

# Λειτουργικά Συστήματα

## Κεφάλαιο 2

Dr. Garmpis Aristogiannis - EPDO  
TEI Messolonghi

# Τι είναι ένα Λειτουργικό Σύστημα(Λ.Σ.)

- Είναι μια επεκταμένη μηχανή
  - Κρύβει τις λεπτομέρειες που πραγματοποιούνται κατά την εκτέλεση
  - Προσφέρει στο χρήστη μια απλούστερη εικονική μηχανή για επικοινωνία με το υλικό
- Είναι ένας διαχειριστής πόρων
  - Κάθε πρόγραμμα παίρνει **χρόνο** από έναν πόρο
  - Κάθε πρόγραμμα παίρνει **χώρο** από έναν πόρο



# Στόχοι Λειτουργικών Συστημάτων

- Ευκολία
  - Κάνει τον υπολογιστή φιλικότερο προς τον χρήστη
- Απόδοση
  - Βοηθά στην καλύτερη χρήση των πόρων του συστήματος
- Δυνατότητα αναβάθμισης
  - Προσφέρει την δυνατότητα αποδοτικής δημιουργίας, ελέγχου και βελτιώσεων νέων υπηρεσιών συστήματος χωρίς τον επηρεασμό του συστήματος



# Επίπεδα Υπολογιστικών Συστημάτων

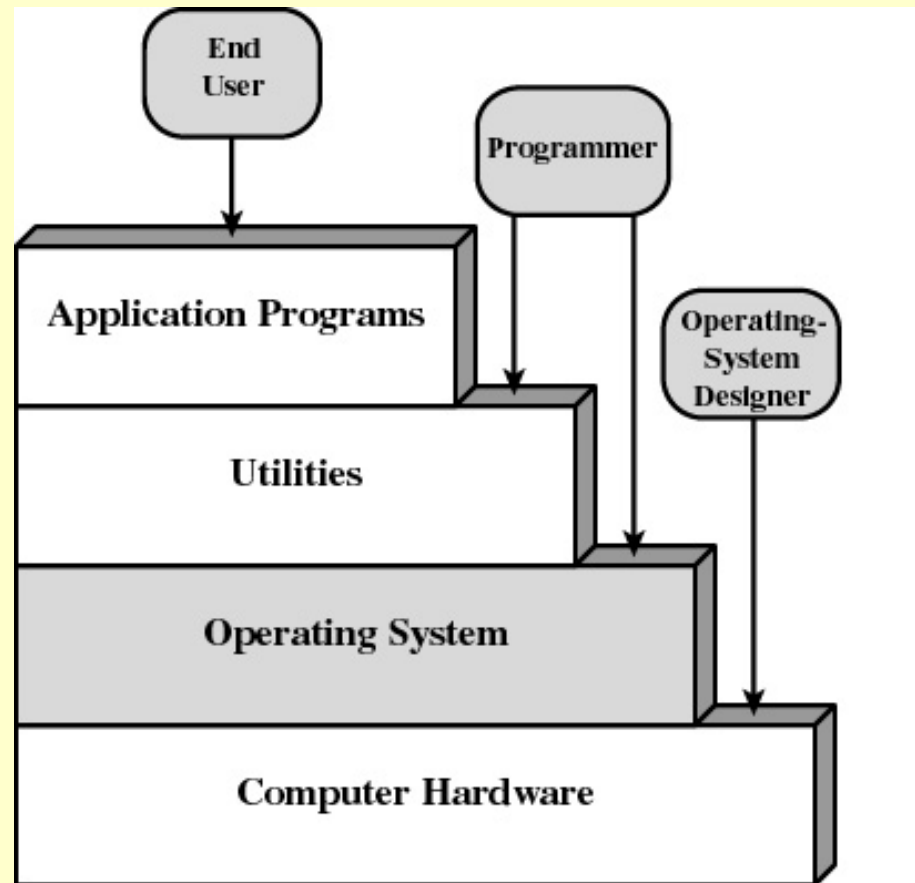
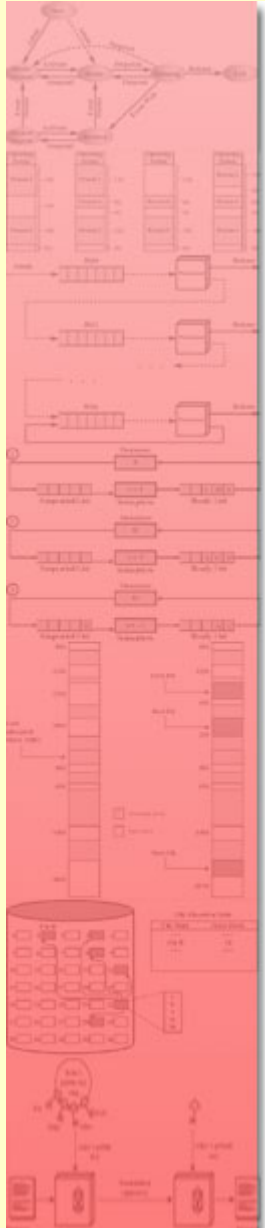


Figure 2.1 Layers and Views of a Computer System  
IEI Messolonghi



# Υπηρεσίες που προσφέρει ένα Λειτουργικό Σύστημα

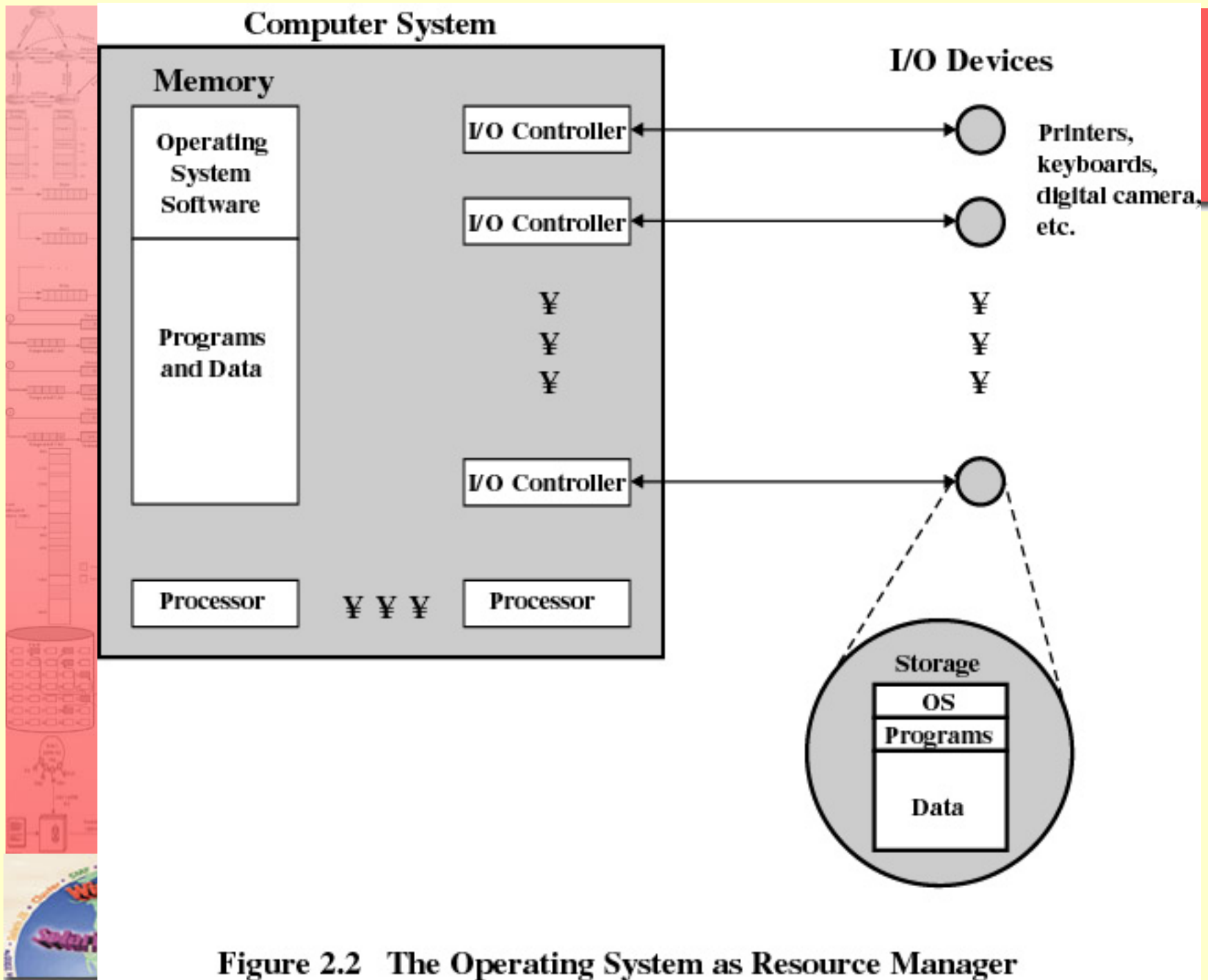
- Προγραμματισμός
  - Αποσφαλματωτές & Editors
- Εκτέλεση προγραμμάτων
- Πρόσβαση σε μονάδες I/O
- Έλεγχος πρόσβασης σε αρχεία
- Έλεγχος πρόσβασης στο σύστημα



# Υπηρεσίες που προσφέρει ένα Λειτουργικό Σύστημα (2)

- Έλεγχος Λαθών
  - Εσωτερικά & Εξωτερικά σφάλματα υλικού
    - Λάθη μνήμης
    - Σφάλματα συσκευών
  - Σφάλματα Λογισμικού
    - Αριθμητική υπερχείλιση
    - Πρόσβαση σε μη επιτρεπτή θέση μνήμης
  - Αδυναμία ικανοποίησης αιτήσεων για παραχώρηση πόρων από εφαρμογές
- «Λογιστικές» Υπηρεσίες
  - Στατιστικά
  - Έλεγχος απόδοσης





**Figure 2.2 The Operating System as Resource Manager**

# Πυρήνας

- Το τμήμα του λειτουργικού συστήματος που βρίσκεται συνεχώς στην κύρια μνήμη
- Περιλαμβάνει τις πλέον χρησιμοποιούμενες εντολές
- Θα το βρείτε στην βιβλιογραφία σαν nucleus, kernel, core



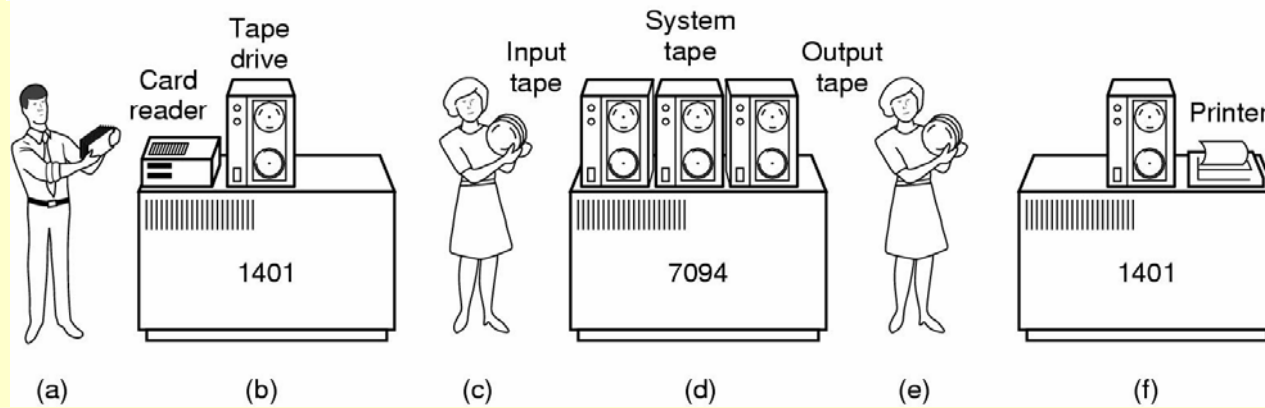


# Εξέλιξη των Υπολογιστικών Συστημάτων(1)

- Σειριακή Επεξεργασία
  - Ανυπαρξία Λ.Σ.
  - Οι μηχανές ελέγχονταν από κονσόλα με λυχνίες και διακόπτες, μηχανή εισόδου και εκτυπωτή
  - Υπήρχε βιβλίο εκτέλεσης εργασιών
  - Η προετοιμασία για την εκτέλεση ενός προγράμματος απαιτούσε κάθε φορά την φόρτωση του μεταγλωττιστή, του πηγαίου κώδικα, την μεταγλώττιση και διασύνδεση του προγράμματος, την φόρτωση των δεδομένων και τέλος την εκτέλεση του προγράμματος



# Εξέλιξη των Υπολογιστικών Συστημάτων(1)



- Απλά Συστήματα Δέσμης (Batch Systems)

- Σύστημα Ελέγχου (Πρώιμο Λ.Σ.)

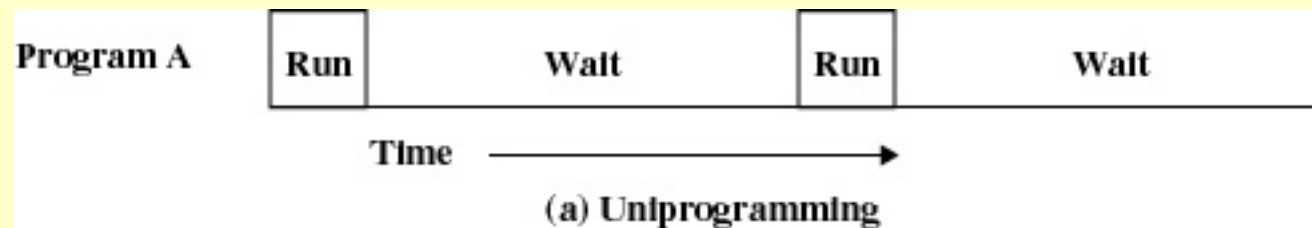
- Πρόγραμμα που ελέγχει την εκτέλεση των προγραμμάτων
    - Εκτέλεση προγραμμάτων σε δέσμη
    - Τα προγράμματα σώζουν τα αποτελέσματα και ειδοποιούν τον χρήστη
    - Ο χρήστης παραλαμβάνει και εκτυπώνει τα αποτελέσματα

Dr. Garmpis Aristogiannis - EPDO  
TEI Messolonghi

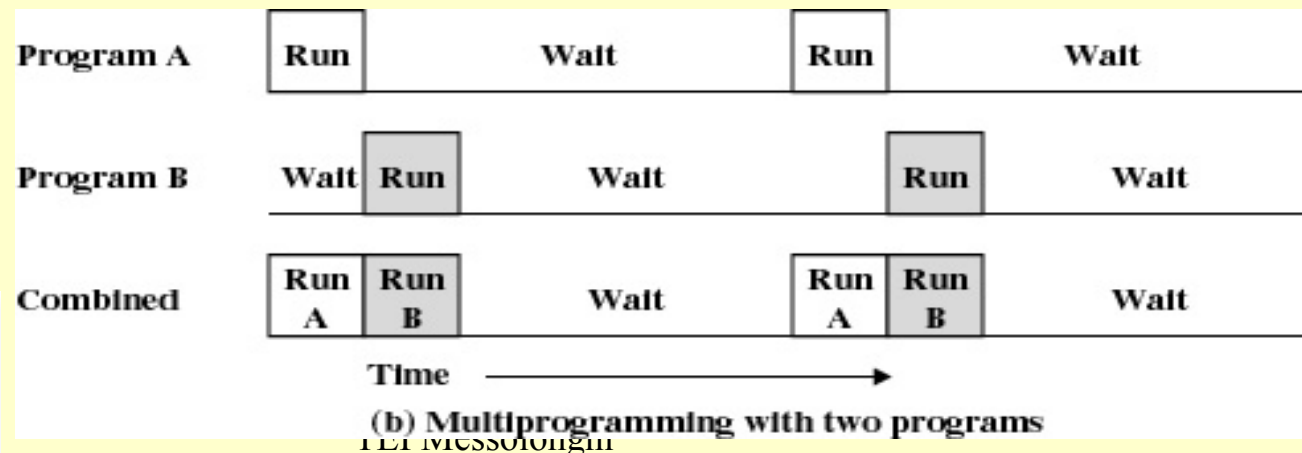


# Σειριακός προγραμματισμός & Πολυπρογραμματισμός(1)

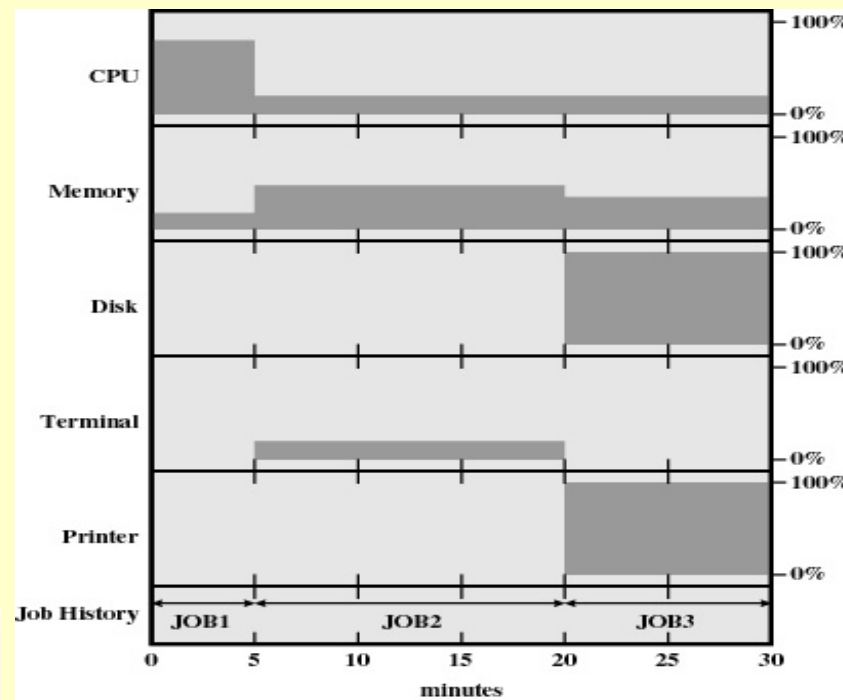
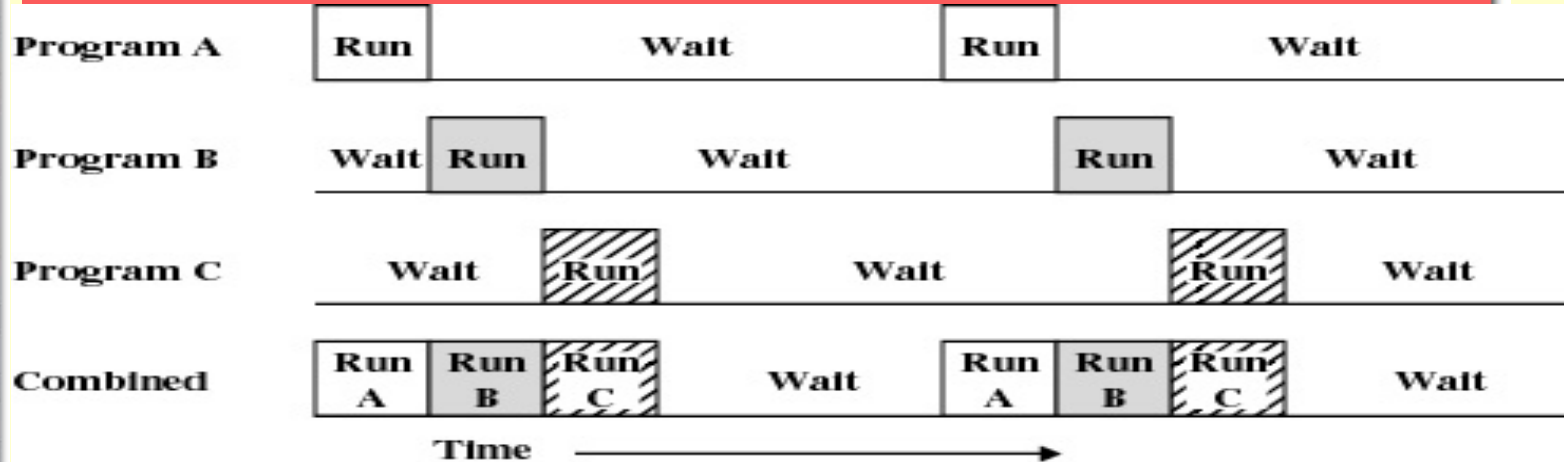
- Ο επεξεργαστής περιμένει να ολοκληρωθεί η εκτέλεση μιας εντολής I/O πριν συνεχίσει



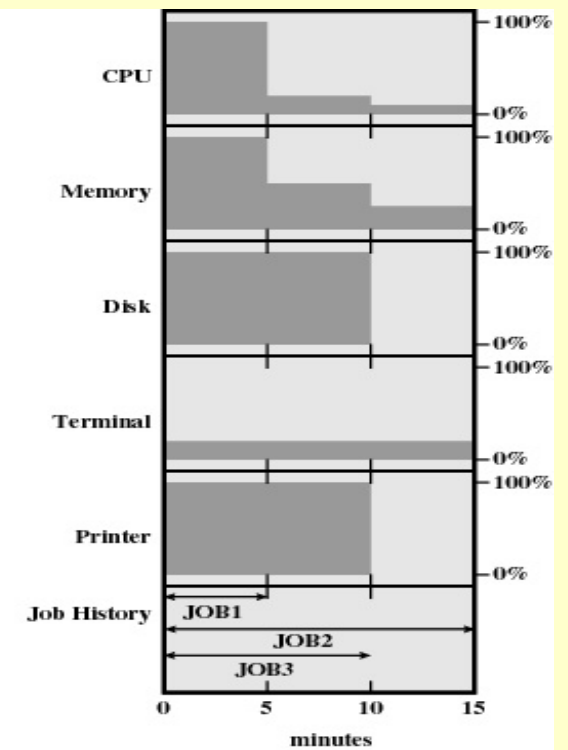
- Όταν ένα πρόγραμμα πρέπει να περιμένει για μια εντολή I/O, ο επεξεργαστής εκτελεί ένα άλλο πρόγραμμα



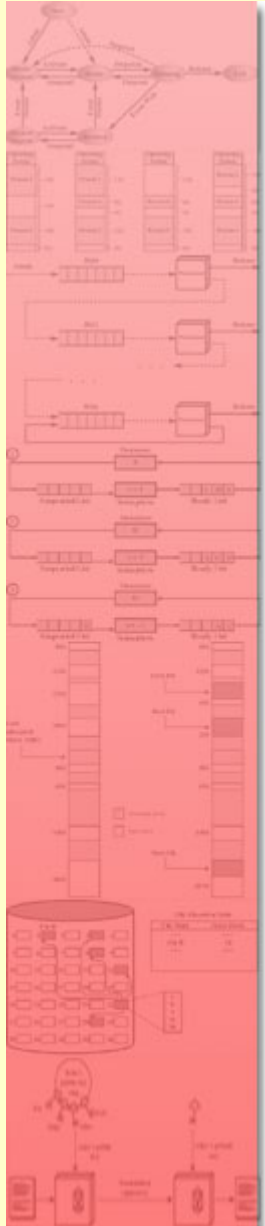
# Σειριακός προγραμματισμός & Πολυπρογραμματισμός(2)



(a) Uniprogramming



(b) Multiprogramming



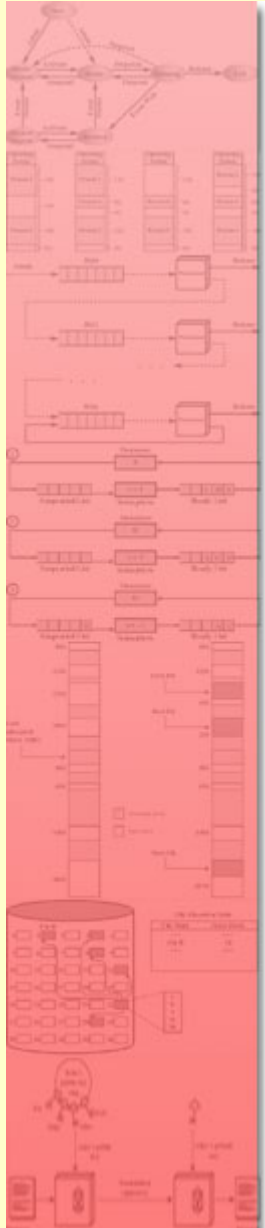
# Παράδειγμα

	ΠΡΓ1	ΠΡΓ2	ΠΡΓ3
Τύπος	Υπολογισμοί	I/O	I/O
Διάρκεια	5 min.	15 min.	10 min.
Μνήμη	50K	100 K	80 K
Δίσκος?	Όχι	Όχι	Ναί
Τερματικό	Όχι	Ναι	Όχι
Εκτυπωτής?	Όχι	Όχι	Ναί

	Σειριακή Εκτέλεση	Πολυπρογραμματισμός
Χρήση Επεξεργαστή	22%	43%
Χρήση Μνήμης	30%	67%
Χρήση Δίσκου	33%	67%
Χρήση Εκτυπωτή	33%	67%
Διάρκεια Εκτέλεσης	30 min.	15 min.
Απόδοση	6 πργ/hr	12 πργ/hr
Μέσ. χρόνος ολοκλ.	18 min.	10 min.

Dr. Garmpis Aristogiannis - EPDO

Μέσ. χρόνος ολοκλ. TEI Messolonghi



# Διαμέριση Χρόνου

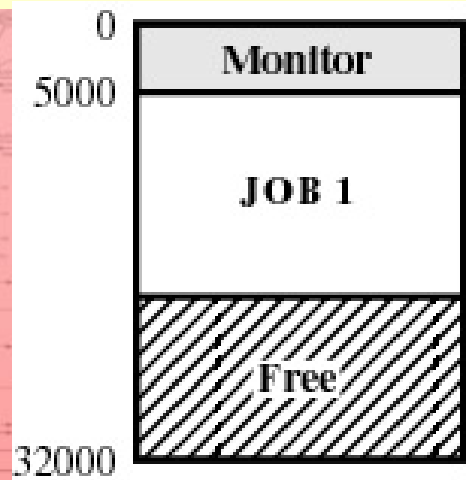
- Χρήση πολυπρογραμματισμού για την αντιμετώπιση πολλαπλών προγραμμάτων
- Πολλαπλοί χρήστες κάνουν ταυτόχρονη χρήση του συστήματος μέσω τερματικών
- Ο χρόνος του επεξεργαστή διαμοιράζεται ανάμεσα στους χρήστες



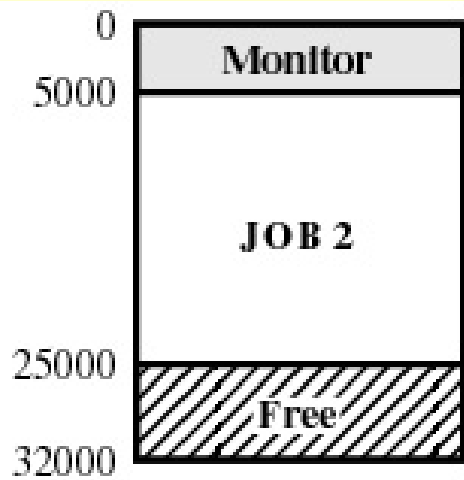
# Batch Multiprogramming versus Time Sharing

	<b>Batch Multiprogramming</b>	<b>Time Sharing</b>
Principal objective	Maximize processor use	Minimize response time
Source of directives to operating system	Job control language commands provided with the job	Commands entered at the terminal

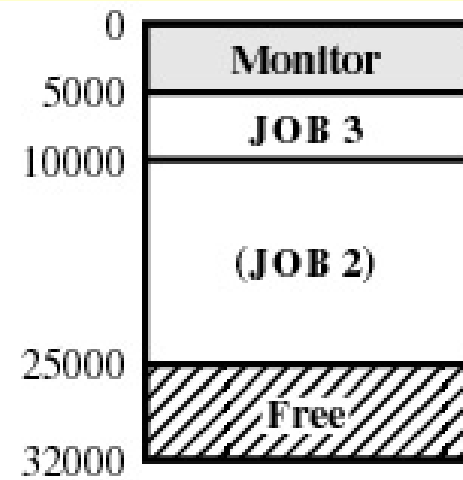




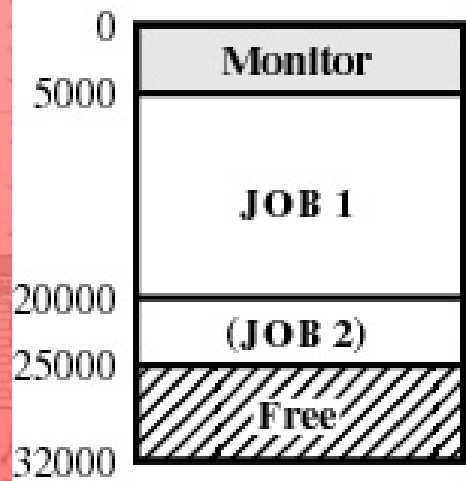
(a)



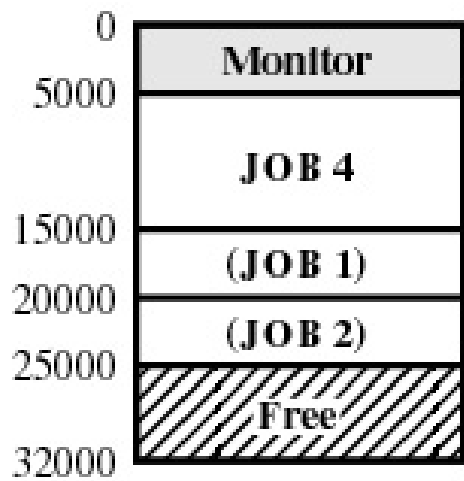
(b)



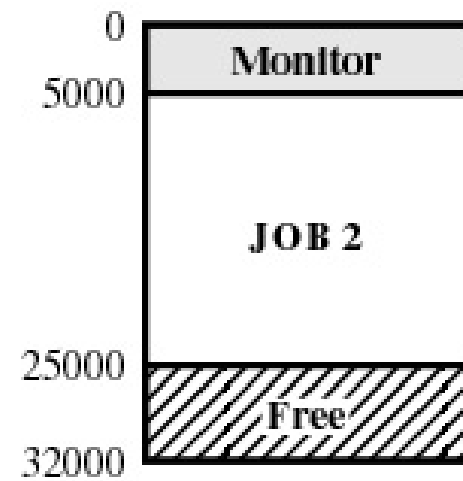
(c)



(d)



(e)



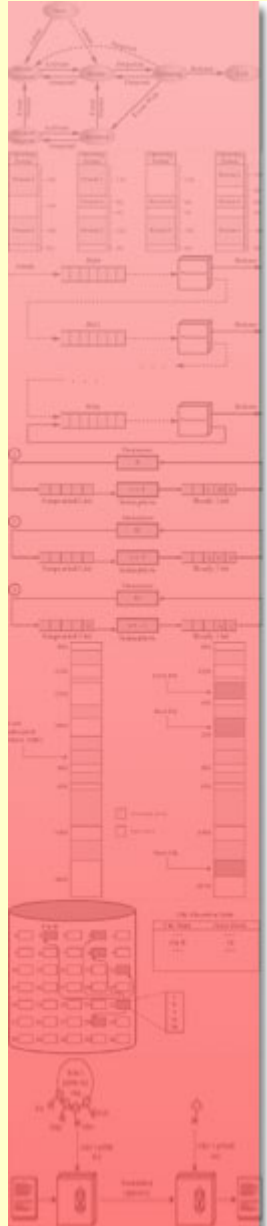
(f)

Figure 2.7 CTSS Operation



# Major Achievements

- Processes
- Memory Management
- Information protection and security
- Scheduling and resource management
- System structure



Dr. Garmpis Aristogiannis - EPDO  
TEI Messolonghi

# Processes

- A program in execution
- An instance of a program running on a computer
- The entity that can be assigned to and executed on a processor
- A unit of activity characterized by a single sequential thread of execution, a current state, and an associated set of system resources



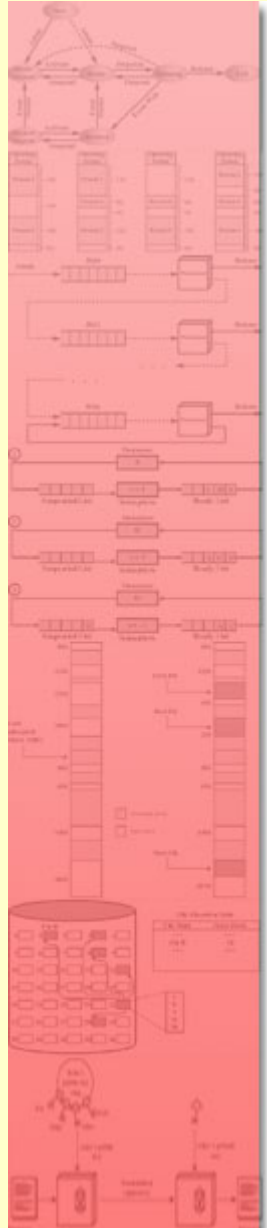
# Difficulties with Designing System Software

- Improper synchronization
  - ensure a process waiting for an I/O device receives the signal
- Failed mutual exclusion
- Nondeterminate program operation
  - program should only depend on input to it, not relying on common memory areas
- Deadlocks



# Process

- Consists of three components
  - An executable program
  - Associated data needed by the program
  - Execution context of the program
    - All information the operating system needs to manage the process



# Process

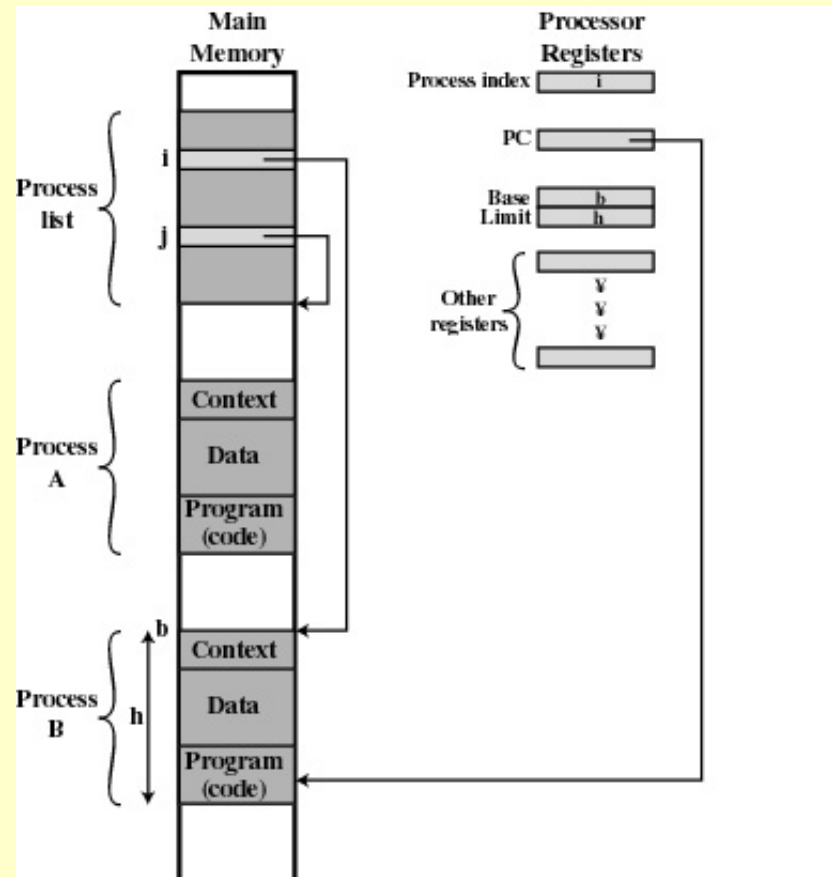
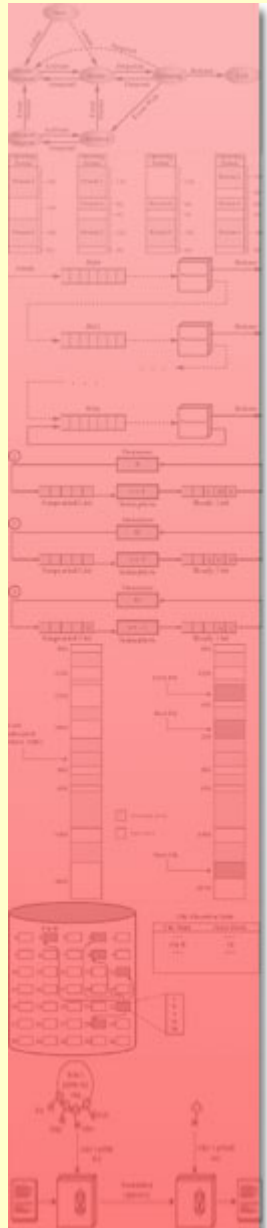


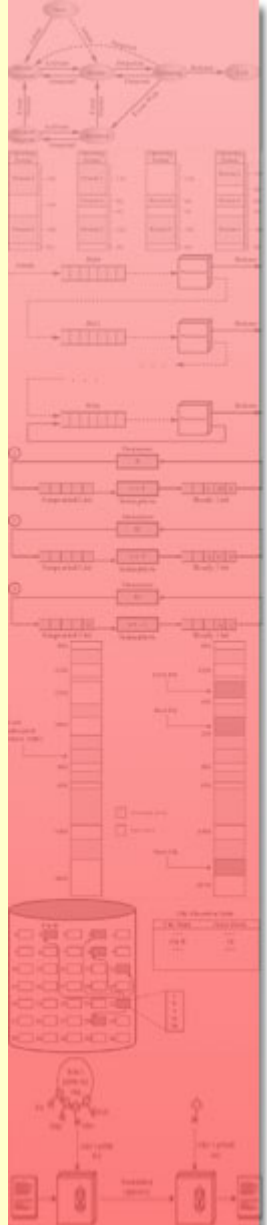
Figure 2.8 Typical Process Implementation

TEI Messolonghi



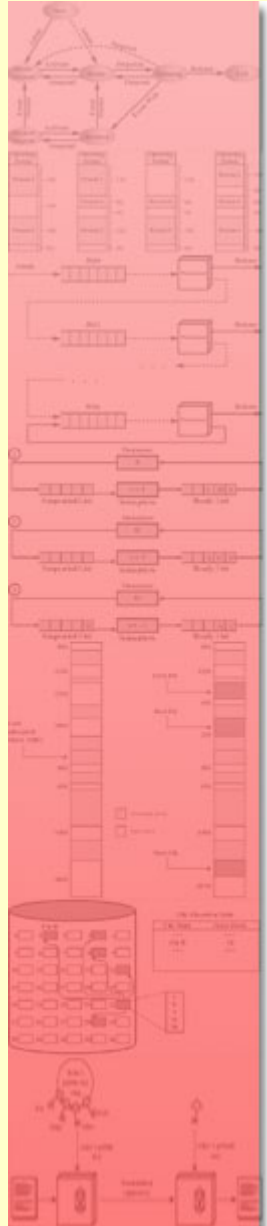
# Memory Management

- Process isolation
- Automatic allocation and management
- Support for modular programming
- Protection and access control
- Long-term storage



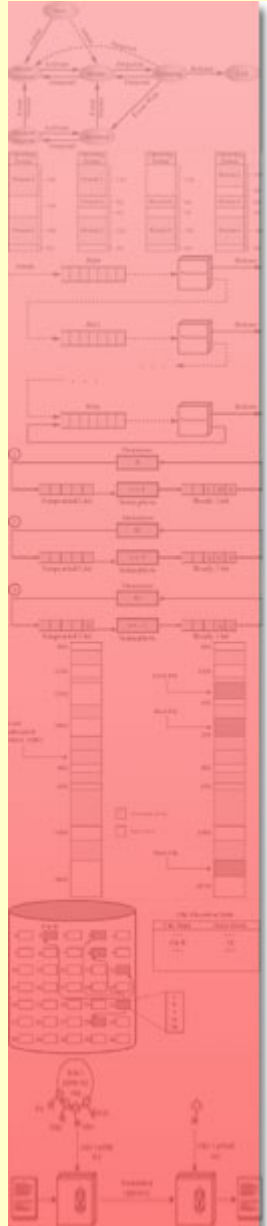
# Virtual Memory

- Allows programmers to address memory from a logical point of view
- While one process is written out to secondary store and the successor process read in there in no hiatus



# File System

- Implements long-term store
- Information stored in named objects called files

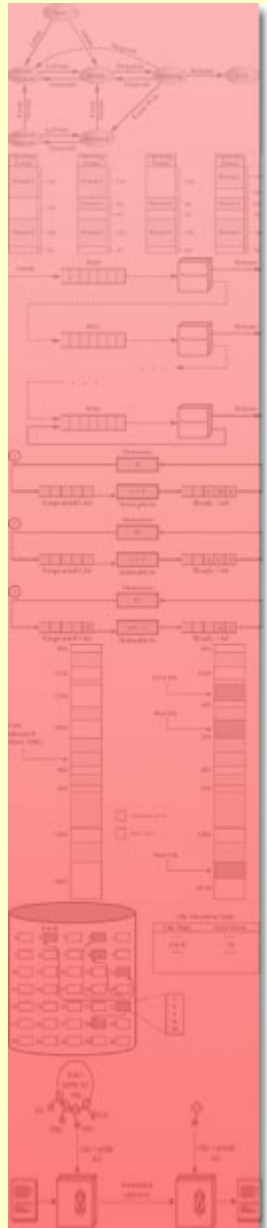




# Paging

- Allows process to be comprised of a number of fixed-size blocks, called pages
- Virtual address is a page number and an offset within the page
- Each page may be located any where in main memory
- Real address or physical address in main memory

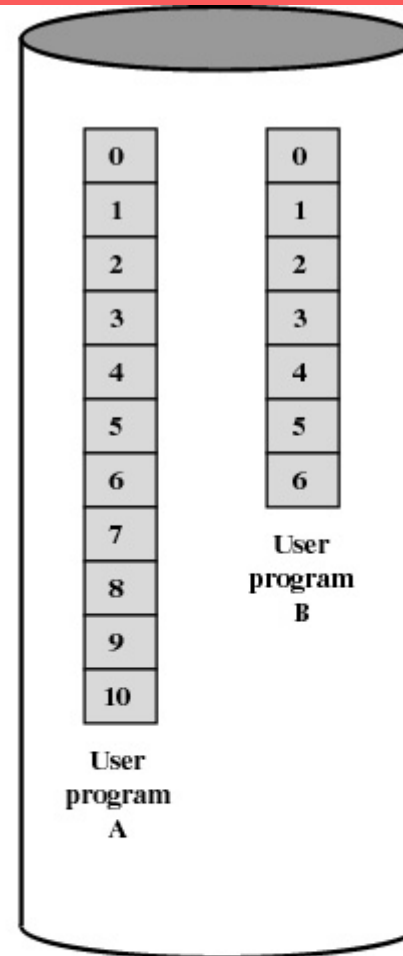




A.1			
	A.0	A.2	
	A.5		
B.0	B.1	B.2	B.3
		A.7	
	A.9		
		A.8	
B.4	B.5	B.6	

**Main Memory**

Main memory consists of a number of fixed-length frames, equal to the size of a page. For a program to execute, some or all of its pages must be in main memory.



**Disk**

Secondary memory (disk) can hold many fixed-length pages. A user program consists of some number of pages. Pages for all programs plus the operating system are on disk, as are files.

**Figure 2.9 Virtual Memory Concepts**  
TEI Messolonghi

# Virtual Memory Addressing

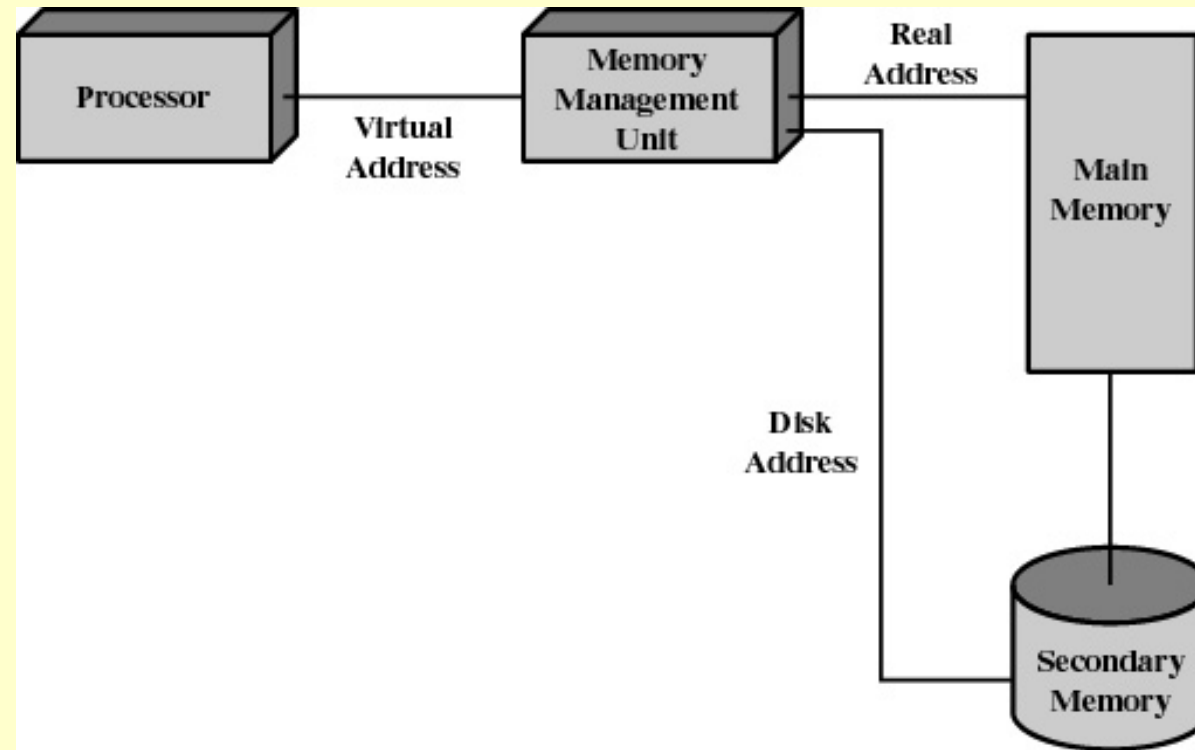
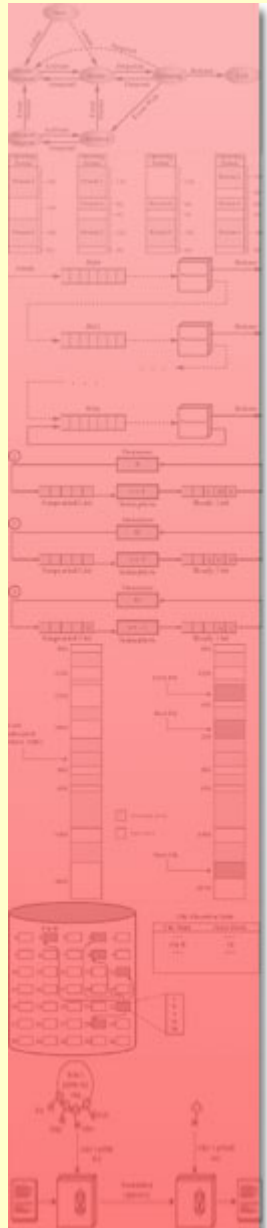


Figure 2.10 Virtual Memory Addressing



# Information Protection and Security

- Access control
  - regulate user access to the system
- Information flow control
  - regulate flow of data within the system and its delivery to users
- Certification
  - proving that access and flow control perform according to specifications



# Scheduling and Resource Management

- Fairness
  - give equal and fair access to all processes
- Differential responsiveness
  - discriminate between different classes of jobs
- Efficiency
  - maximize throughput, minimize response time, and accommodate as many uses as possible



# Major Elements of Operating System

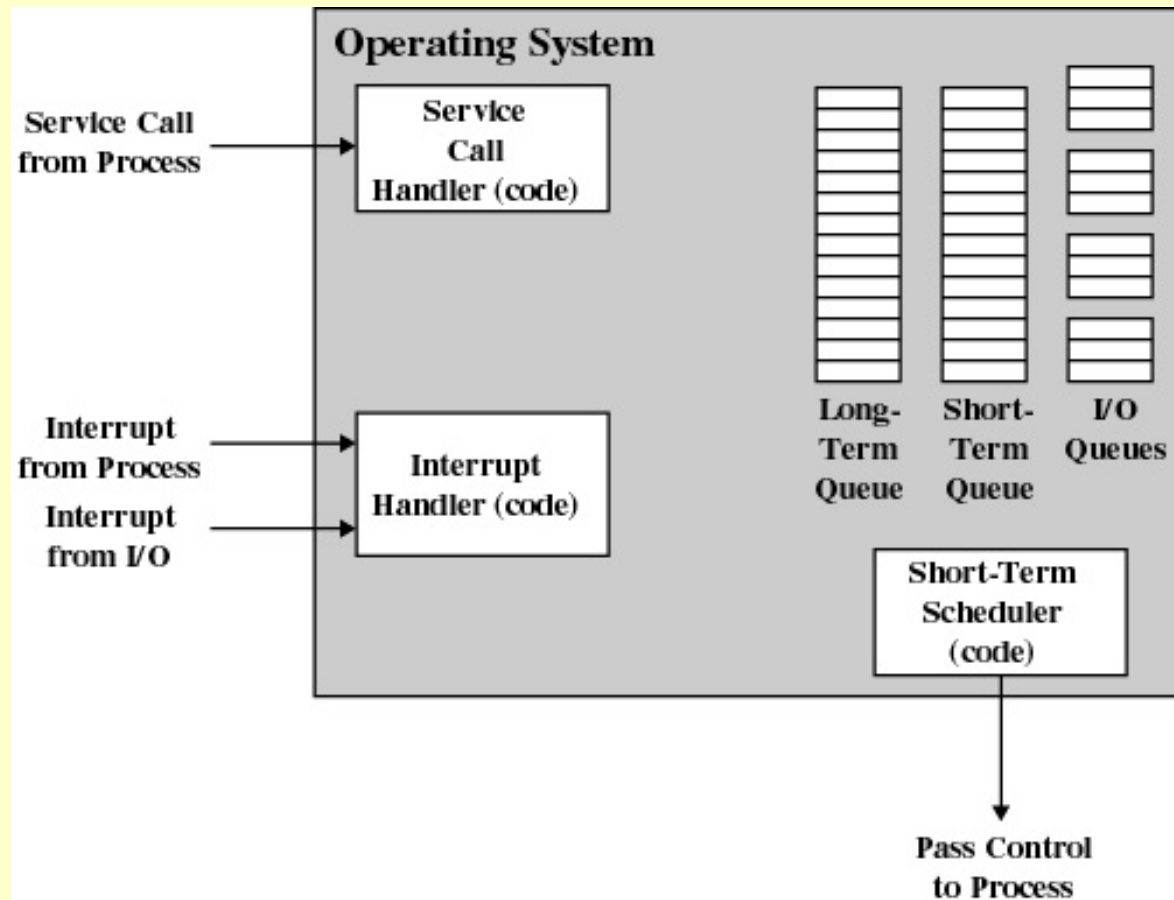
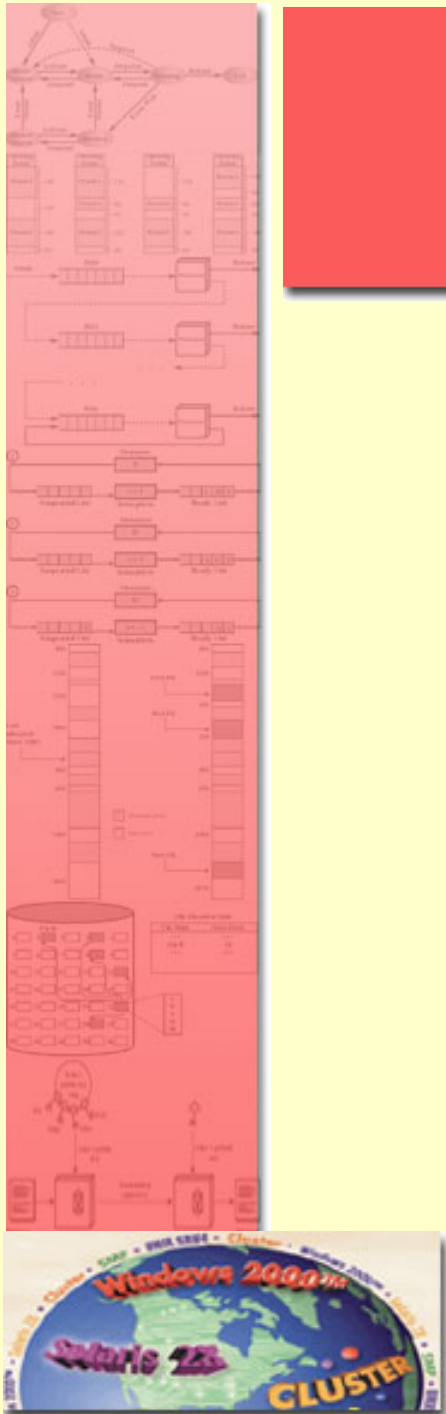


Figure 2.11 Key Elements of an Operating System for Multiprogramming



# System Structure

- View the system as a series of levels
- Each level performs a related subset of functions
- Each level relies on the next lower level to perform more primitive functions
- This decomposes a problem into a number of more manageable subproblems



# Operating System Design Hierarchy

Level	Name	Objects	Example Operations
13	Shell	User programming environment	Statements in shell language
12	User processes	User processes	Quit, kill, suspend, resume
11	Directories	Directories	Create, destroy, attach, detach, search, list
10	Devices	External devices, such as printer, displays and keyboards	Open, close, read, write
9	File system	Files	Create, destroy, open, close read, write
8	Communications	Pipes	Create, destroy, open. close, read, write





# Operating System Design Hierarchy

Level	Name	Objects	Example Operations
7	Virtual Memory	Segments, pages	Read, write, fetch
6	Local secondary store	Blocks of data, device channels	Read, write, allocate, free
5	Primitive processes	Primitive process, semaphores, ready list	Suspend, resume, wait, signal



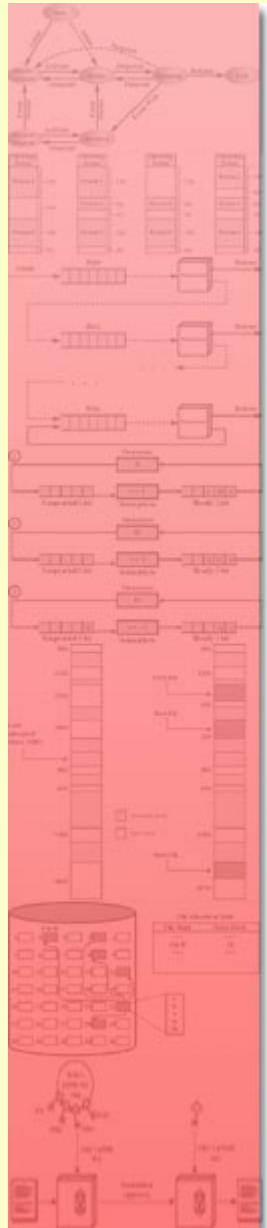
# Operating System Design Hierarchy

Level	Name	Objects	Example Operations
4	Interrupts retry programs	Interrupt-handling	Invoke, mask, unmask,
3	Procedures	Procedures, call stack, display	Mark stack, call, return
2	Instruction Set	Evaluation stack, micro- program interpreter, scalar and array data	Load, store, add, subtract branch
1	Electronic circuits	Registers, gates, buses, etc.	Clear, transfer, activate, complement



# Characteristics of Modern Operating Systems

- Microkernel architecture
  - assigns only a few essential functions to the kernel
    - address space
    - interprocess communication (IPC)
    - basic scheduling



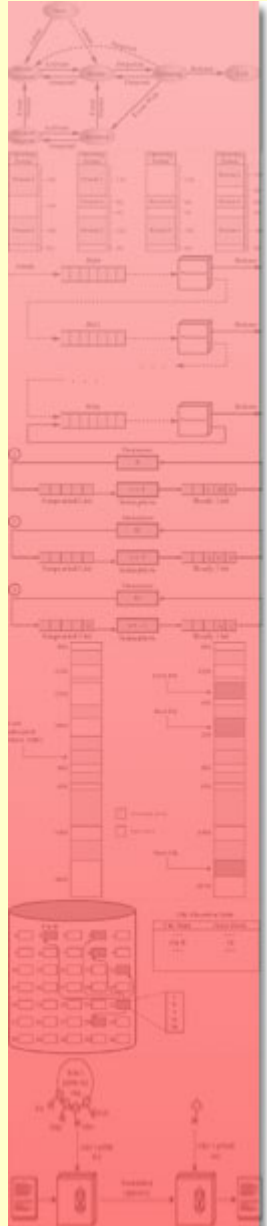
# Characteristics of Modern Operating Systems

- Multithreading
  - process is divided into threads that can run simultaneously
- Thread
  - dispatchable unit of work
  - executes sequentially and is interruptable
- Process is a collection of one or more threads



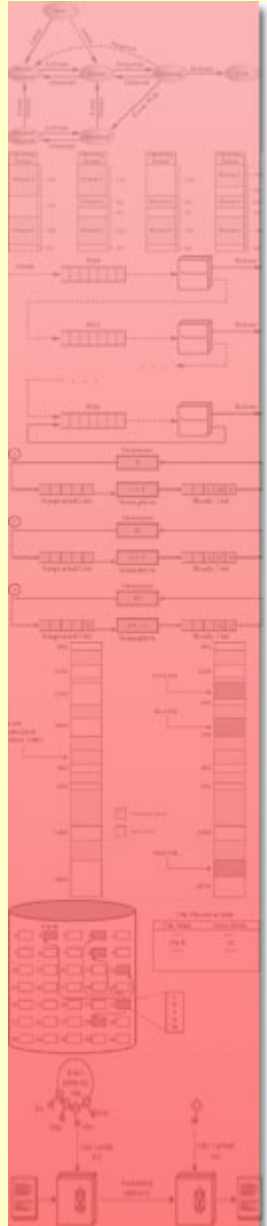
# Characteristics of Modern Operating Systems

- Symmetric multiprocessing
  - there are multiple processors
  - these processors share same main memory and I/O facilities
  - All processors can perform the same functions



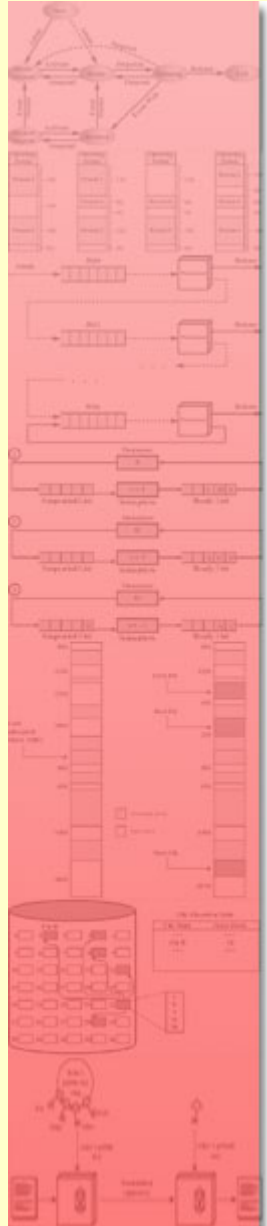
# Characteristics of Modern Operating Systems

- Distributed operating systems
  - provides the illusion of a single main memory and single secondary memory space
  - used for distributed file system



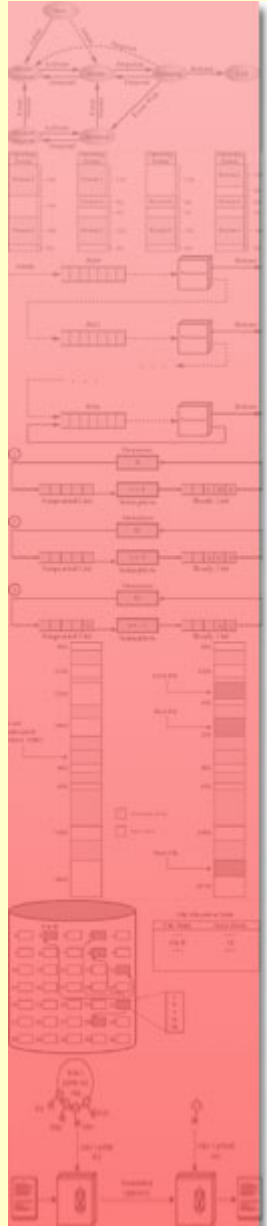
# Characteristics of Modern Operating Systems

- Object-oriented design
  - used for adding modular extensions to a small kernel
  - enables programmers to customize an operating system without disrupting system integrity



# Windows 2000

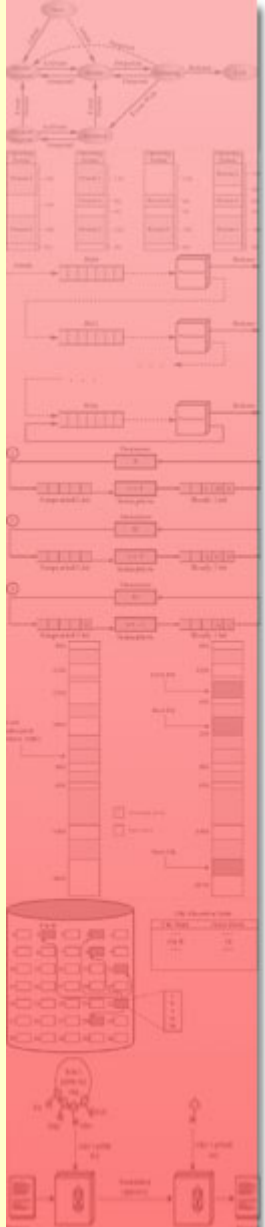
- Exploits the power of today's 32-bit microprocessors
- Provides full multitasking in a single-user environment
- Client/Server computing





# Windows 2000 Architecture

- Modular structure for flexibility
- Executes on a variety of hardware platforms
- Supports application written for a variety of other operating system



# OS Organization

- Modified microkernel architecture
  - Not a pure microkernel
  - Many system functions outside of the microkernel run in kernel mode
- Any module can be removed, upgraded, or replaced without rewriting the entire system



# Layered Structure

- Hardware abstraction layer (HAL)
  - Isolates the operating system from platform-specific hardware differences
- Microkernel
  - Most-used and most fundamental components of the operating system
- Device drivers
  - Translate user I/O function calls into specific hardware device I/O requests



# W2K Executive

- I/O manager
- Object manager
- Security reference monitor
- Process/thread manager
- Local procedure call (LPC) Facility
- Virtual memory manager
- Cache manager
- Windows/graphics modules



# User Processes

- Special system support processes
  - Ex: logon process and the session manager
- Server processes
- Environment subsystems
- User applications



# Client/Server Model

- Simplifies the Executive
  - possible to construct a variety of APIs
- Improves reliability
  - each service runs as a separate process with its own partition of memory
  - clients cannot not directly access hardware
- Provides a uniform means for applications to communicate via LPC
- Provides base for distributed computing



# Threads and SMP

- Different routines can execute simultaneously on different processors
- Multiple threads of execution within a single process may execute on different processors simultaneously
- Server processes may use multiple threads
- Share data and resources between process



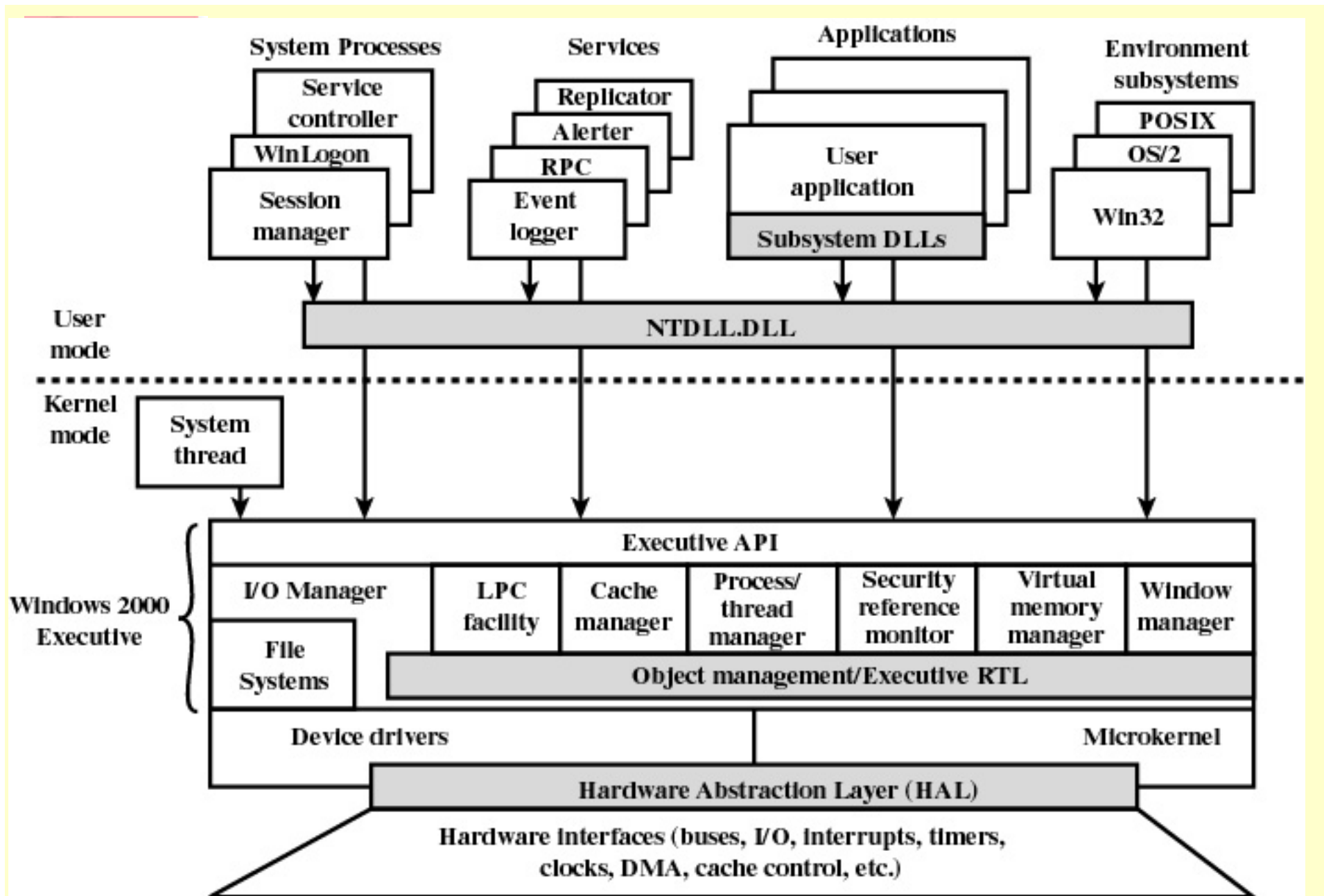


Figure 2.13 Windows 2000 Architecture



# UNIX

- Hardware is surrounded by the operating-system
- Operating system is called the kernel
- Comes with a number of user services and interfaces
  - shell
  - C compiler



# UNIX

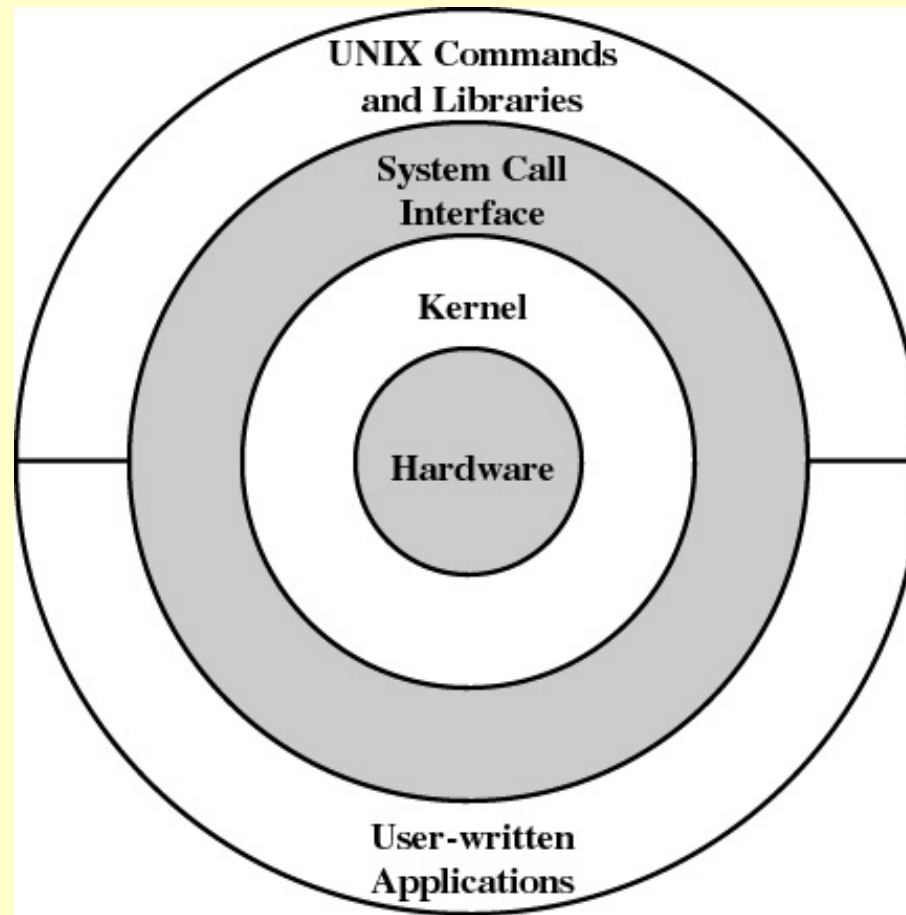
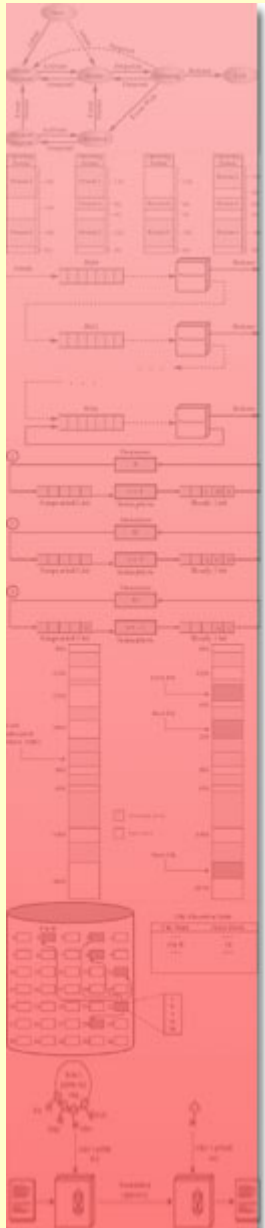
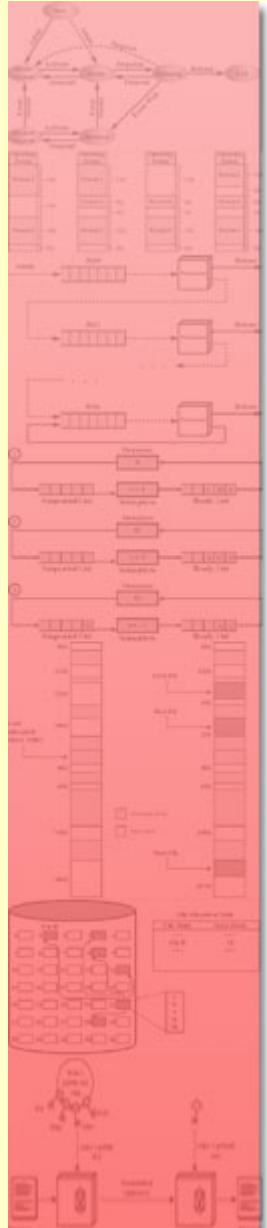


Figure 2.15 General UNIX Architecture



# Modern UNIX Systems

- System V Release 4 (SVR4)
- Solaris 2.x
- 4.4BSD
- Linux



Dr. Garpis Aristogiannis - EPDO  
TEI Messolonghi