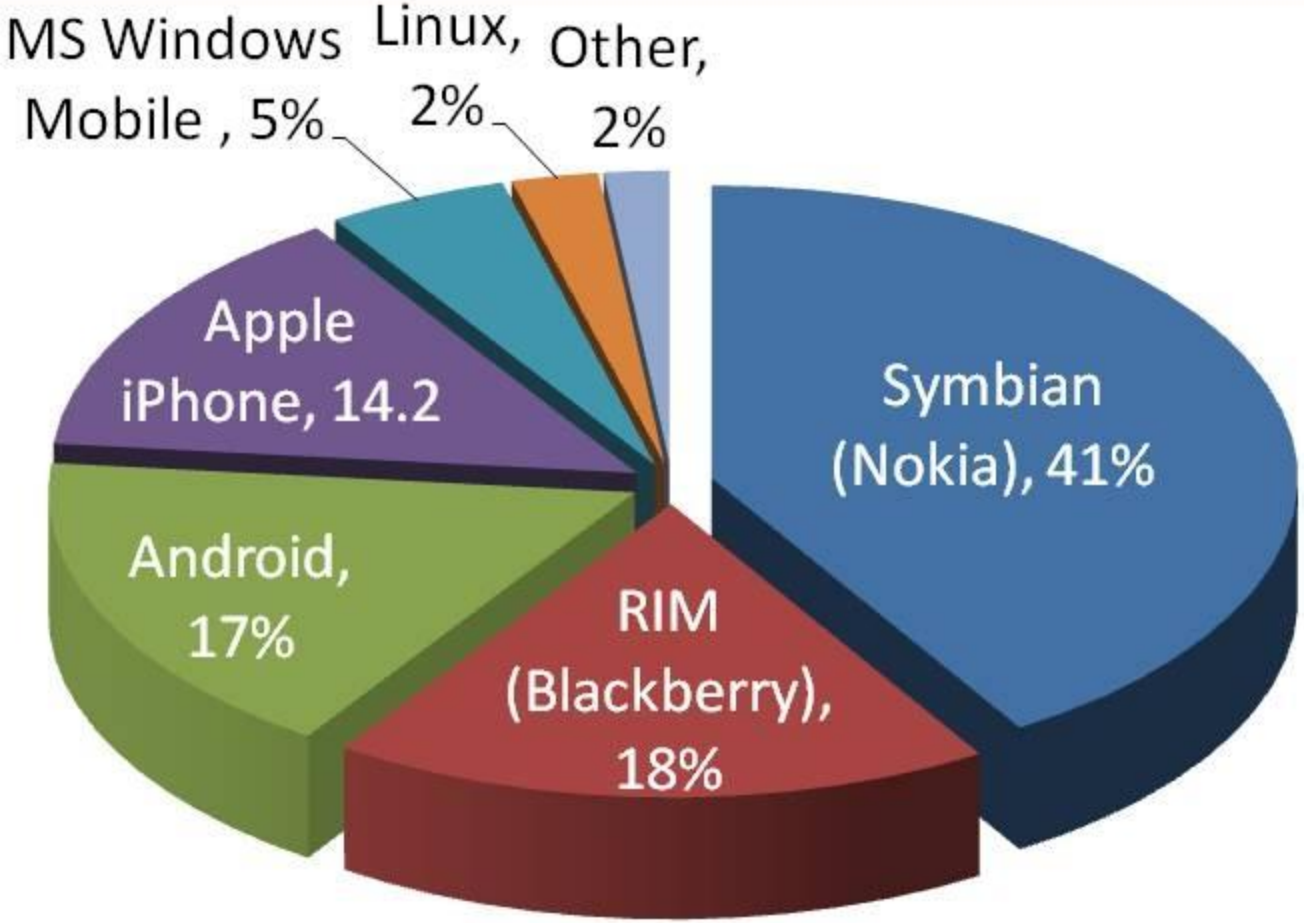


Example OS: Android ™

Android-System-Architecture



Global Smartphone Market Share by Operating System



Source: *Financial Times*, August 13, 2010

Global Smartphone Market Share by Operating System

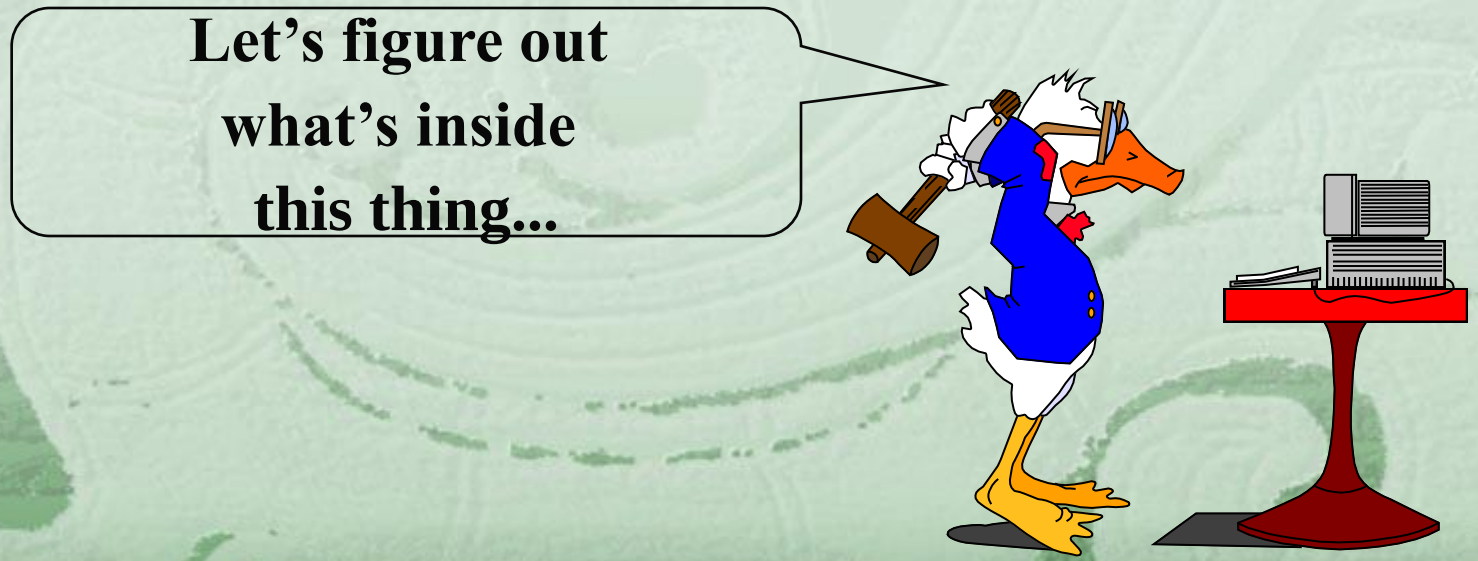
Source: IDC, Nov 2016

Period	Android	iOS	Windows Phone	Others
2015Q4	79.6%	18.7%	1.2%	0.5%
2016Q1	83.5%	15.4%	0.8%	0.4%
2016Q2	87.6%	11.7%	0.4%	0.3%
2016Q3	86.8%	12.5%	0.3%	0.4%

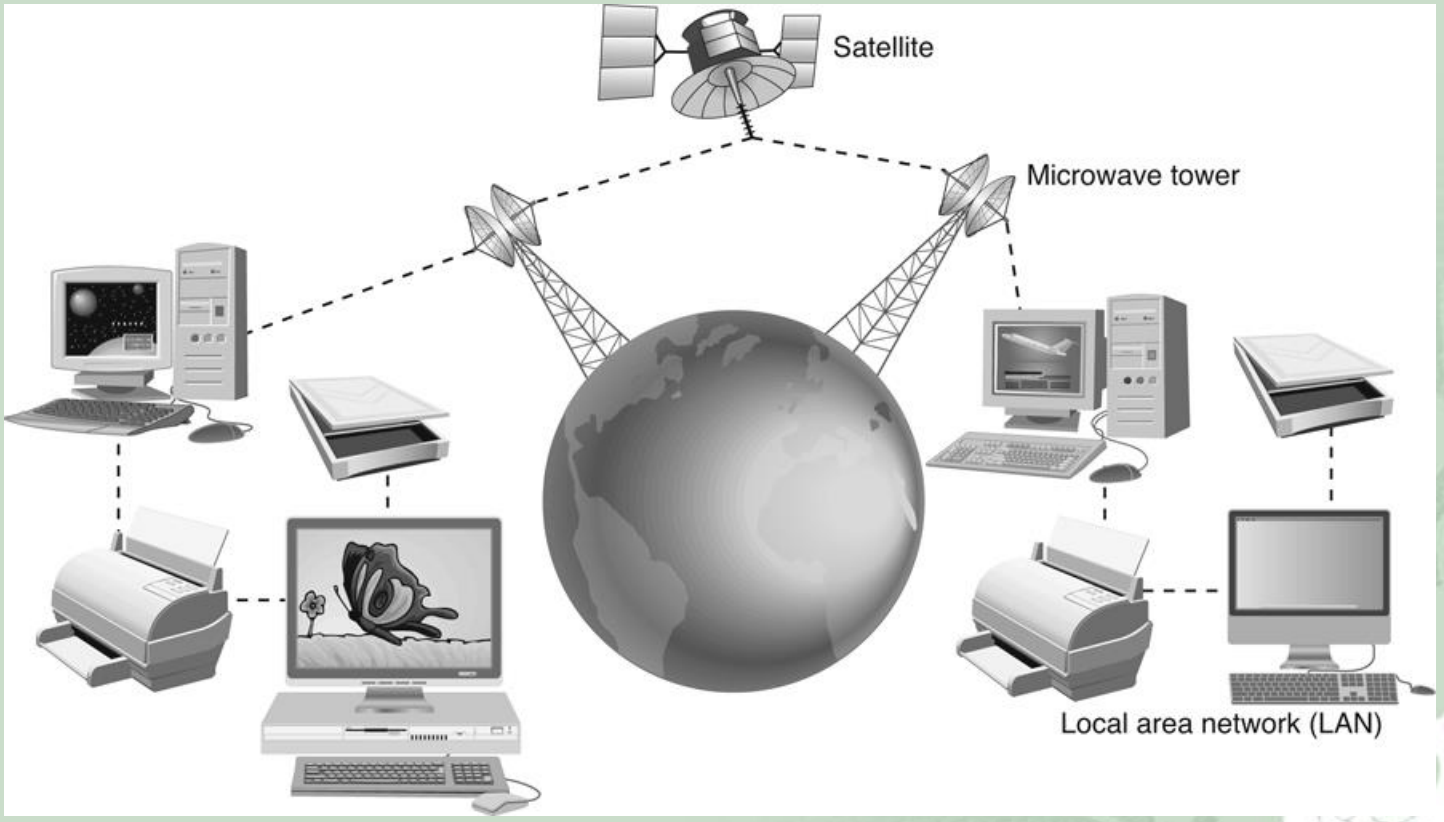
Finally

- Operating systems are so large no one person understands whole system. Outlives any of its original builders.
- The major problem facing computer science today is how to build large, reliable software systems.
- Operating systems are one of very few examples of existing large software systems, and by studying operating systems we may learn lessons applicable to the construction of larger systems.

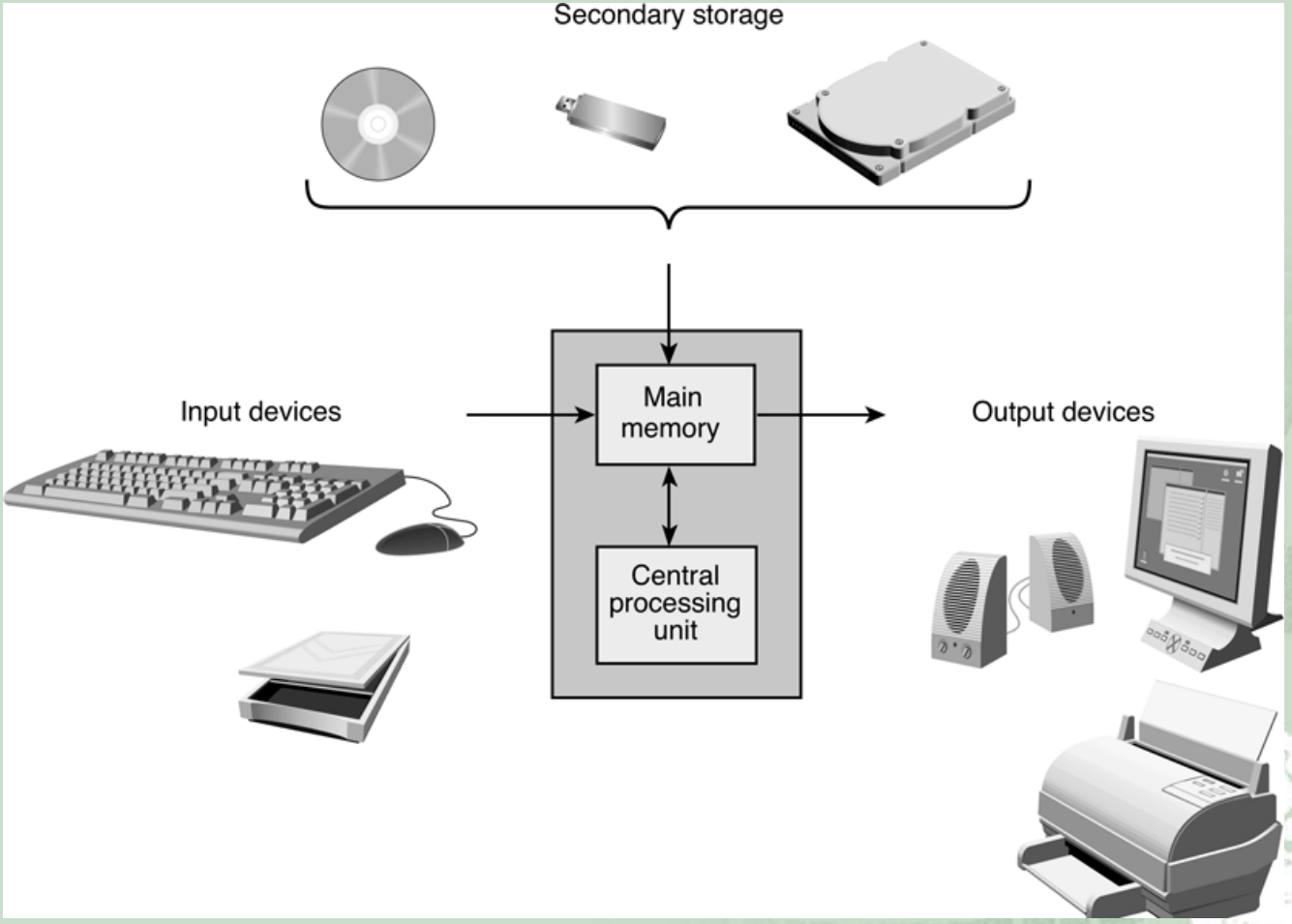
Computer System Overview



Network



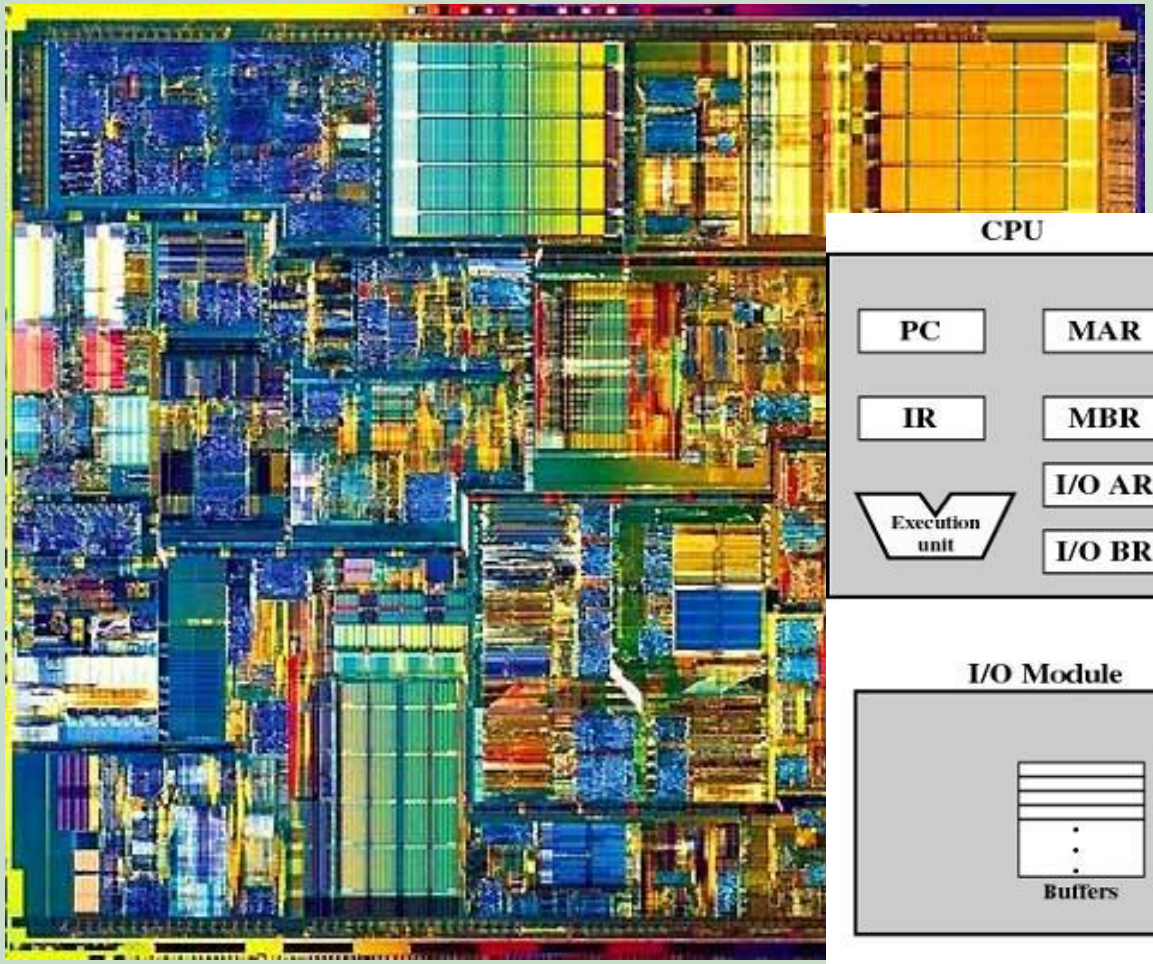
Components of a Computer

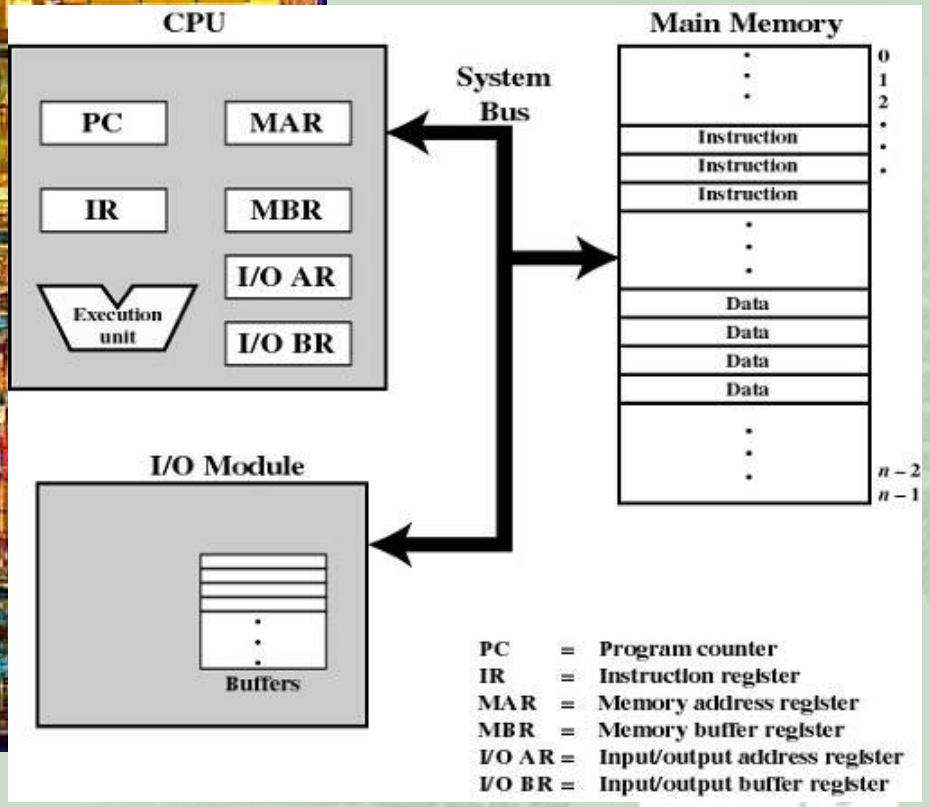


Computer Systems

- Registers
- Interrupts
- Caching
- Input/Output
- Protection
- Summary

CPU



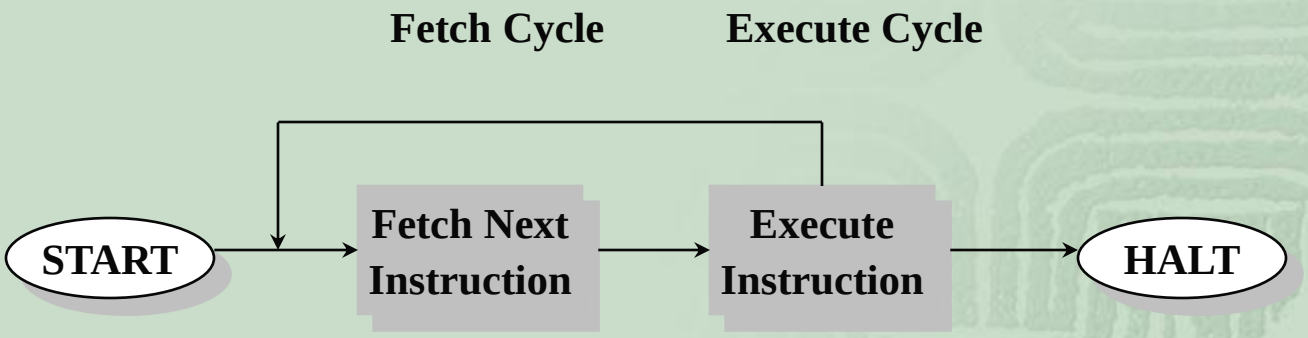

CPU
PC
MAR
IR
MBR
I/O AR
Execution unit
I/O BR
System Bus
Main Memory
0
1
2
Instruction
Instruction
Instruction
Data
Data
Data
Data
n – 2
n – 1
I/O Module
Buffers
PC = Program counter
IR = Instruction register
MAR = Memory address register
MBR = Memory buffer register
I/O AR = Input/output address register
I/O BR = Input/output buffer register

Processor Registers

- User-visible registers
  - May be referenced by machine language
  - Available to all programs - application programs and system programs
    - Data Registers – can be changed by user
    - Address Registers – could be separate from data register
    - Stack Registers – user / supervisor stacks
    - Condition Codes – results of operations
- Control and status registers
  - May or may not be visible
    - Program Counter (PC) – address of next instruction
    - Instruction Register (IR) – most recently fetched instruction
    - MAR/MBR – memory reference registers
    - Program Status Word (PSW) – condition codes, interrupts, mode

Instruction Execution

- Processor executes instructions in a program
- Instructions are fetched from memory on at a time



Computer Systems

- Registers
- Interrupts
- Caching
- Input/Output
- Protection
- Summary

Interrupts

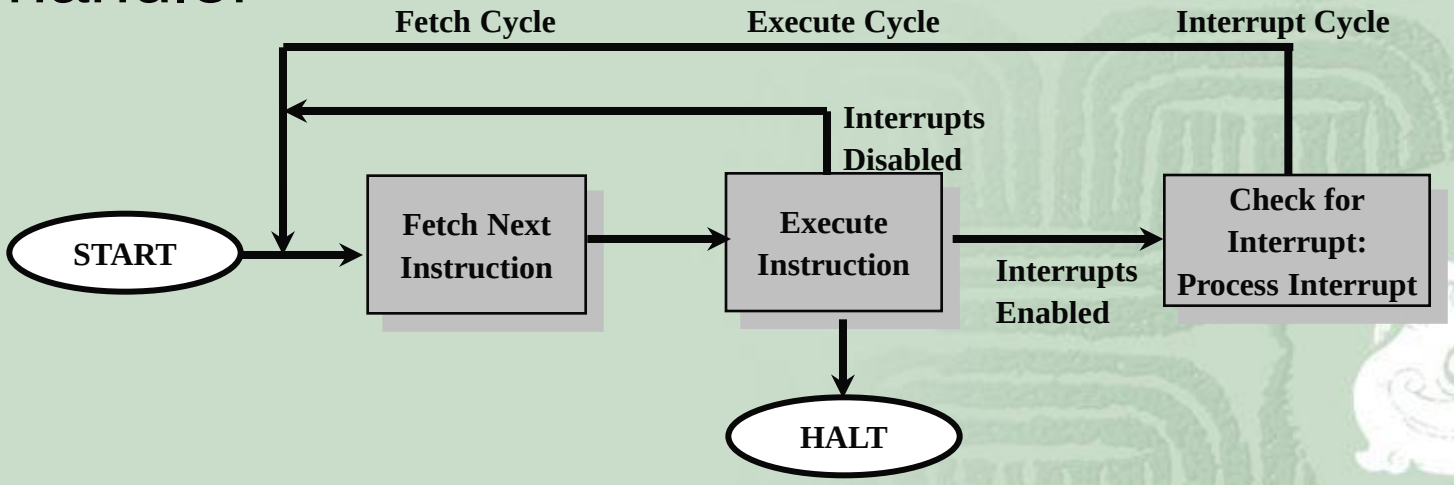
- The interrupt was the principle tool available to system programmers in developing multi-tasking systems!
- Improves processing efficiency by allowing the processor to execute other instructions while an I/O operation is in progress
- A suspension of a process caused by an event external to that process and performed in such a way that the process can be resumed

Processing of Interrupts

- Classes of Interrupts
  - Program
    - arithmetic overflow
    - division by zero
    - execute illegal instruction
    - reference outside user's memory space
  - Timer
  - I/O
  - Hardware failure
- An interrupt handler determines nature of the interrupt and performs whatever actions are needed
  - Control is transferred to this program
  - Generally part of the operating system

Interrupt Cycle

- Processor checks for interrupts
- If no interrupts fetch the next instruction for the current program
- If an interrupt is pending, suspend execution of the current program, and execute the interrupt handler



Simple Interrupt Processing