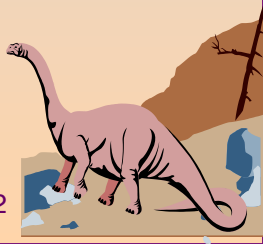
CSC 345

Operating System

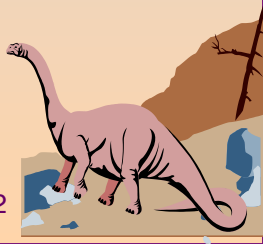
Box Leangsuksun

www.latech.edu/~box



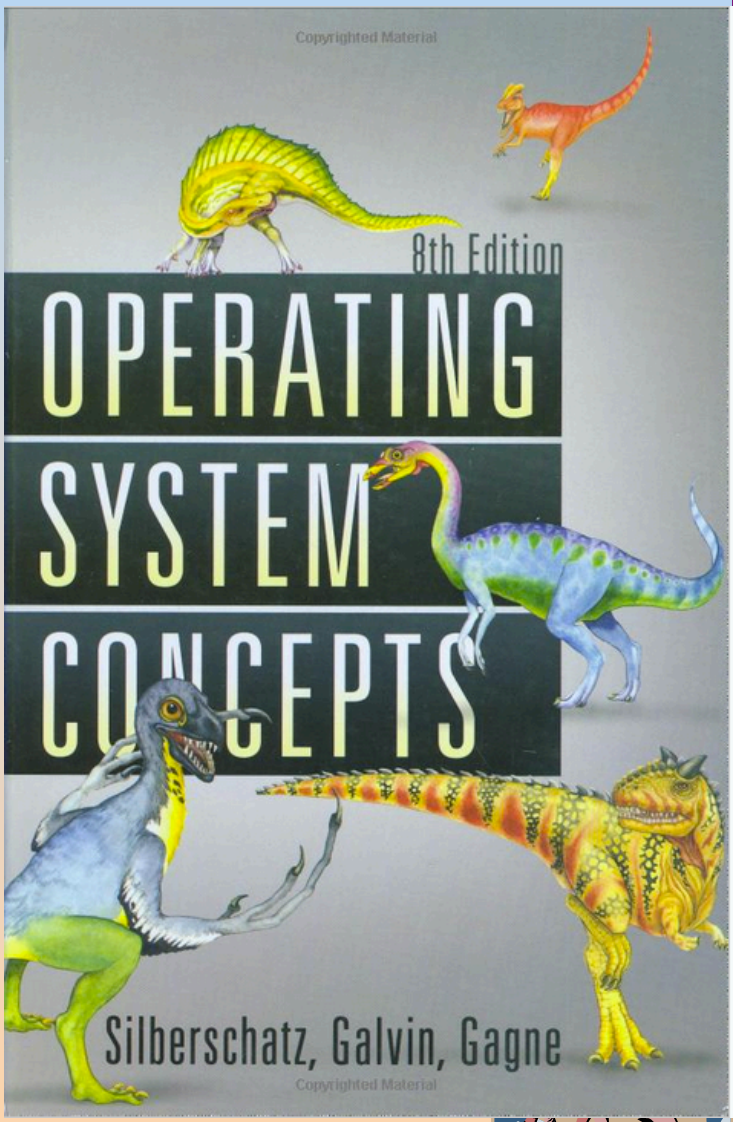
Class Info

- Office Hours: 1-3pm M-T-W-T
- Dr. Box's & Contact Info:
  - www.latech.edu/~box
  - Phone 318-257-3291
  - Email: box@latech.edu or naibox@gmail.com
- Class room: NH 120
  - T-TH 8-9:5am



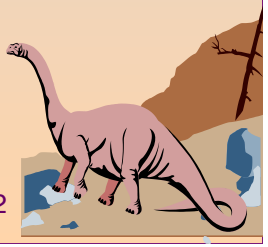
Class Info

- Main Text:
  - Silberschatz, Abraham, and Peter Galvin, Operating System Concepts, 7th or 8th or 9th Ed., John Wiley & Sons, ISBN 0-471-69466-5



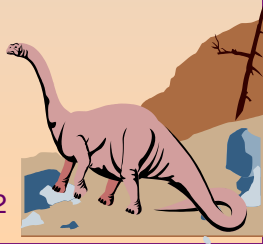
Class Info

- Objective:
  - introductory course to the theory, design and implementation of operating systems
  - OS is the software layer between user programs and the computer hardware
  - abstractions (processes, file system, etc.) of the underlying hardware



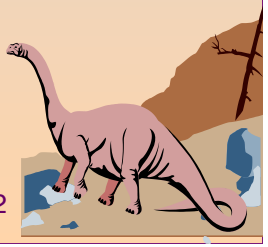
Grading

- 3 Programming assignments 30%
- 2-5 Homework assignments 15%
- Term Paper 10%
- Midterm exam 20%
- Final exam 25%



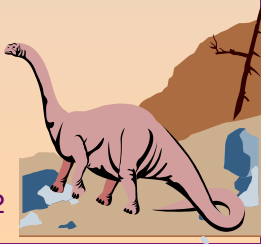
Important Dates

- April 11, 2013: Midterm exam
- April 23, 2013: Term Paper
- May 9, 2013 : **Final exam**



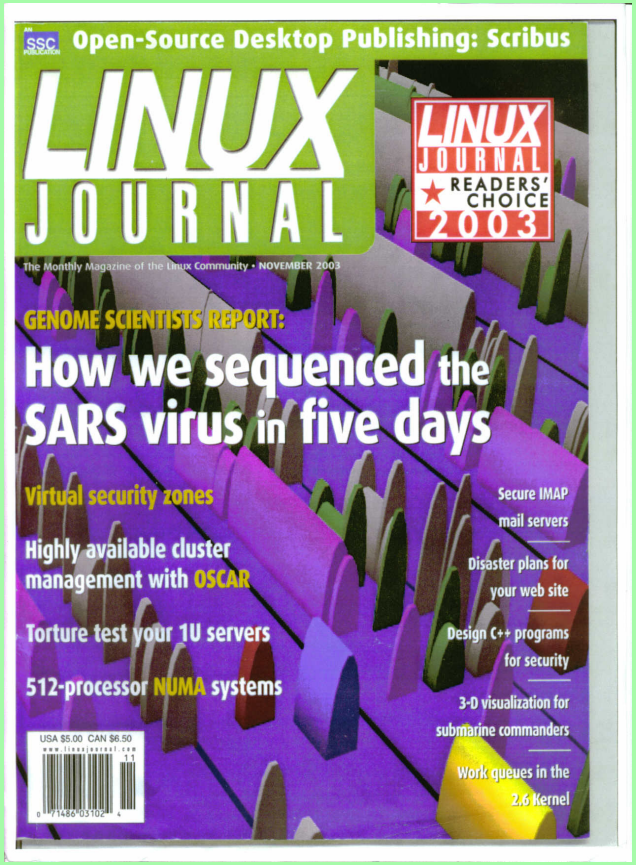
Box's 1 minute Bio

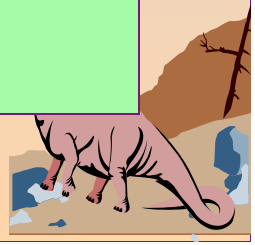
- **PhD in CS (1995):**
  - PhD Thesis: Resource management/allocation in Heterogeneous Parallel Distributed Computing
- **7 years in industry labs (Bell-Labs, Lucent Technologies)**
  - Highly Reliable Software/system (IN, Service Management)
  - Architect, PM, Tech lead (15-30 team size)
  - R&D -> 4 major network management products
- **Associate Professor in CS since 2002.**
  - 15 graduate students (4 PhD)
- **Research Interest**
  - Cluster computing, Fault Tolerance OS/Runtime, Reliability , Availability and Serviceability and Security (RASS) in HPC/HEC, Software Engineering
- **Services**
  - IEEE Cluster Computing Program committee member 2004-2005
  - A founder and CO-Chair: High Availability and Performance Computing 2003-2004
  - 2003 Outstanding Teach Award, COES, Louisiana Tech U.
  - Creator of www.searchkatrina.org



Appeared in a front cover in two major Linux magazines, various technical papers, research exhibitions.
web site: http://xcr.cenit.latech.edu/ha-oscar

HA-OSCAR beta was released to open source community





Research Collaborators

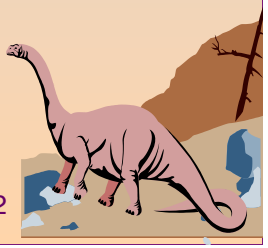
- **National, Academic and Industry La**

  - ✓ ORNL
  - ✓ Intel, Dell, Ericsson
  - ✓ Lucent, CRAY
  - ✓ IU, NCSA, OSU, NCSU, UNM, TTU, UIUC, ASU
  - ✓ Systran
  - ✓ Sandia, Ames

  - ✓ More on the way; US Army, NetQoS, MIT, IBM, Apple

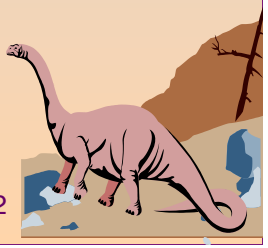
Chapter 1: Introduction

- What is an Operating System?
- Mainframe Systems
- Desktop Systems
- Multiprocessor Systems
- Distributed Systems
- Clustered System
- Real -Time Systems
- Handheld Systems
- Computing Environments



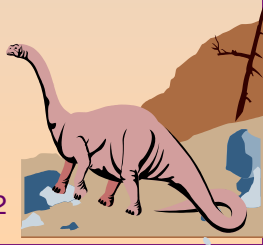
What is an Operating System?

- A program that acts as an intermediary between a user of a computer and the computer hardware.
- Operating system goals:
  - Execute user programs and make solving user problems easier.
  - Make the computer system convenient to use.
- Use the computer hardware in an efficient manner.

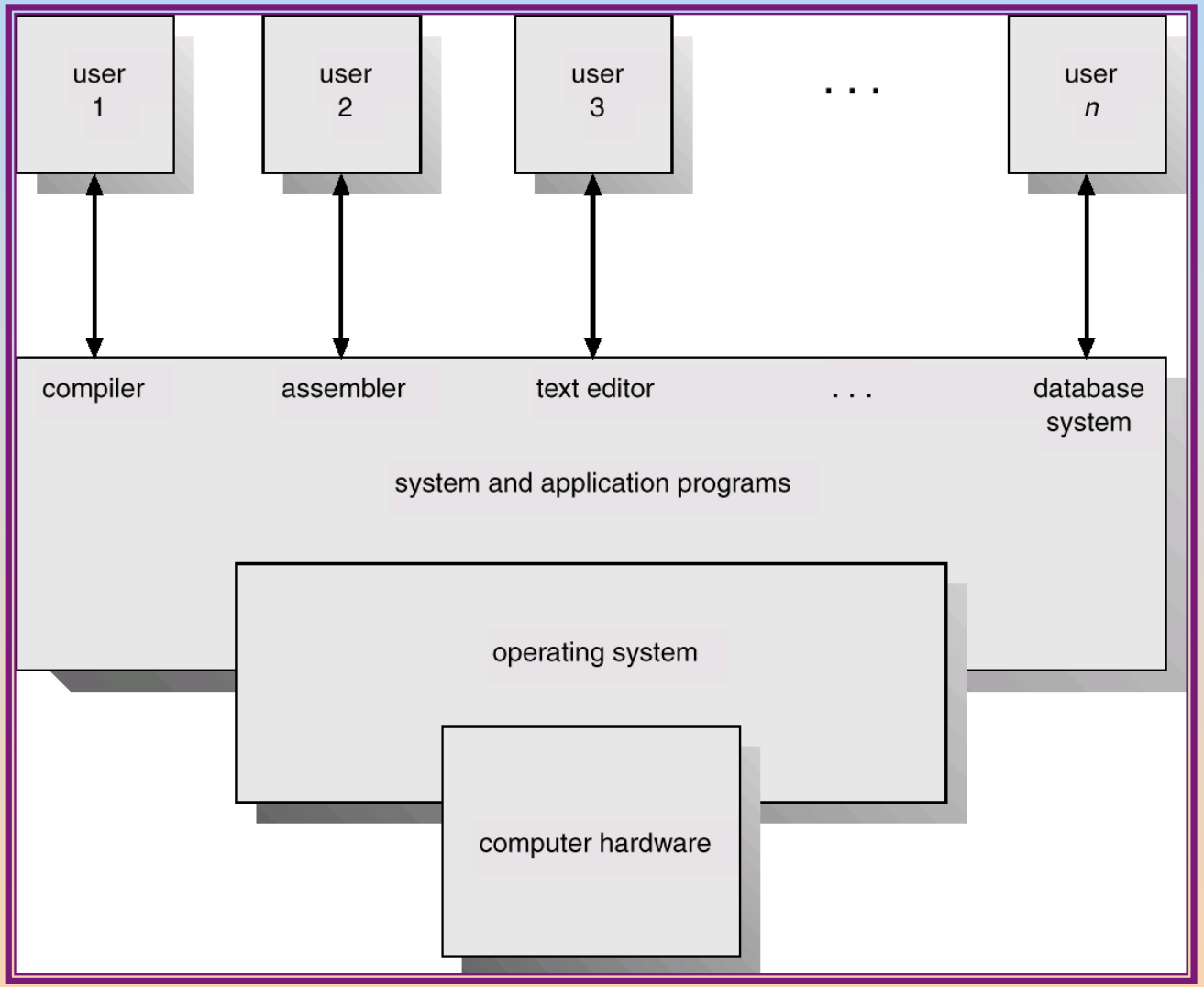


Computer System Components

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).

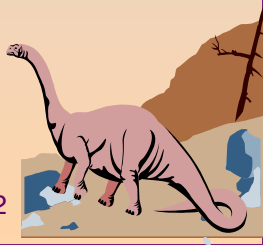


Abstract View of System Components



Operating System Definitions

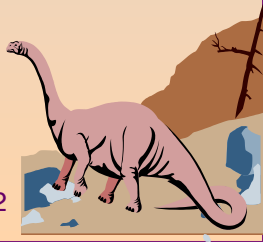
- Resource allocator – manages and allocates resources.
- Control program – controls the execution of user programs and operations of I/O devices .
- Kernel – the one program running at all times (all else being application programs).



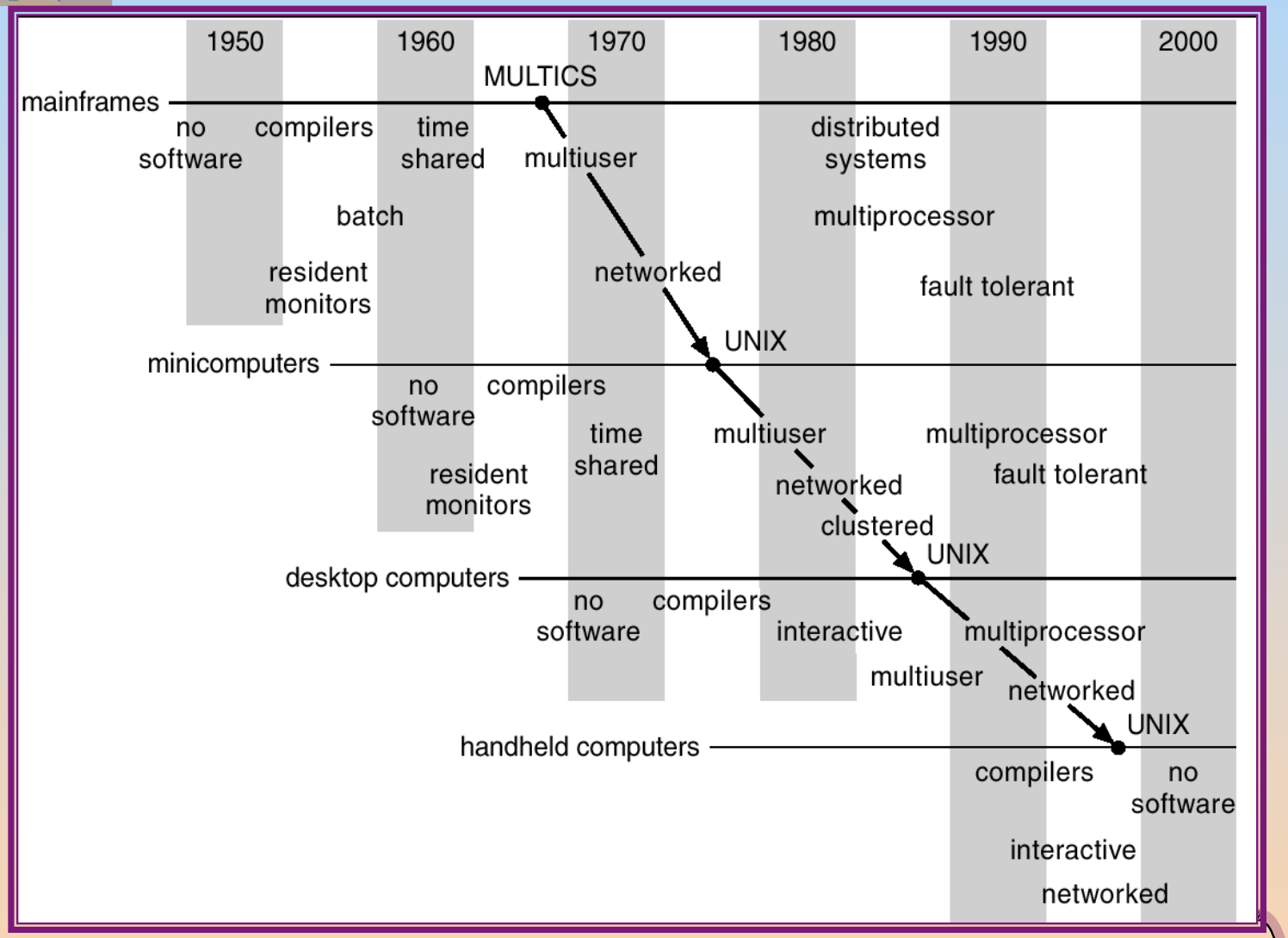
Hardware platforms & OS

- Question?
- What are hardware and OS you have used?

- Switch to intro to Linux

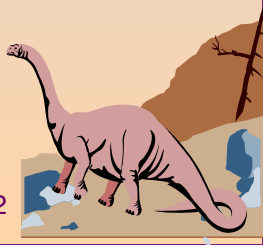


Migration of Operating-System Concepts and Features

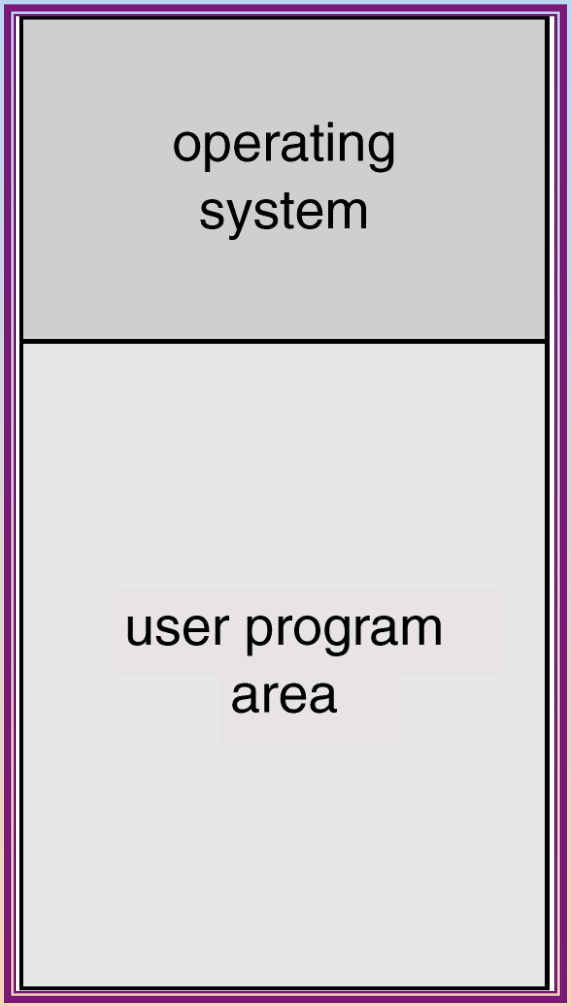


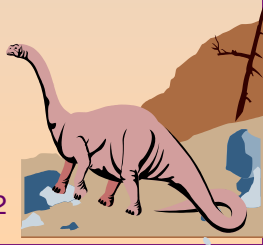
Mainframe Systems

- Reduce setup time by batching similar jobs
- Automatic job sequencing – automatically transfers control from one job to another. First rudimentary operating system.
- Resident monitor
  - initial control in monitor
  - control transfers to job
  - when job completes control transfers pack to monitor



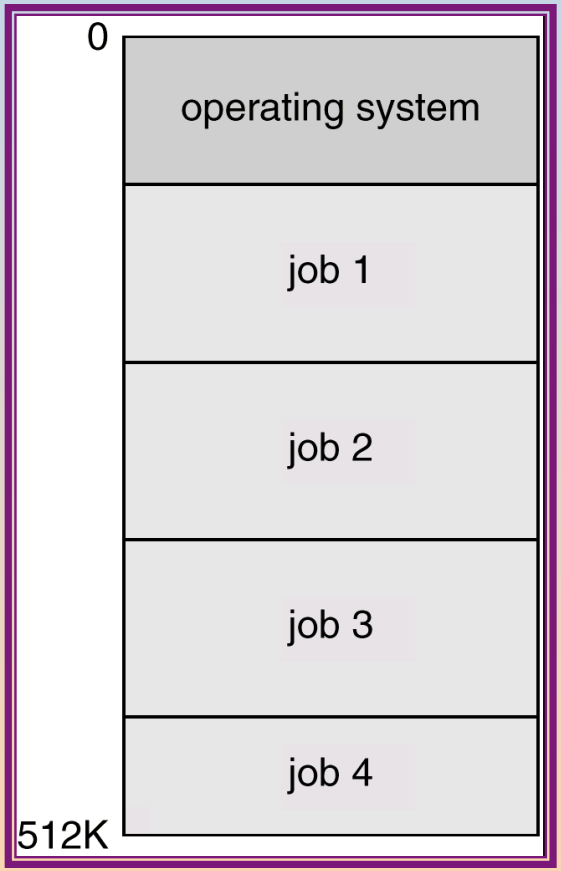
Memory Layout for a Simple Batch System

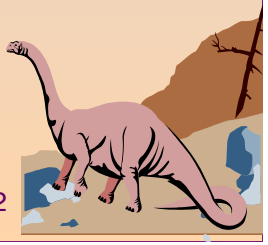




Multiprogrammed Batch Systems

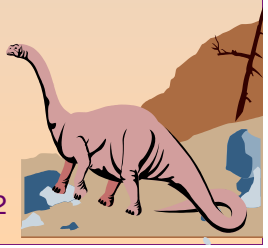
Several jobs are kept in main memory at the same time, and the CPU is multiplexed among them.





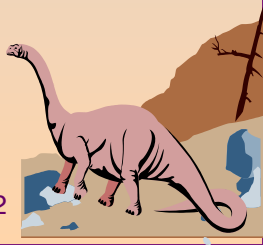
OS Features Needed for Multiprogramming

- 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 devices.



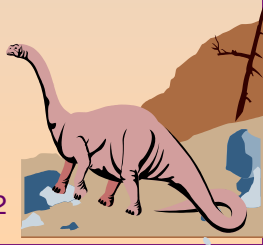
Time-Sharing Systems–Interactive Computing

- 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.



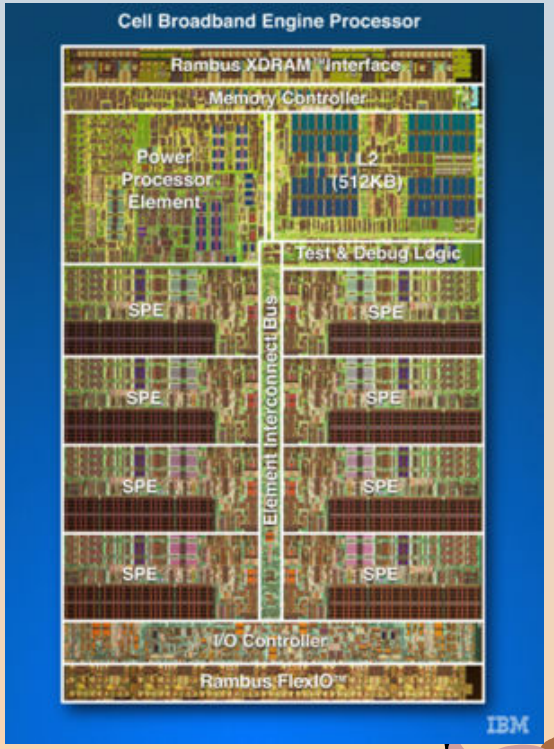
Desktop Systems

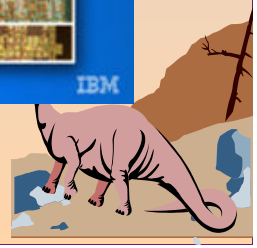
- *Personal computers* – computer system dedicated to a single user.
- I/O devices – keyboards, mice, display screens, small printers.
- User convenience and responsiveness.
- Can adopt technology developed for larger operating system’ often individuals have sole use of computer and do not need advanced CPU utilization of protection features.
- May run several different types of operating systems (Windows, MacOS, UNIX, Linux)



New computers with multicore architecture

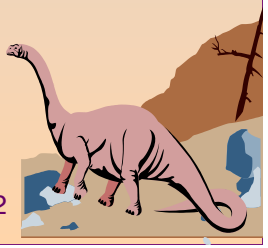
- more Hz is more speed – improve application runtime
- Issues in the past were thermo wall – heat and power consumption too high
- Solution trends
  - More than one CPUs in one die.
  - Intel & AMD mutlcore CPUs
  - Sony PS3





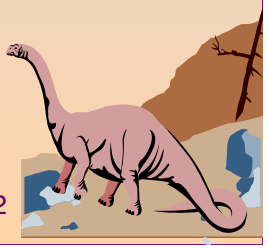
Parallel Systems

- Multiprocessor or multi-core systems with more than on CPU in relatively close communication.
- *Tightly coupled system* – processors share memory and a clock; communication usually takes place through the shared memory.
- Loosely coupled system – cluster computing
- Advantages of parallel system:
  - Increased *throughput*
  - Economical
  - Increased reliability
    - ✓ graceful degradation
    - ✓ fail-soft systems

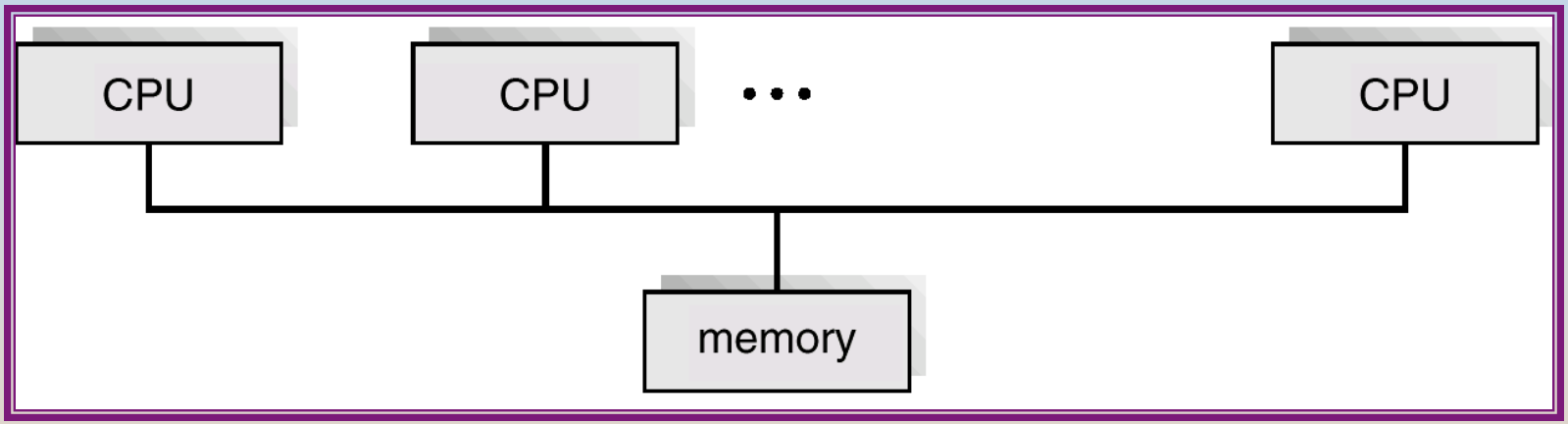


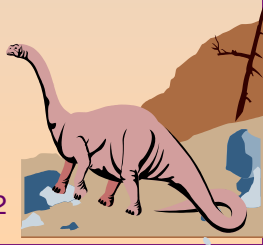
Parallel Systems (Cont.)

- *Symmetric multiprocessing (SMP)*
  - Each processor runs and identical copy of the operating system.
  - Many processes can run at once without performance deterioration.
  - Most modern operating systems support SMP
- *Asymmetric multiprocessing*
  - Each processor is assigned a specific task; master processor schedules and allocated work to slave processors.
  - More common in extremely large systems



Symmetric Multiprocessing Architecture





IBM Cell Processor

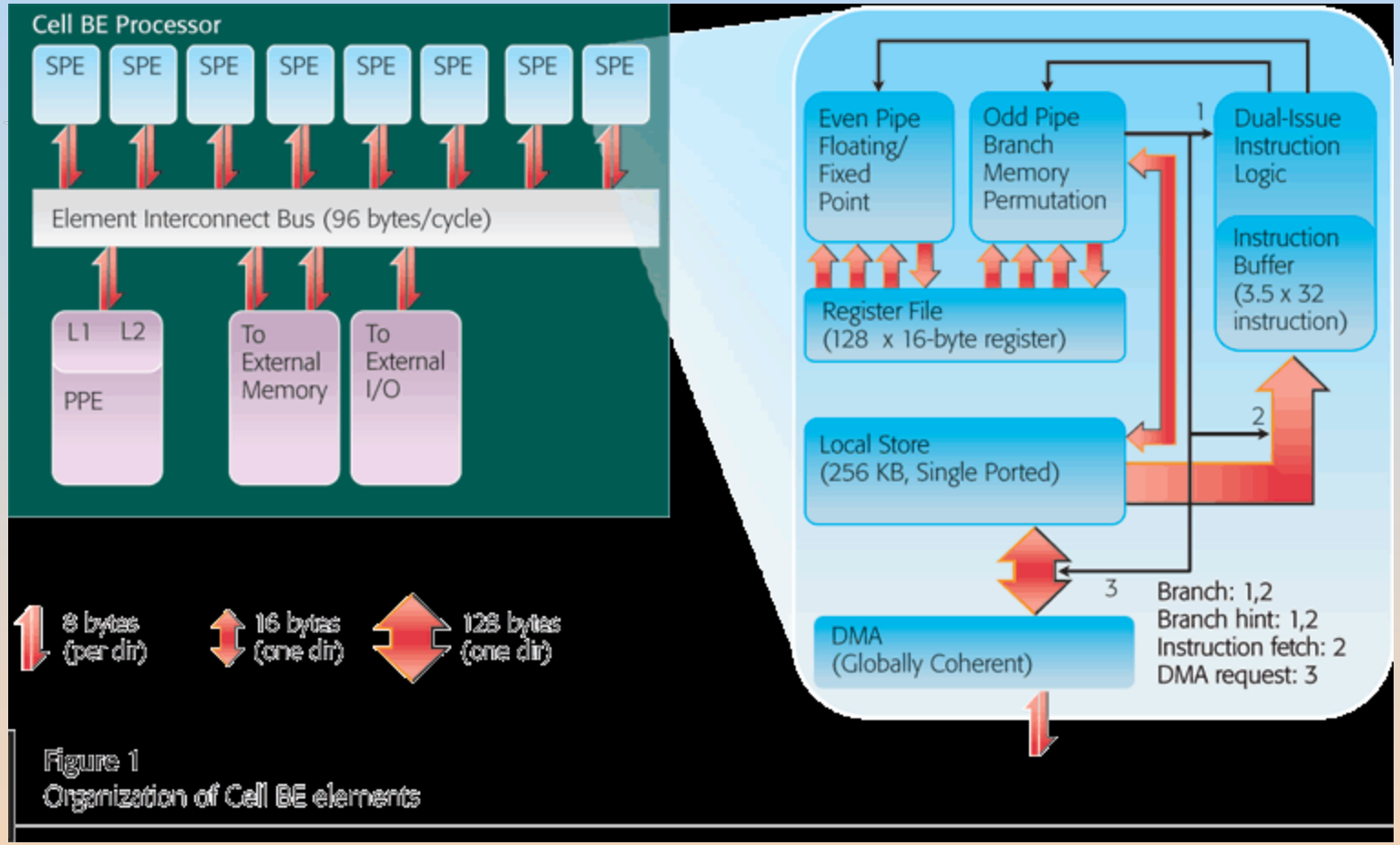
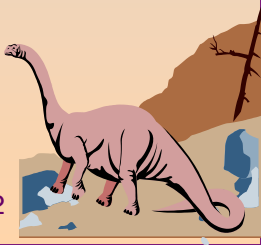
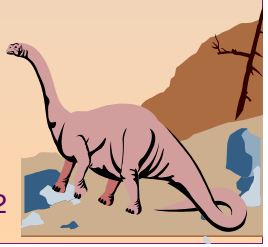


Figure 1
Organization of Cell BE elements

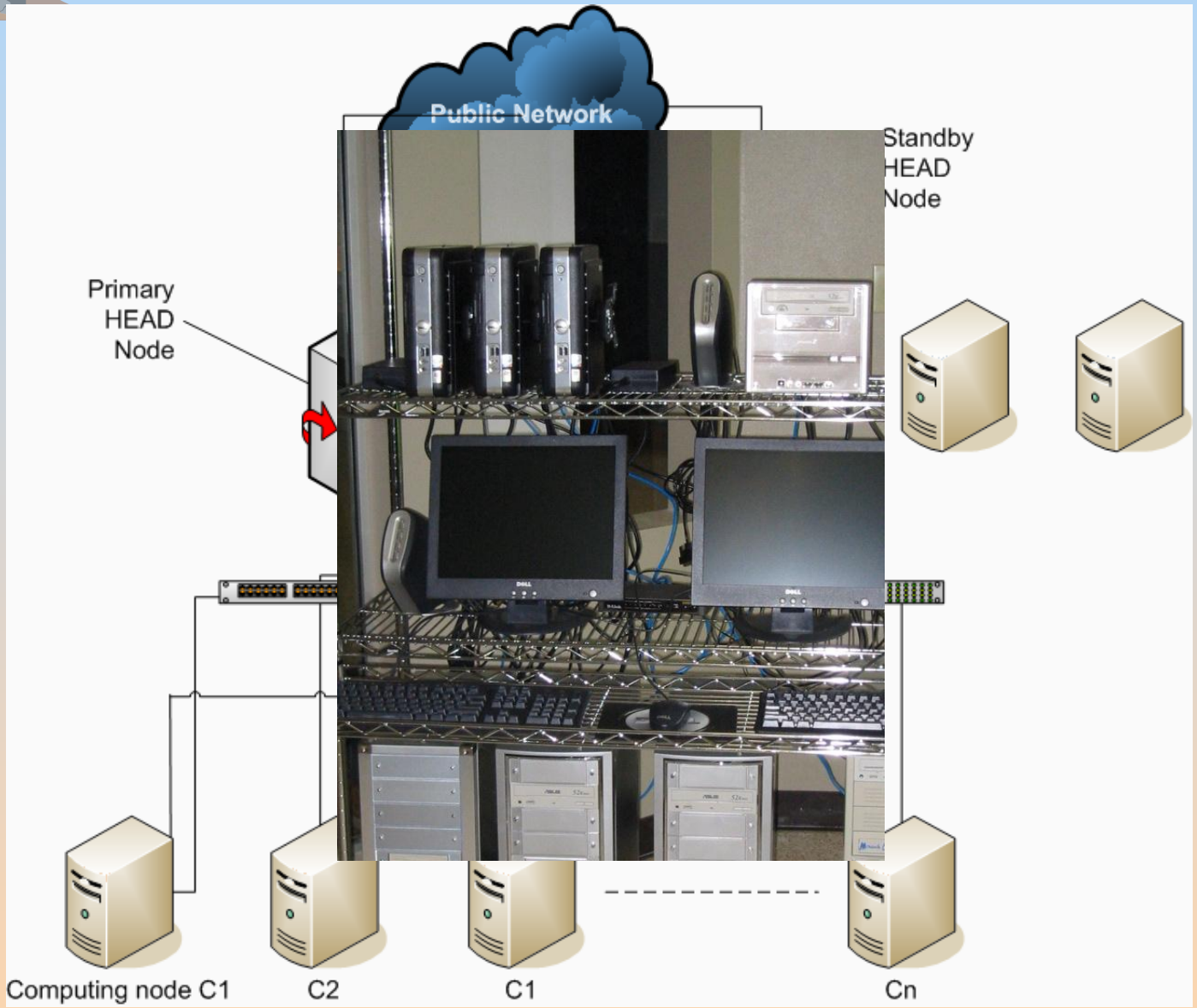


Clustered Systems

- Clustering allows two or more systems to share storage.
- Provides high reliability and/or performance
- *Asymmetric clustering*: one server runs the application while other servers standby.
- *Symmetric clustering*: all N hosts are running the application.

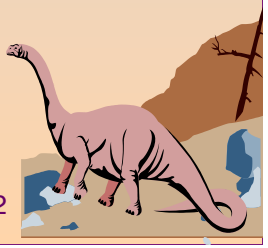


Clustered Systems (continue)



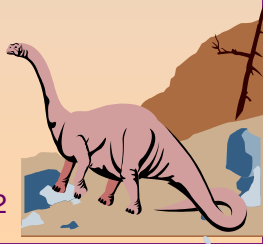
Distributed Systems

- Distribute the computation among several physical processors.
- *Loosely coupled system* – each processor has its own local memory; 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

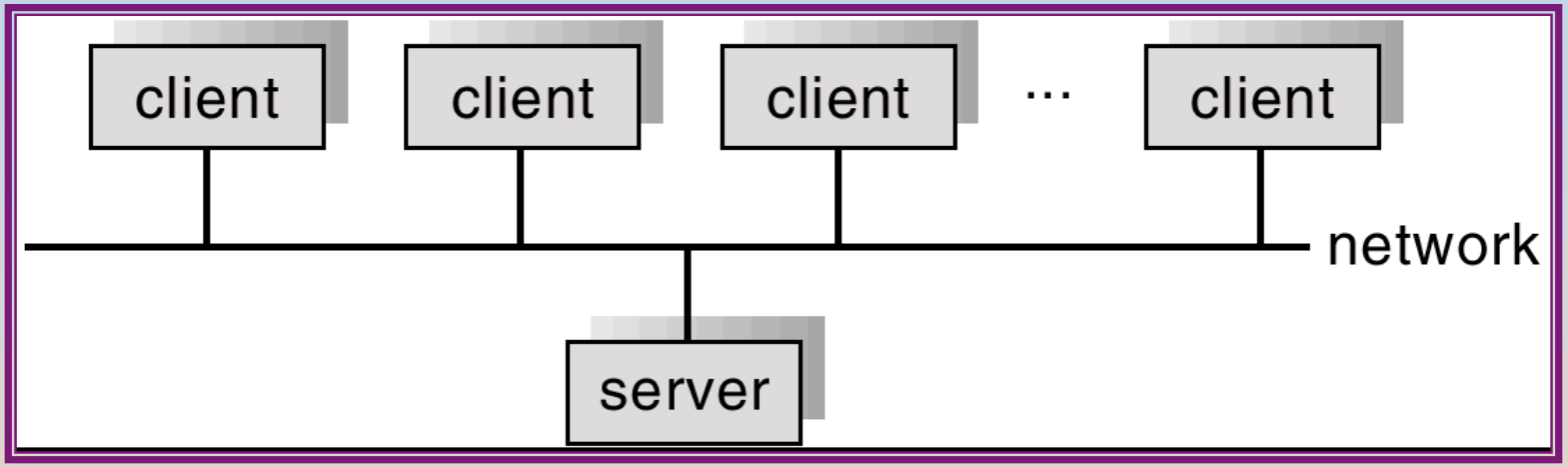


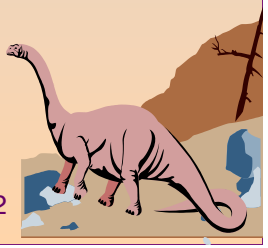
Distributed Systems (cont)

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



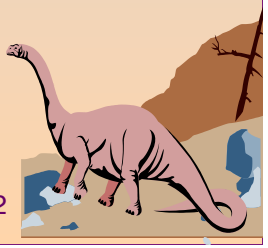
General Structure of Client-Server





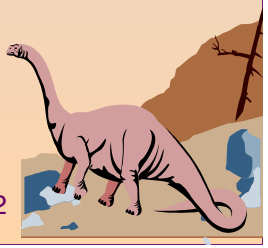
Peer-to-Peer

- "*Any network that relies on computing power at the edges (ends) of a connection rather than in the network itself. Any node is able to initiate or complete any supported transaction with any other node.*" from wikipedia, http://en.wikipedia.org/wiki/Peer-to-peer
- B-2-B
- File sharing services, napster, kazza etc.



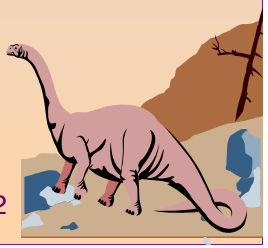
Real-Time Systems

- Often used as a control device in a dedicated application such as controlling scientific experiments, medical imaging systems, industrial control systems, and some display systems.
- Well-defined fixed-time constraints.
- Real-Time systems may be either *hard* or *soft* real-time.



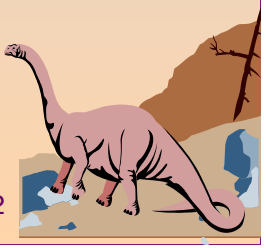
Real-Time Systems (Cont.)

- Hard real-time:
  - Secondary storage limited or absent, data stored in short term memory, or read-only memory (ROM)
  - Conflicts with time-sharing systems, not supported by general-purpose operating systems.
- Soft real-time
  - Limited utility in industrial control of robotics
  - Useful in applications (multimedia, virtual reality) requiring advanced operating-system features.



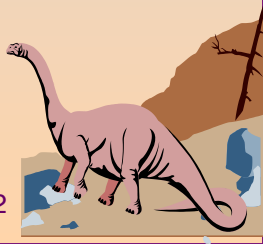
Handheld Systems

- Personal Digital Assistants to Smart telephones
- Old Issues:
  - Limited memory
  - Slow processors
  - Small display screens.
- Smart telephones
  - Much more capable
  - More powerful
  - Issue remain on power consumption

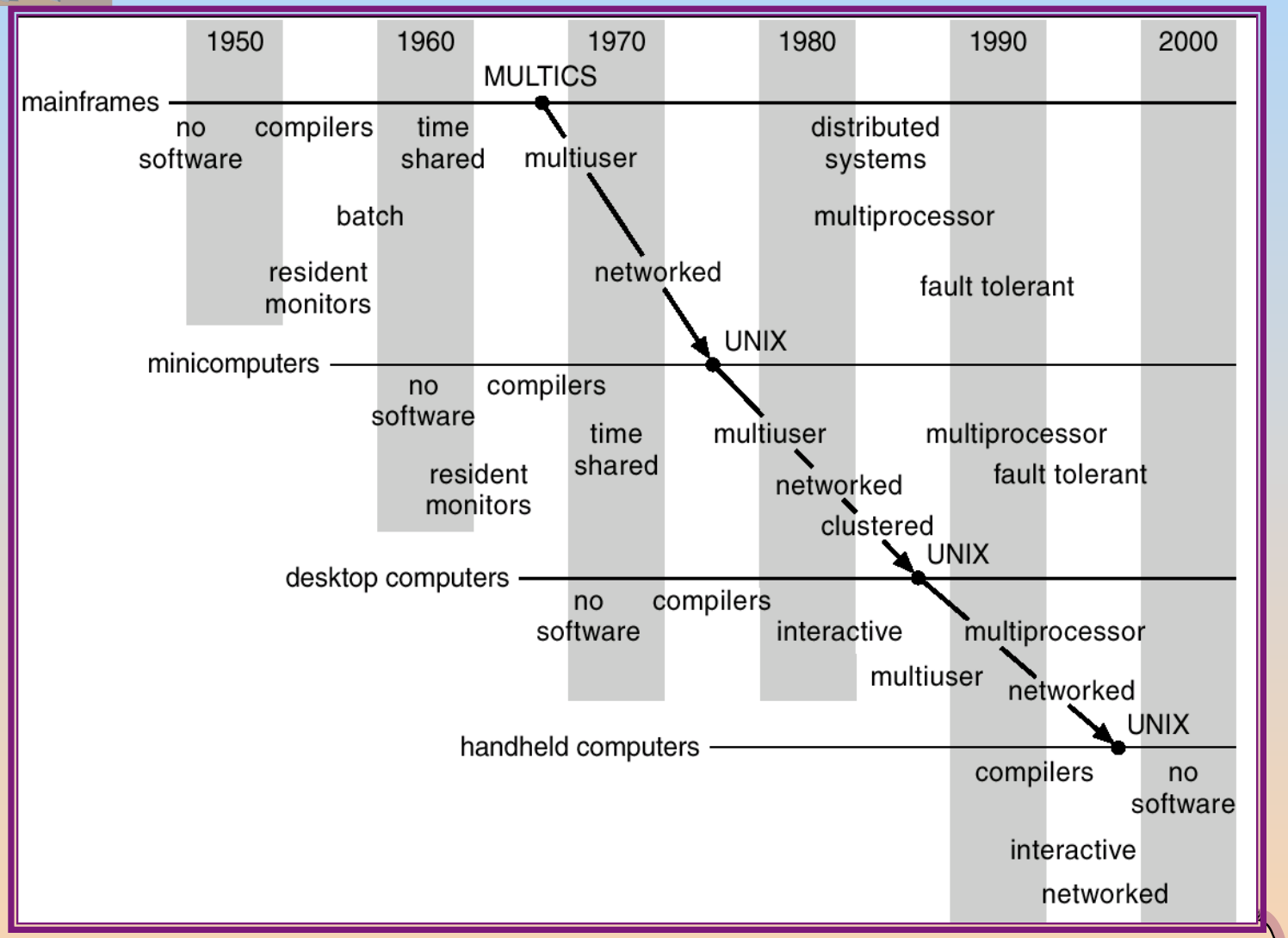


Web-Based Computing

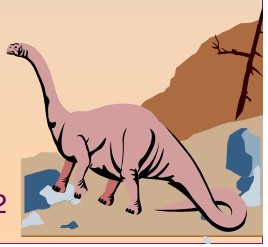
- Web has become ubiquitous
- PCs most prevalent devices
- More devices becoming networked to allow web access
- New category of devices to manage web traffic among similar servers: **load balancers**
- Use of operating systems like Windows, client-side, have evolved into Andriod, IoS Linux and Windows 8, which can be clients and servers & more mobile



Migration of Operating-System Concepts and Features

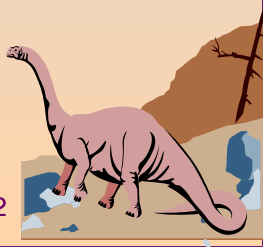


Recap..

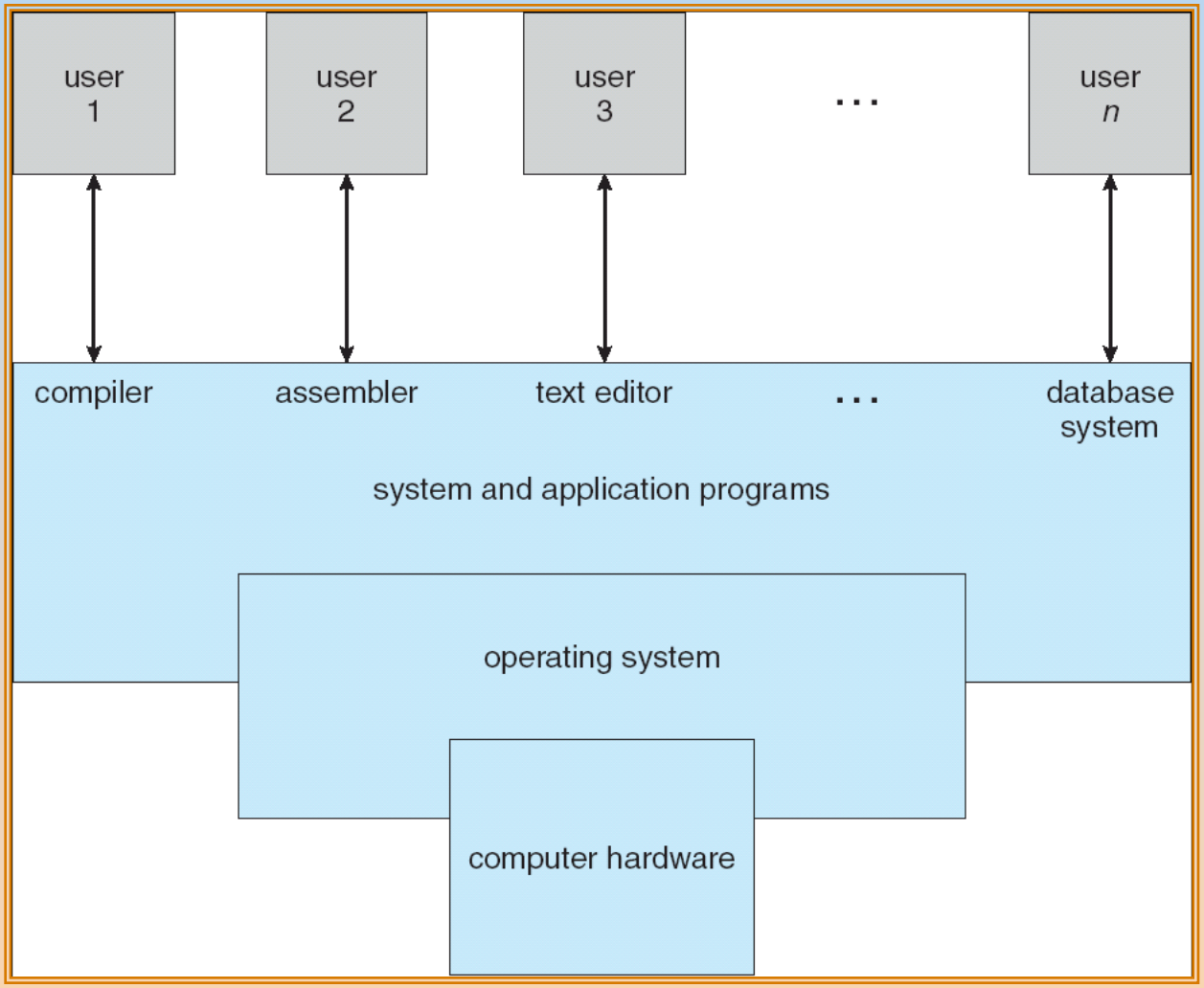


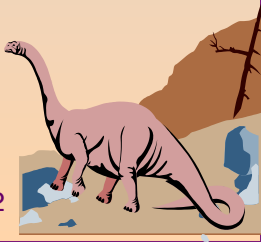
Computer System Structure

- Computer system can be divided into four components
  - Hardware – provides basic computing resources
    - CPU, memory, I/O devices
  - Operating system
    - Controls and coordinates use of hardware among various applications and users
  - Application programs – define the ways in which the system resources are used to solve the computing problems of the users
    - Word processors, compilers, web browsers, database systems, video games
  - Users
    - People, machines, other computers



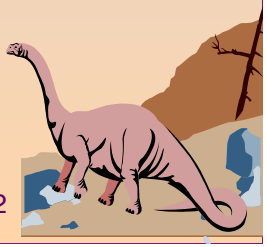
Four Components of a Computer System





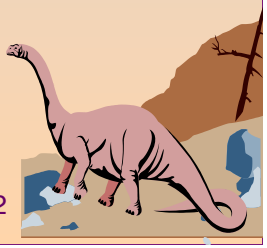
Operating System Definition

- OS is a **resource allocator**
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the computer



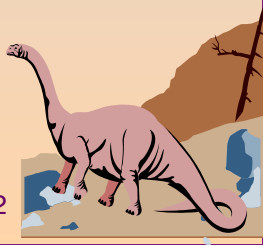
Operating System Definition (Cont.)

- No universally accepted definition
- "Everything a vendor ships when you order an operating system" is good approximation
  - But varies wildly
- "The one program running at all times on the computer" is the **kernel.** Everything else is either a system program (ships with the operating system) or an application program



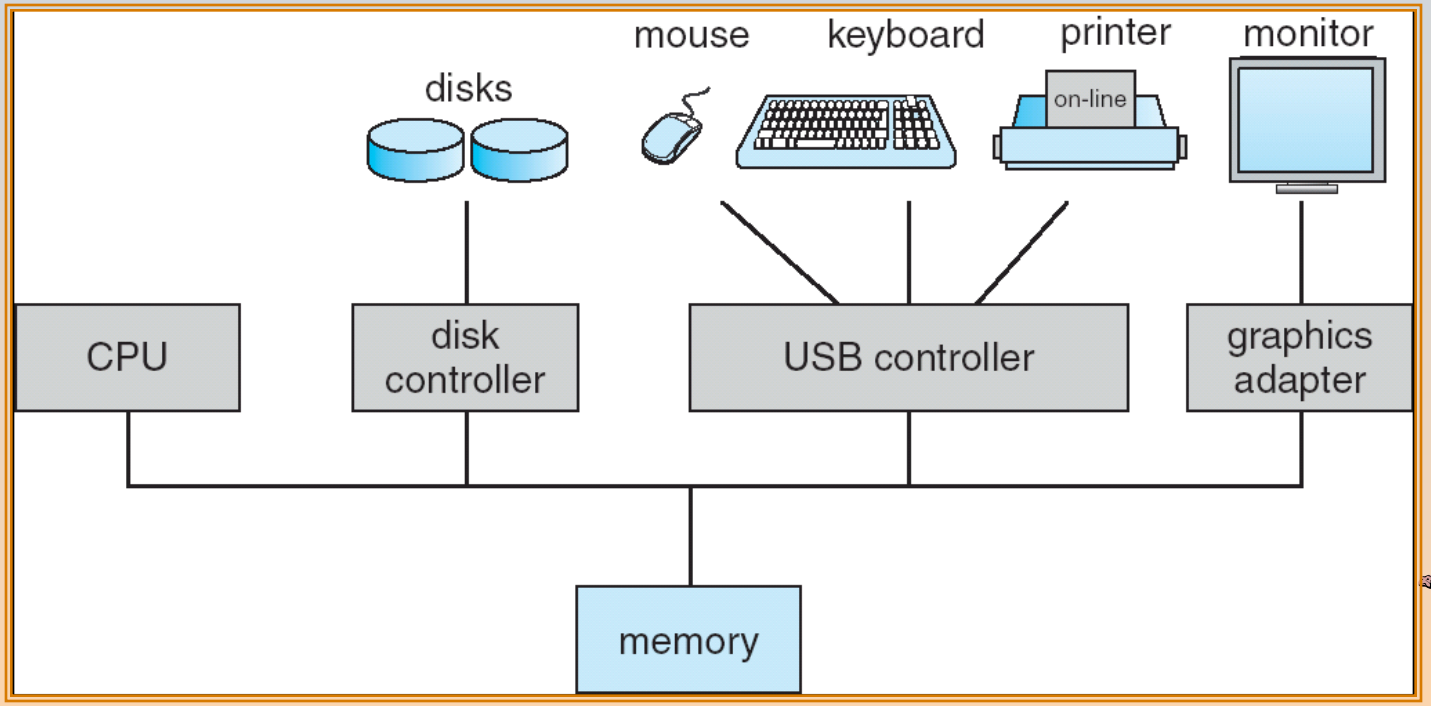
Computer Startup

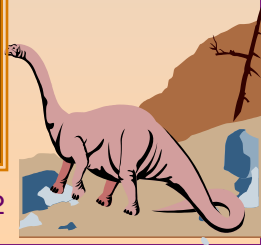
- **bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM or EEPROM, generally known as **firmware**
  - Initializates all aspects of system
  - Loads operating system kernel and starts execution



Computer System Organization

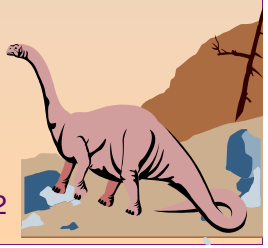
- Computer-system operation
  - One or more CPUs, device controllers connect through common bus providing access to shared memory
  - Concurrent execution of CPUs and devices competing for memory cycles



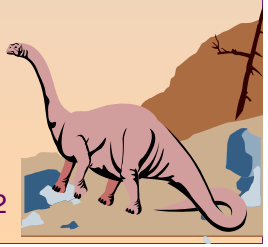


Computer-System Operation

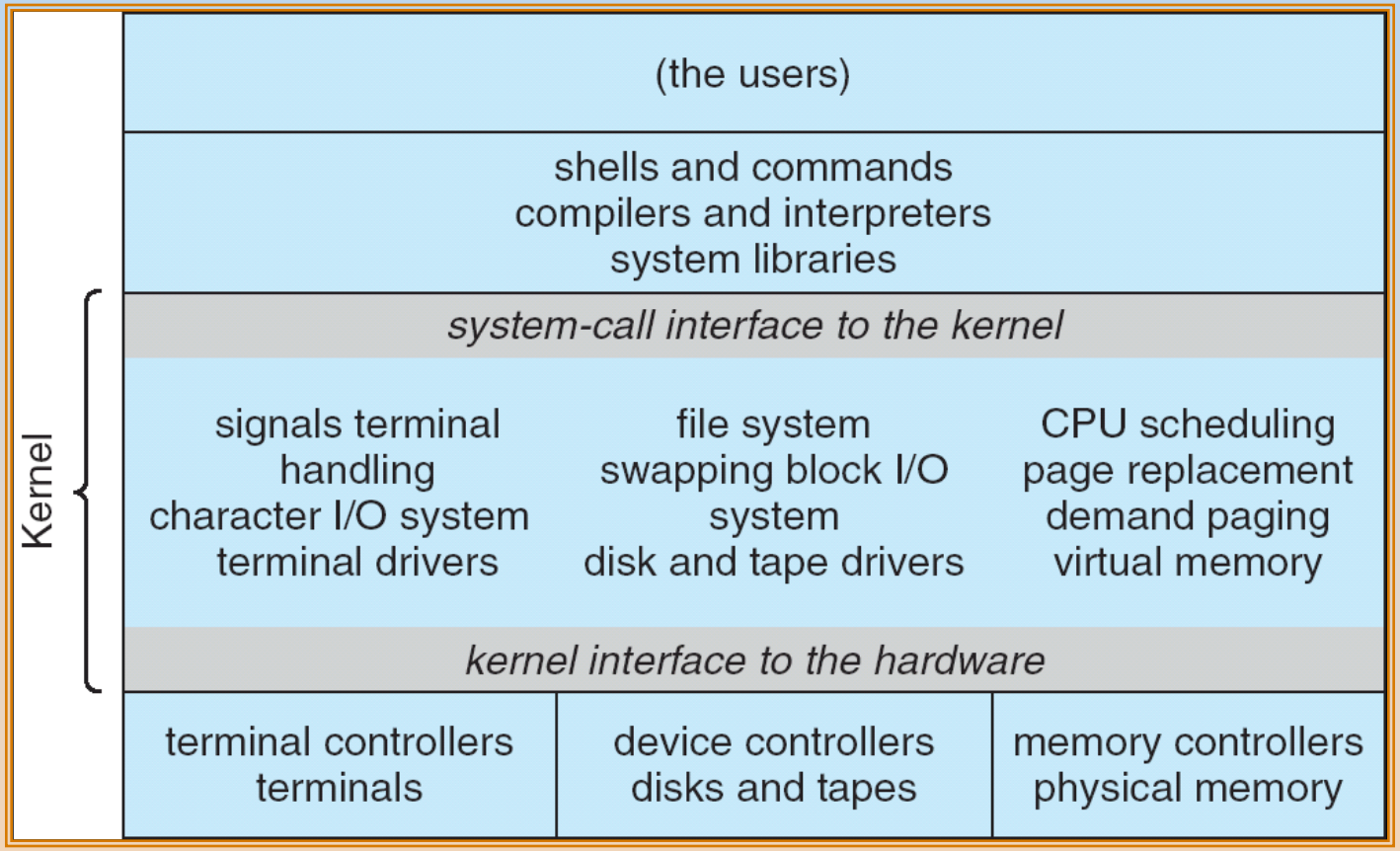
- I/O devices and the CPU can execute concurrently.
- Each device controller is in charge of a particular device type.
- Each device controller has a local buffer.
- CPU moves data from/to main memory to/from local buffers
- I/O is from the device to local buffer of controller.
- Device controller informs CPU that it has finished its operation by causing an *interrupt*.

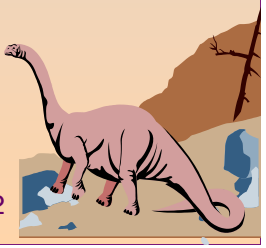


A Quick Tour on various OS Architectures

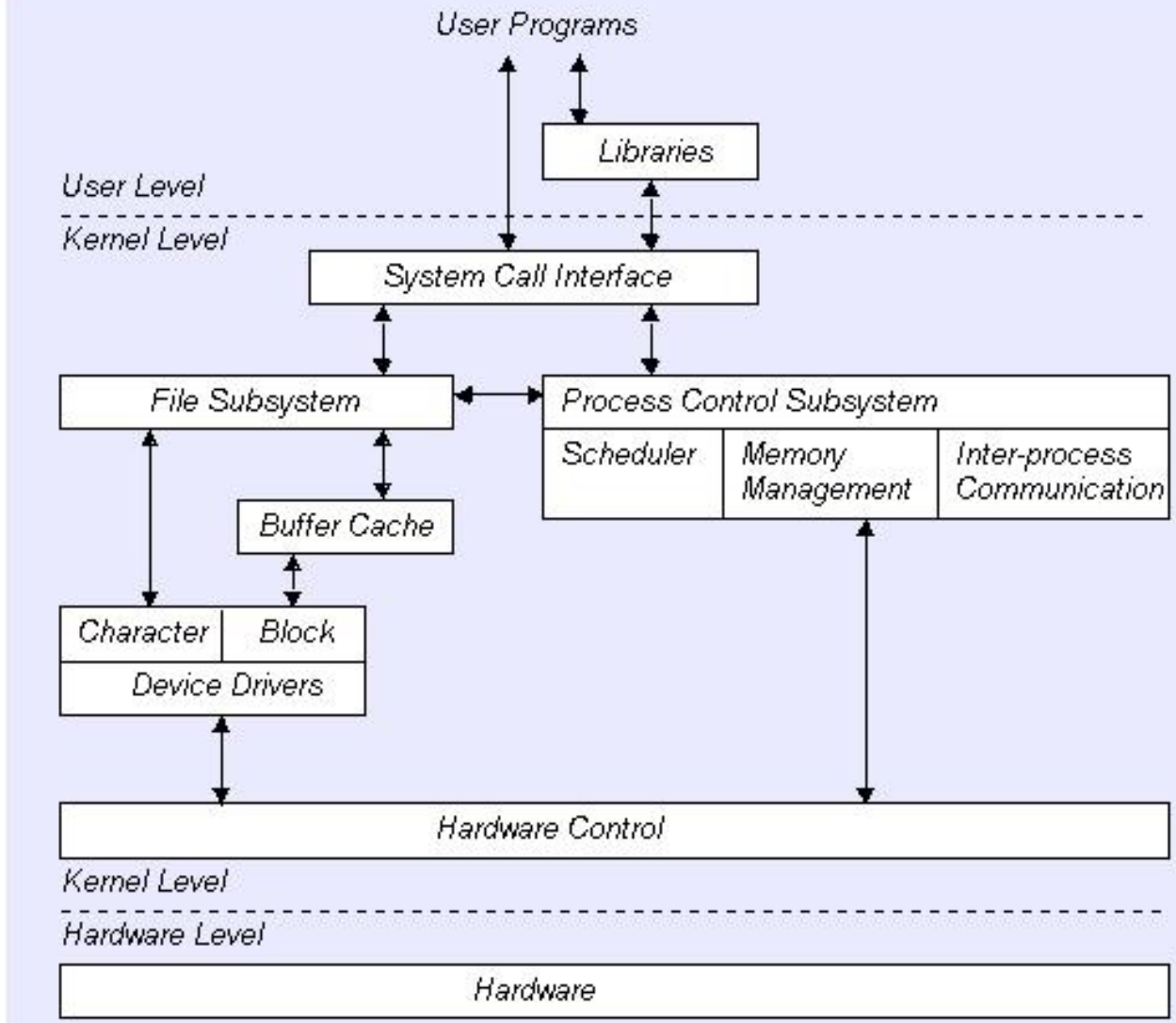


UNIX System Structure



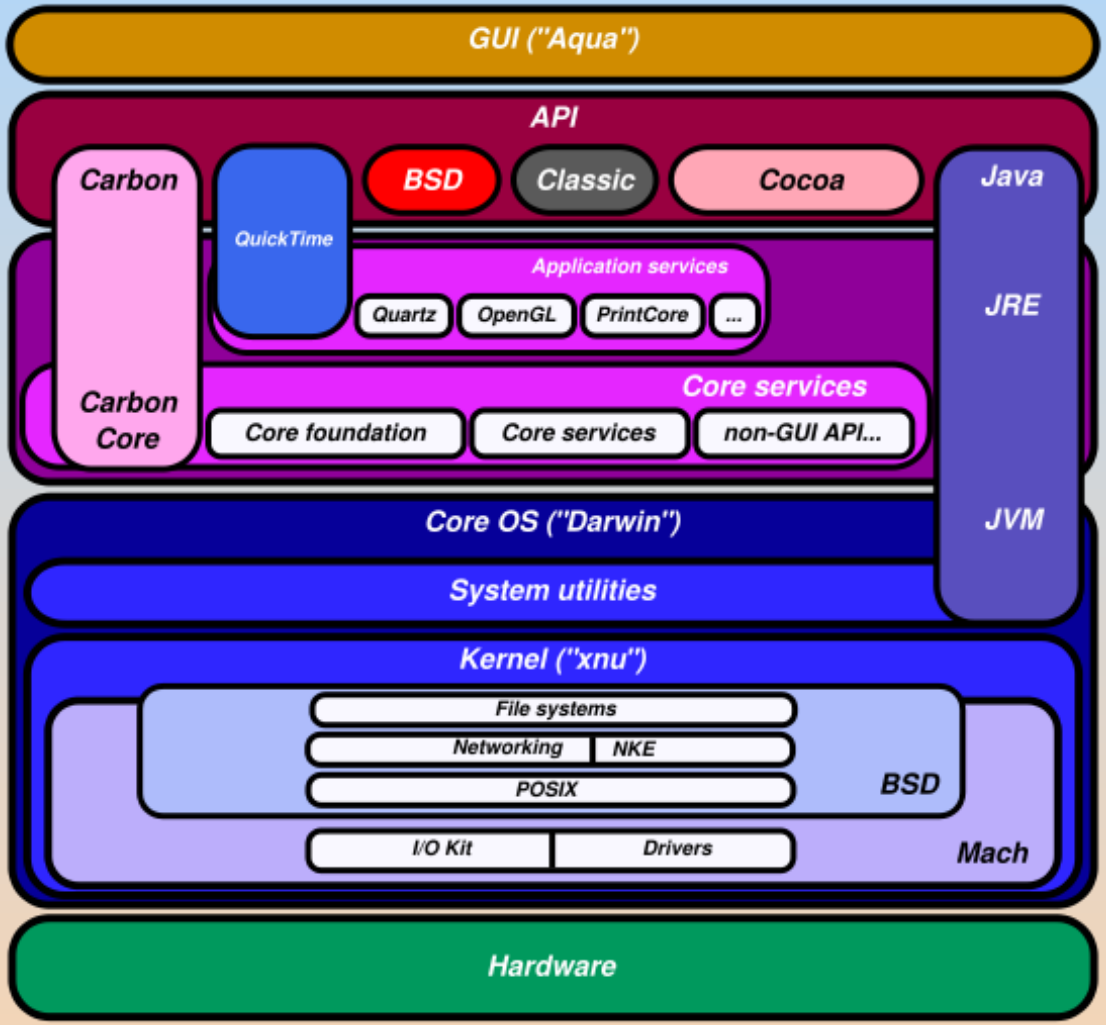


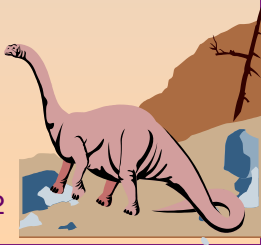
Unix/Linux Kernel interface



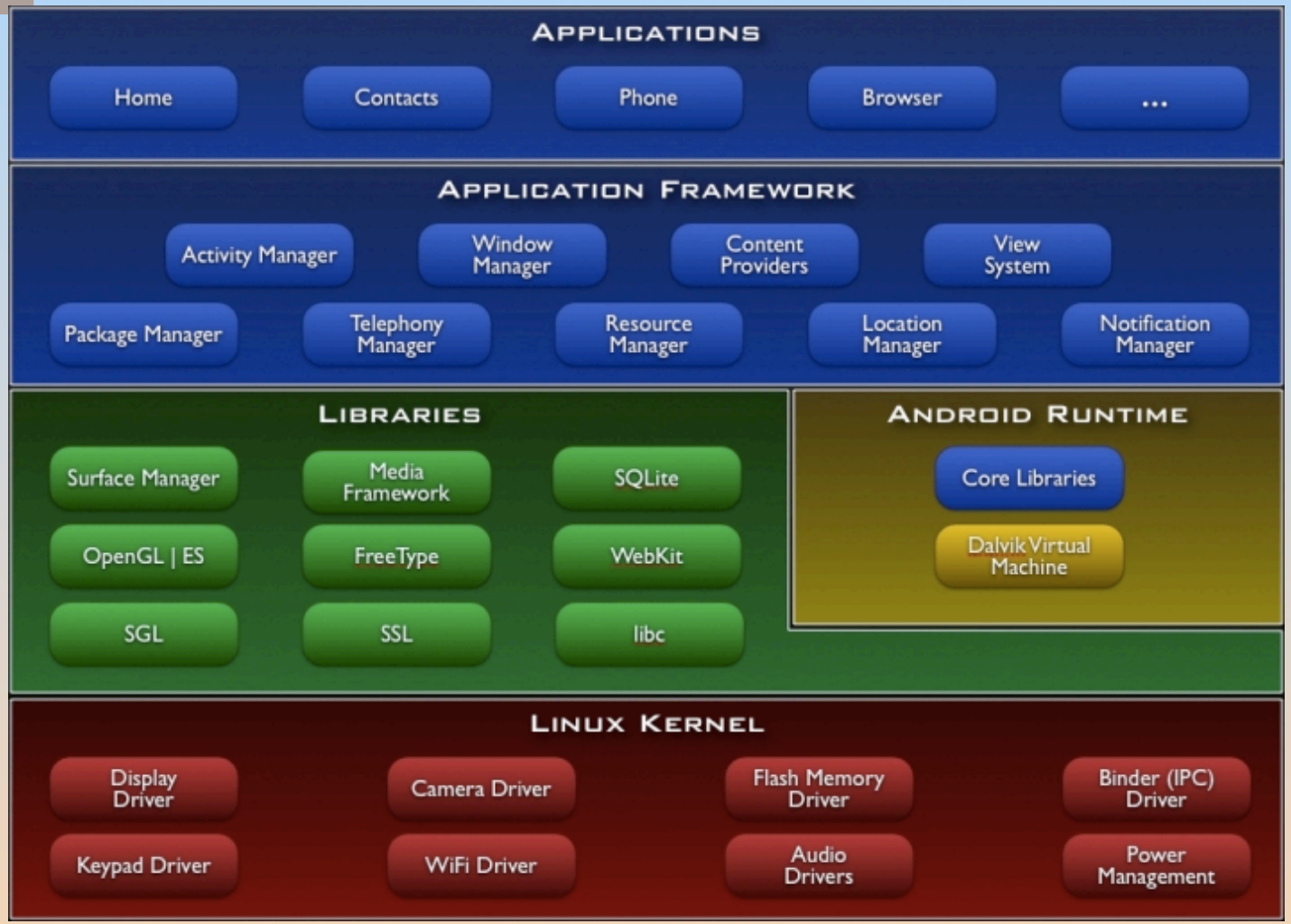
Note: This picture is excerpted from Write a Linux Hardware Device Driver, Andrew O' Shauqhnessy, Unix world

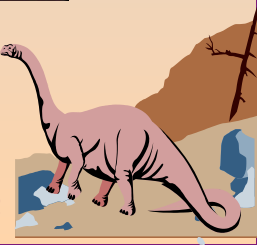
Mac OS


GUI ("Aqua")
API
Carbon
QuickTime
BSD
Classic
Cocoa
Java
Application services
Quartz
OpenGL
PrintCore
...
JRE
Core services
Carbon Core
Core foundation
Core services
non-GUI API...
Core OS ("Darwin")
JVM
System utilities
Kernel ("xnu")
File systems
Networking
NKE
POSIX
BSD
I/O Kit
Drivers
Mach
Hardware

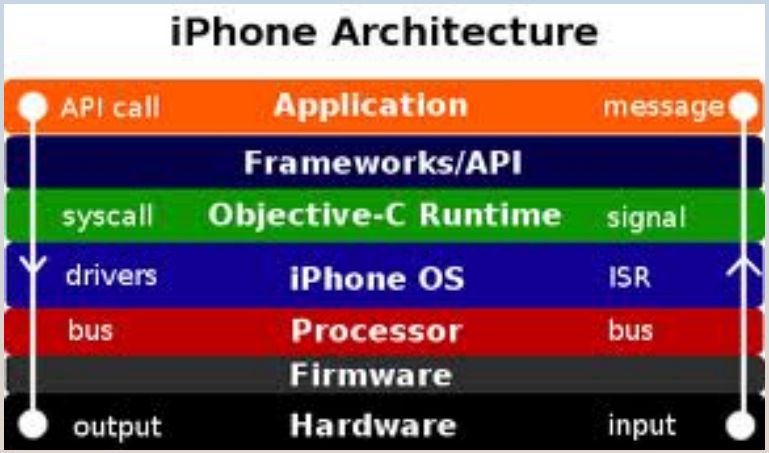


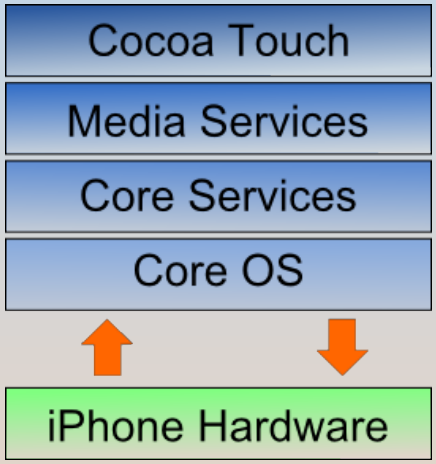
Android

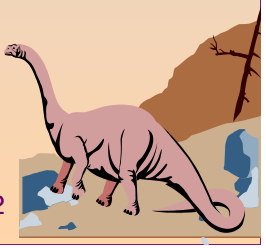

Applications
Home
Contacts
Phone
Browser
...
Application Framework
Activity Manager
Window Manager
Content Providers
View System
Package Manager
Telephony Manager
Resource Manager
Location Manager
Notification Manager
Libraries
Surface Manager
Media Framework
SQLite
OpenGL | ES
FreeType
WebKit
SGL
SSL
libc
Android Runtime
Core Libraries
Dalvik Virtual Machine
Linux Kernel
Display Driver
Camera Driver
Flash Memory Driver
Binder (IPC) Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management

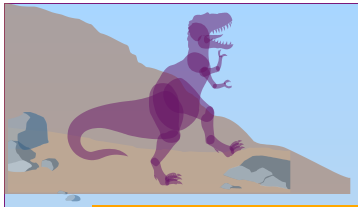


IOS

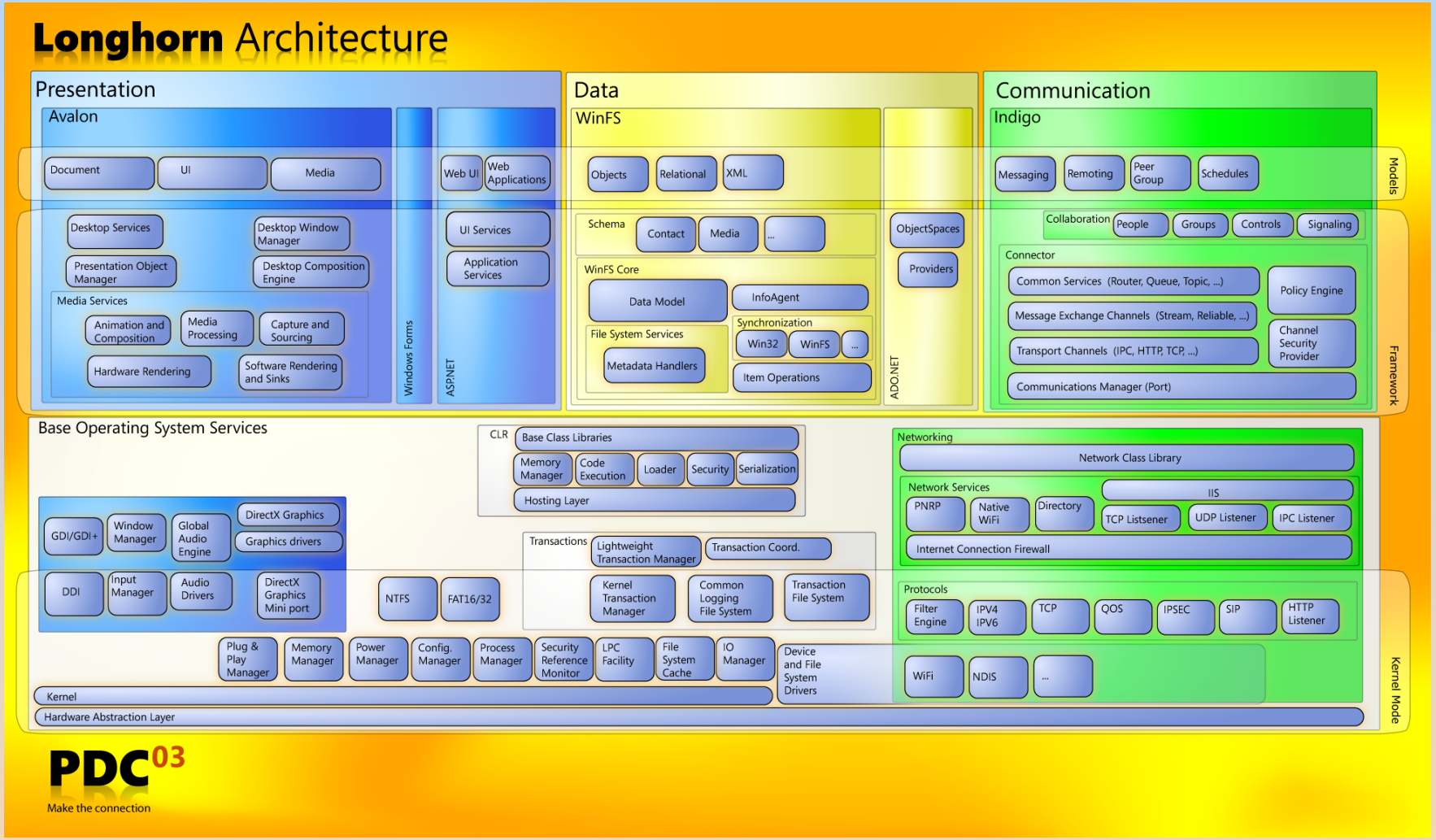

iPhone Architecture
API call
Application
message
Frameworks/API
syscall
Objective-C Runtime
signal
drivers
iPhone OS
ISR
bus
Processor
bus
Firmware
output
Hardware
input

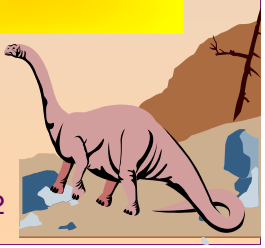

Cocoa Touch
Media Services
Core Services
Core OS
iPhone Hardware





Window


Longhorn Architecture
Presentation
Avalon
Document
UI
Media
Web UI
Web Applications
Desktop Services
Desktop Window Manager
Presentation Object Manager
Desktop Composition Engine
Media Services
Animation and Composition
Media Processing
Capture and Sourcing
Hardware Rendering
Software Rendering and Sinks
Windows Forms
ASP.NET
UI Services
Application Services
Data
WinFS
Objects
Relational
XML
Schema
Contact
Media
...
WinFS Core
Data Model
InfoAgent
Synchronization
Win32
WinFS
...
File System Services
Metadata Handlers
Item Operations
ObjectSpaces
Providers
ADO.NET
Communication
Indigo
Messaging
Remoting
Peer Group
Schedules
Collaboration
People
Groups
Controls
Signaling
Connector
Common Services (Router, Queue, Topic, ...)
Message Exchange Channels (Stream, Reliable, ...)
Transport Channels (IPC, HTTP, TCP, ...)
Communications Manager (Port)
Policy Engine
Channel Security Provider
Models
Framework
Base Operating System Services
CLR
Base Class Libraries
Memory Manager
Code Execution
Loader
Security
Serialization
Hosting Layer
GDI/GDI+
Window Manager
Global Audio Engine
DirectX Graphics
Graphics drivers
DDI
Input Manager
Audio Drivers
DirectX Graphics Mini port
NTFS
FAT16/32
Transactions
Lightweight Transaction Manager
Transaction Coord.
Kernel Transaction Manager
Common Logging File System
Transaction File System
Networking
Network Class Library
Network Services
IIS
PNRP
Native WiFi
Directory
TCP Listsener
UDP Listener
IPC Listener
Internet Connection Firewall
Protocols
Filter Engine
IPV4 IPV6
TCP
QOS
IPSEC
SIP
HTTP Listener
Plug & Play Manager
Memory Manager
Power Manager
Config. Manager
Process Manager
Security Reference Monitor
LPC Facility
File System Cache
IO Manager
Device and File System Drivers
WiFi
NDIS
...
Kernel
Hardware Abstraction Layer
Kernel Mode
PDC03
Make the connection



End of Chapter 1

End of Chapter 1