Fontys UNIVERSITY OF APPLIED SCIENCES › FOR SOCIETY

Chapter 2 – Operating system overview

Program

1 Computer System Overview

2 Operating System Overview

3 Process Description and Control

4 Threads

5 Concurrency: Mutual exclusion and Sync

6 Concurrency: Deadlock and Starvation

7 Spare

8 Memory Management

9 Virtual Memory

10 Uniprocessor scheduling

11 I/O Management and Disk Scheduling

12 Holiday

13 File Management

Roadmap

- **Operating System Objectives/Functions**
- The Evolution of Operating Systems
- Major Achievements
- Developments Leading to Modern Operating Systems
- Microsoft Windows Overview
- UNIX Systems
- Linux
- Android

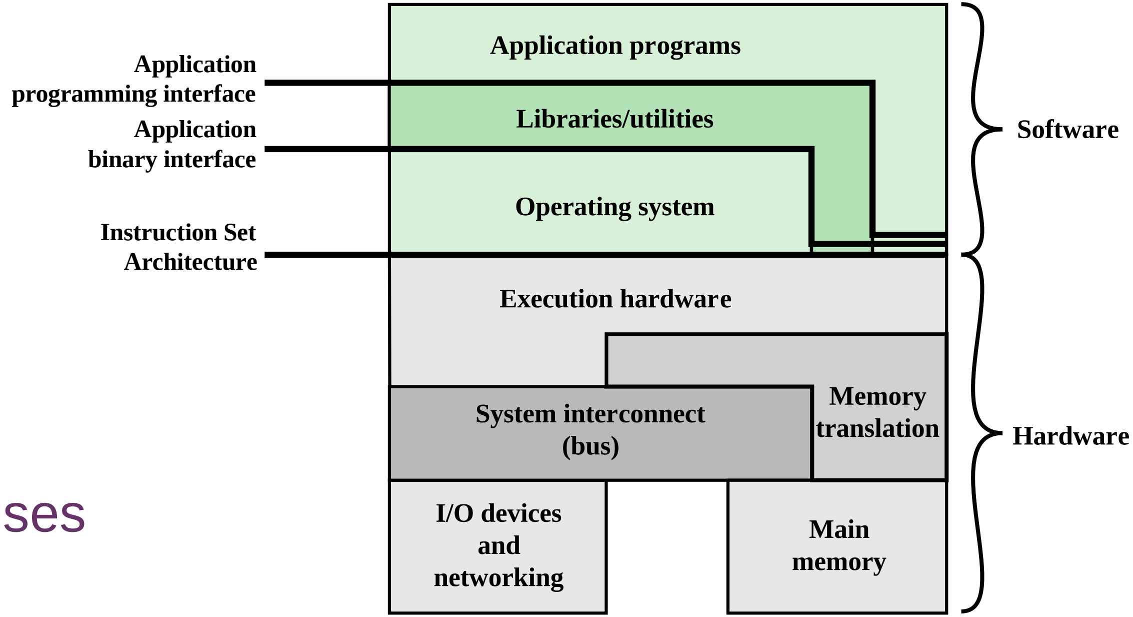
What are the functions of an OS?

Operating System

- A program that controls the execution of application programs
- An interface between applications and hardware
- Main objectives of an OS:
  - Convenience
  - Efficiency
  - Ability to evolve

Services and Layers

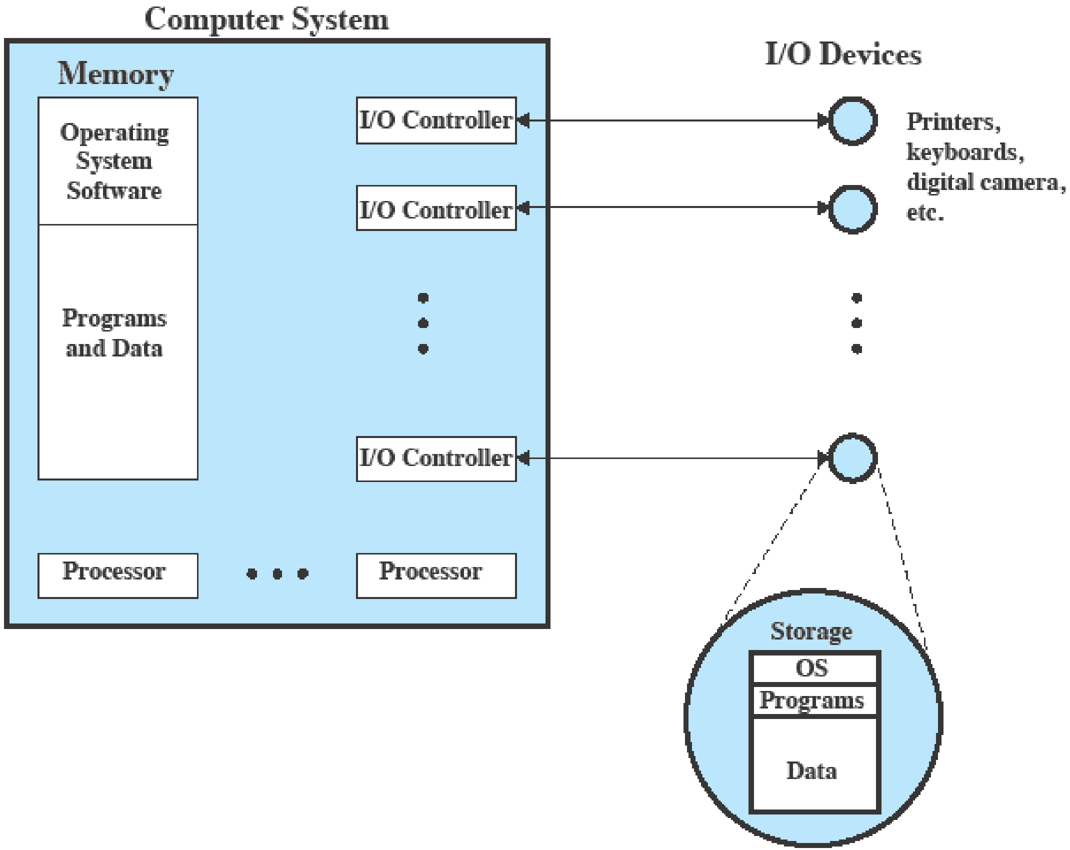
- Program development
- Program execution
- Access I/O devices
- Controled access to files
- System Access
- Error detection and responses
- Accounting



Application programming interface
Application binary interface
Instruction Set Architecture
Application programs
Libraries/utilities
Operating system
Execution hardware
System interconnect (bus)
Memory translation
I/O devices and networking
Main memory
Software
Hardware

Figure 2.1 Computer Hardware and Software Structure

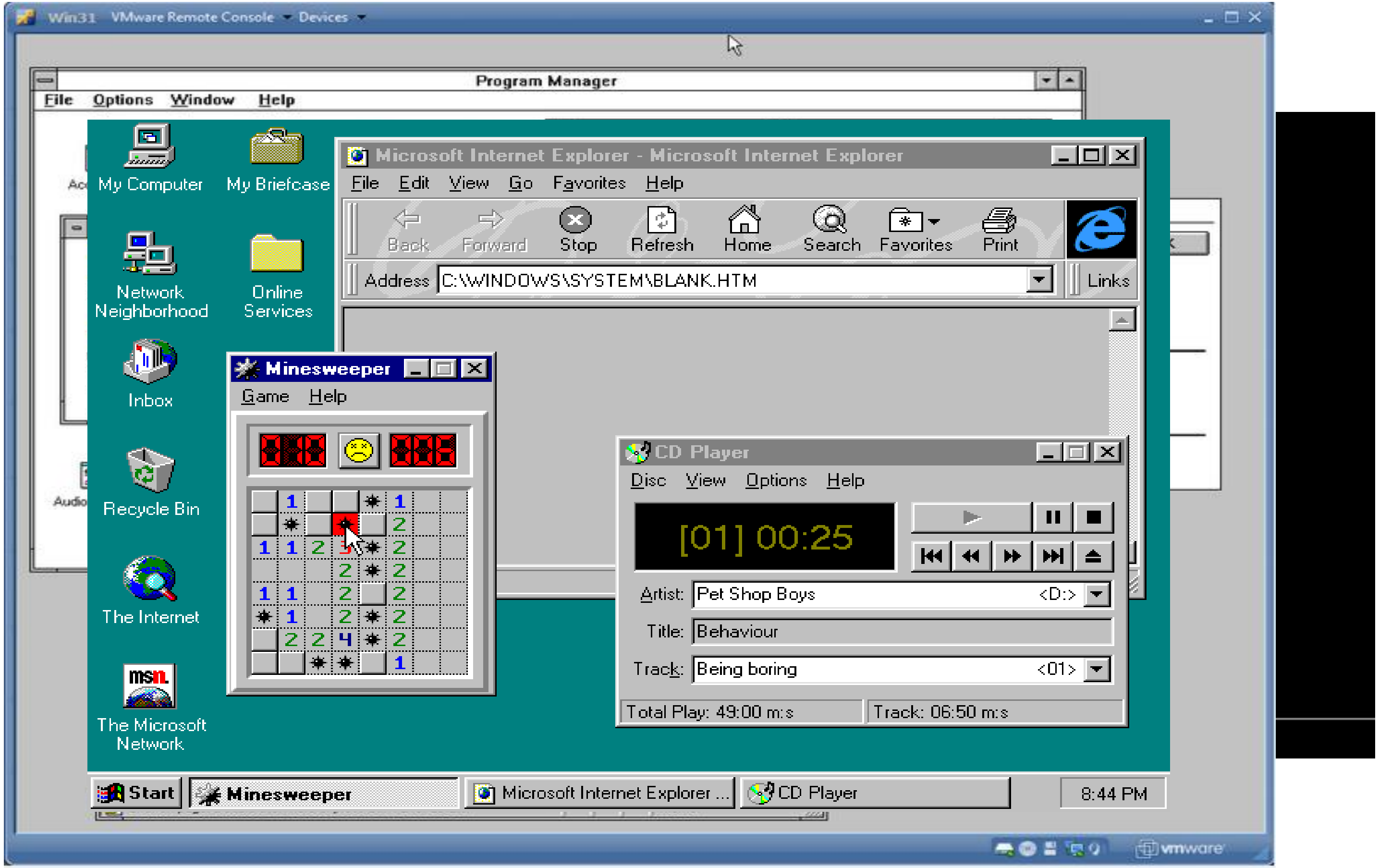
OS as resource manager



Roadmap

- Operating System Objectives/Functions
- **The Evolution of Operating Systems**
- Major Achievements
- Developments Leading to Modern Operating Systems
- Microsoft Windows Overview
- UNIX Systems
- Linux
- Android

DOS b



Address: C:\WINDOWS\SYSTEM\BLANK.HTM

[01] 00:25

Artist: Pet Shop Boys <D:>

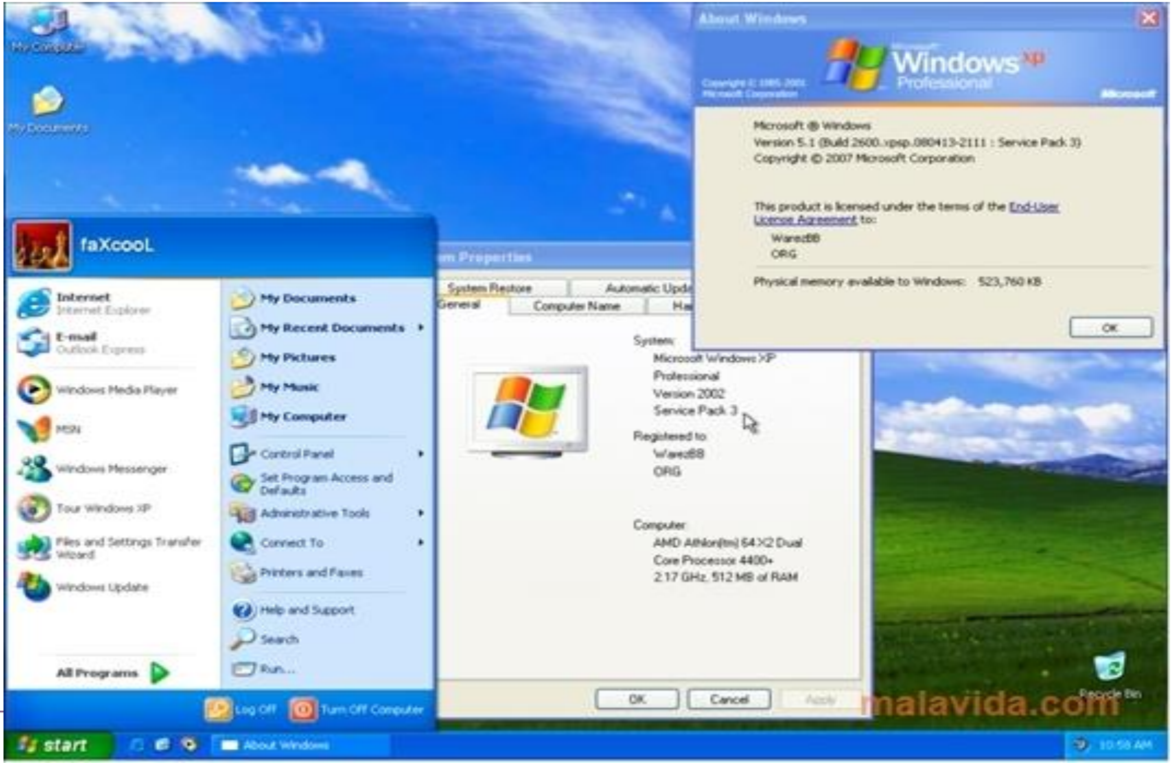
Title: Behaviour

Track: Being boring <01>

Total Play: 49:00 m:s

Track: 06:50 m:s

Windows XP



Microsoft ® Windows
Version 5.1 (Build 2600.xpsp.080413-2111 : Service Pack 3)
Copyright © 2007 Microsoft Corporation

This product is licensed under the terms of the End-User License Agreement to:
WarezBB
ORG

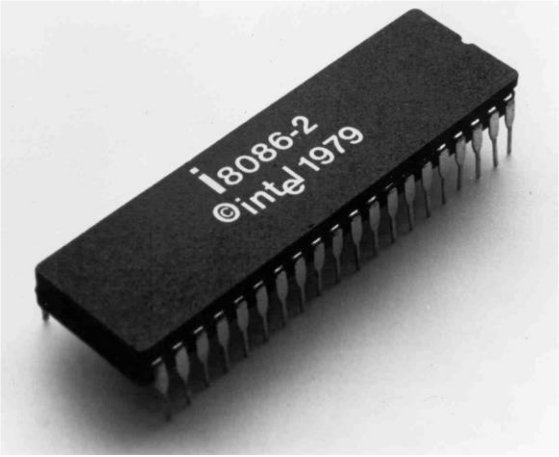
Physical memory available to Windows: 523,760 KB

System:
Microsoft Windows XP
Professional
Version 2002
Service Pack 3

Registered to:
WarezBB
ORG

Computer:
AMD Athlon(tm) 64 X2 Dual
Core Processor 4400+
2.17 GHz, 512 MB of RAM

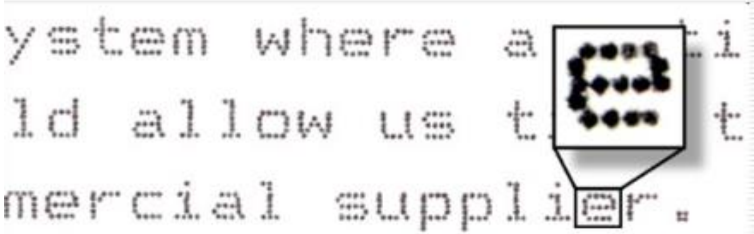
Why do OS-es evolve?



Dot matrix printer

http://nyomtatokszeged.hu/popup_image.php?pID=208

http://richardlagendijk.nl/foto/cip/printer_1550c_01.jpg



http://blog.[illegible].com/wp-
content/uploads/[illegible]

› FOR SOCIETY

The evolution of:
Operating Systems

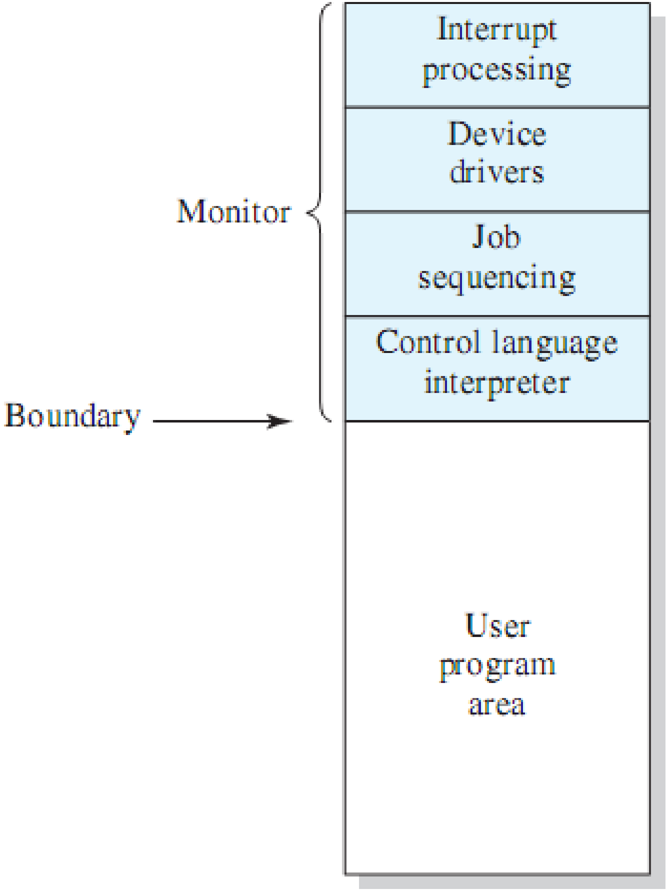
Serial Processing

- No operating system
- Machines run from a console with display lights, toggle switches, input device, and printer
- Problems include:
  - Scheduling
  - Setup time

Simple batch system

Monitor’s perspective

- Monitor controls the sequence of events
- *Resident Monitor* is software always in memory
- Monitor reads in job and gives control
- Job returns control to monitor



Modes of operation

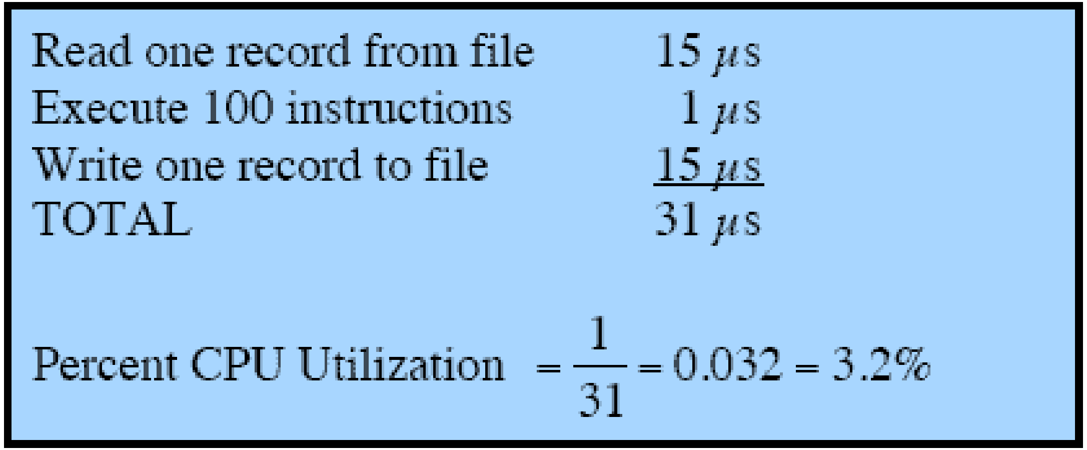
- **User mode**
  - User program executes in user mode
  - Certain areas of memory are protected from user access
  - Certain instructions may not be executed

- **Kernel mode**
  - Monitor executes in kernel mode
  - Privileged instructions may be executed
  - Protected areas of memory may be accessed

Utilisation batch systems

The road to multiprogrammed batch systems

- CPU is often idle ("single process")

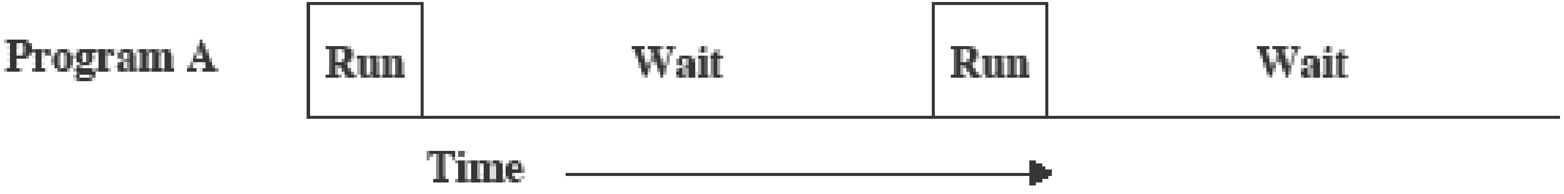


Read one record from file	15 μs
Execute 100 instructions	1 μs
Write one record to file	15 μs
TOTAL	31 μs

$$\text{Percent CPU Utilization} = \frac{1}{31} = 0.032 = 3.2\%$$

(Batched) Uniprogramming

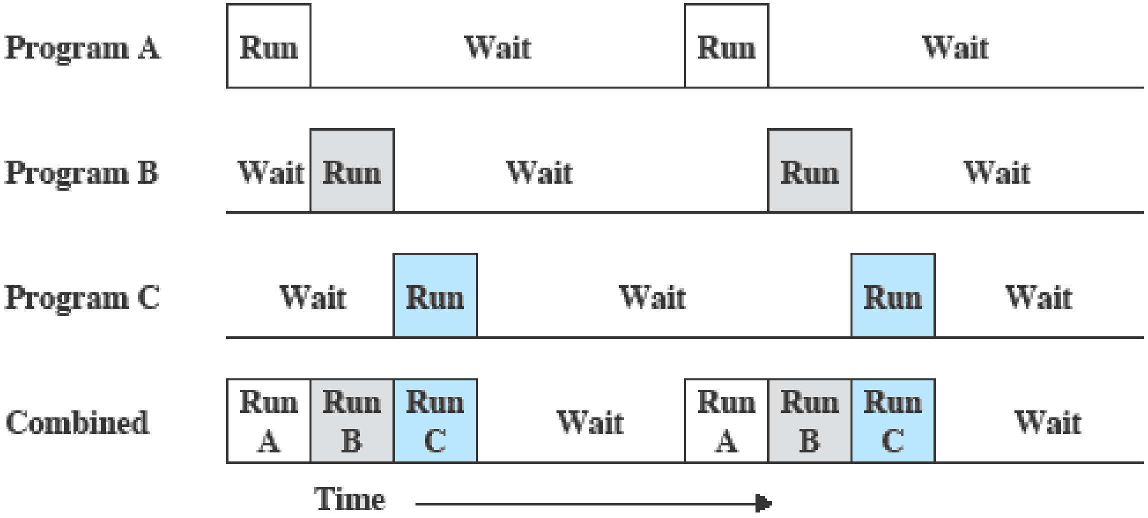
- Processor must wait for I/O instruction to complete before preceding



(a) Uniprogramming

(Batched) Multiprogramming with one processor

- When one job needs to wait for I/O, the processor can switch to another job, and so on



Program A: Run | Wait | Run | Wait

Program B: Wait | Run | Wait | Run | Wait

Program C: Wait | Run | Wait | Run | Wait

Combined: Run A | Run B | Run C | Wait | Run A | Run B | Run C | Wait

Time →

(c) Multiprogramming with three programs

Example

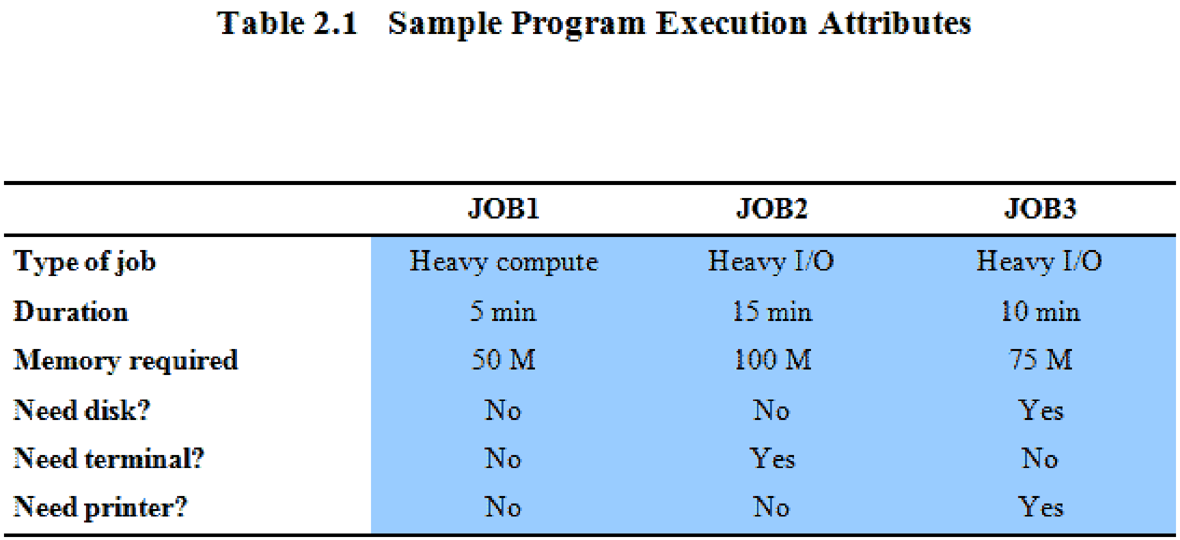
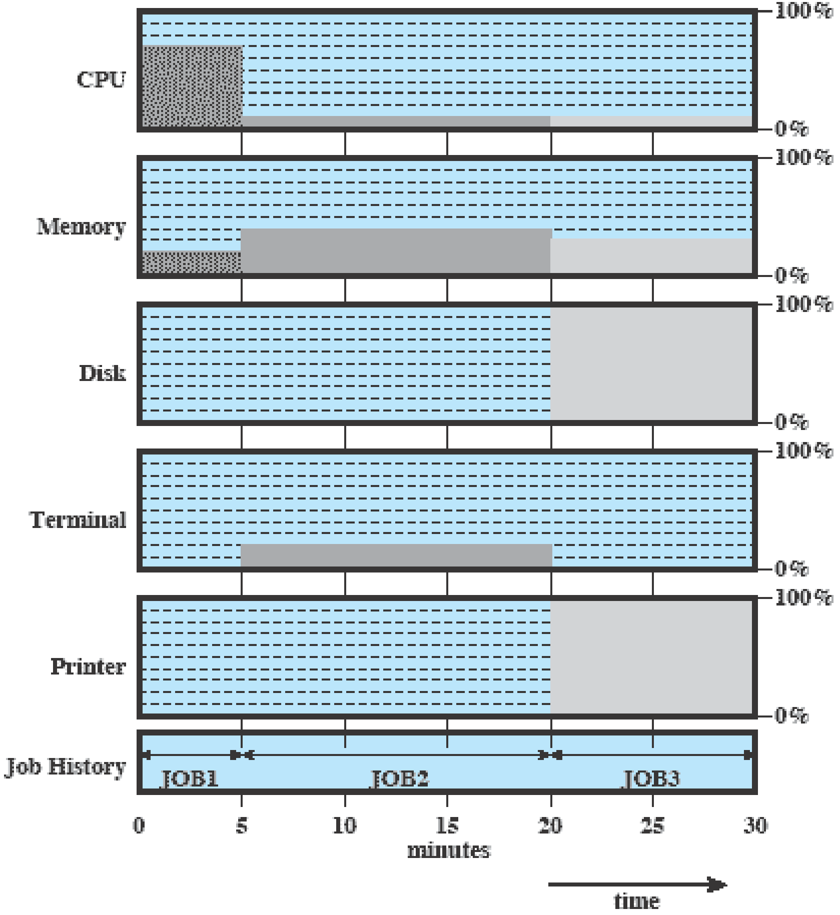


Table 2.1 Sample Program Execution Attributes

	JOB1	JOB2	JOB3
Type of job	Heavy compute	Heavy I/O	Heavy I/O
Duration	5 min	15 min	10 min
Memory required	50 M	100 M	75 M
Need disk?	No	No	Yes
Need terminal?	No	Yes	No
Need printer?	No	No	Yes



(a) Uniprogramming

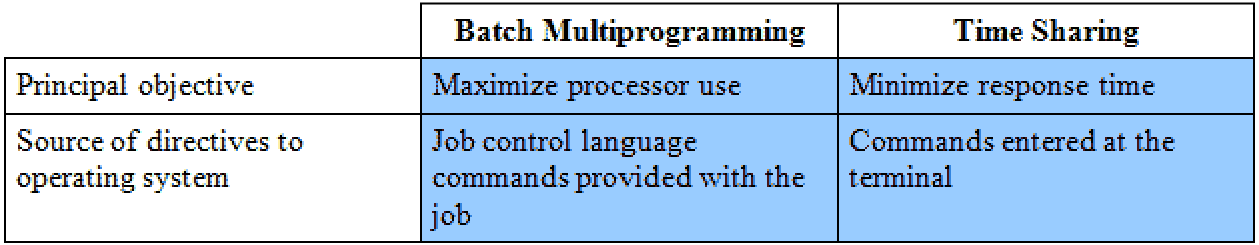
Time Sharing Systems

- Multiprogramming
- Multiple users
- Use of terminals

Computers were expensive

Batch Multiprogramming vs. Time Sharing

Table 2.3 Batch Multiprogramming versus Time Sharing



	Batch Multiprogramming	Time Sharing
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

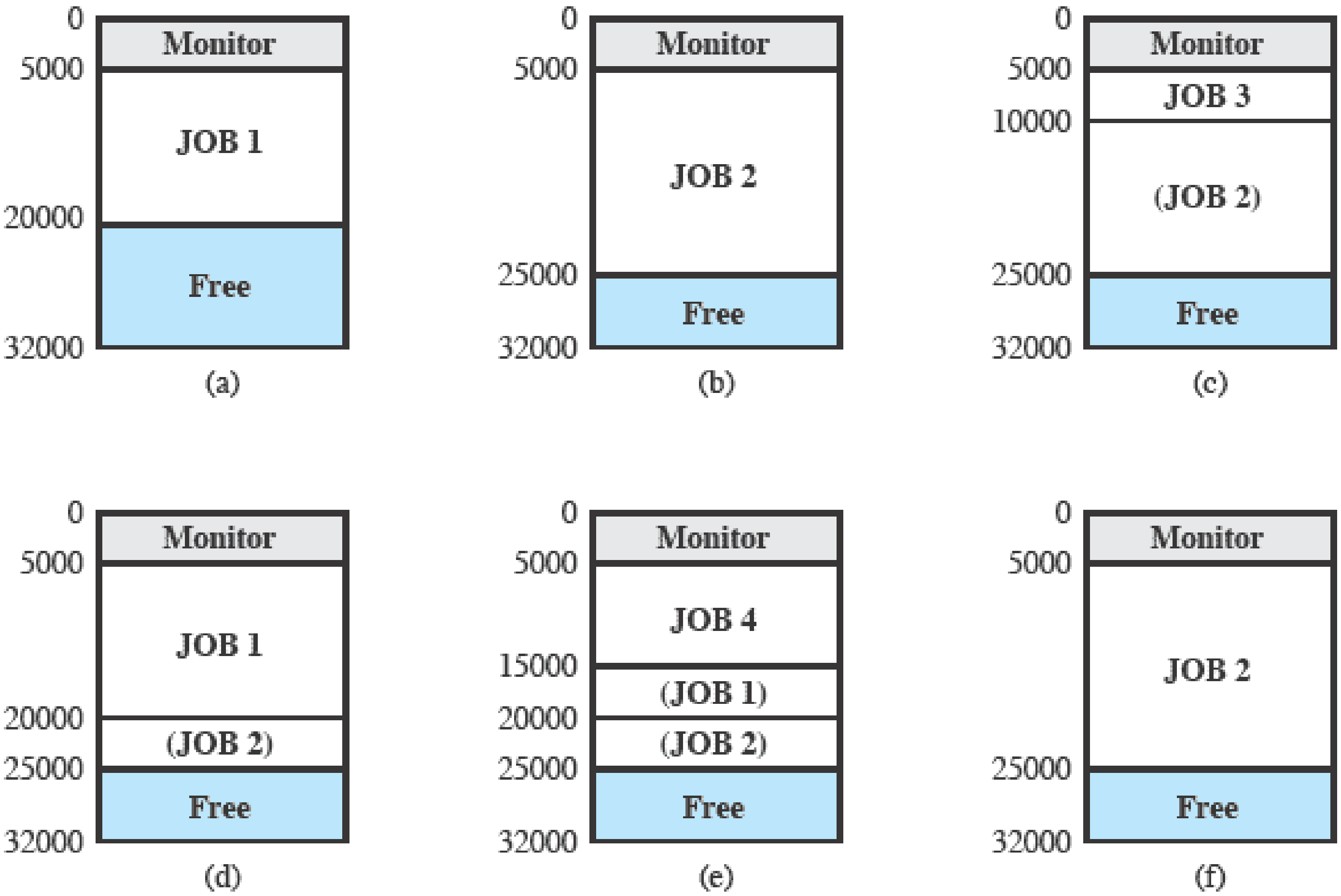
IBM’s dream?

CTSS – Compatible Time-Sharing System

- Time Slicing:
  - Control passed to user
  - User program and data loaded
  - Clock generates interrupts every 0.2 sec
  - Interrupt: OS gained control

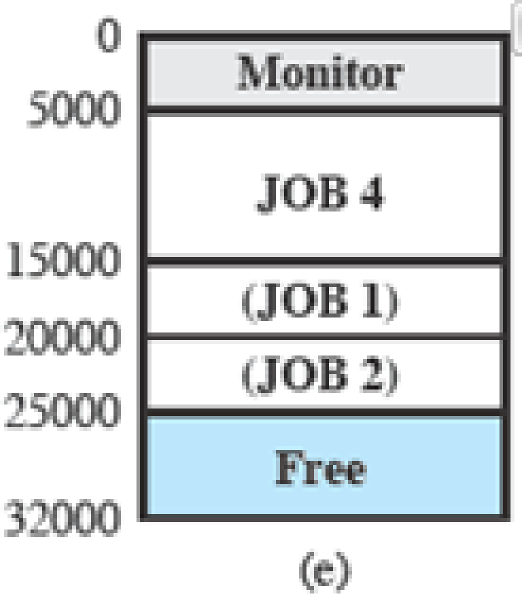
OWN IT!

CTS


0
Monitor
5000
JOB 1
20000
Free
32000
(a)
0
Monitor
5000
JOB 2
25000
Free
32000
(b)
0
Monitor
5000
JOB 3
10000
(JOB 2)
25000
Free
32000
(c)
0
Monitor
5000
JOB 1
20000
(JOB 2)
25000
Free
32000
(d)
0
Monitor
5000
JOB 4
15000
(JOB 1)
20000
(JOB 2)
25000
Free
32000
(e)
0
Monitor
5000
JOB 2
25000
Free
32000
(f)

Problems and issues when using time-sharing

- Memory protection
- Files system protection
- Other resource access protection



Roadmap

- Operating System Objectives/Functions
- The Evolution of Operating Systems
- **Major Achievements**
- Developments Leading to Modern Operating Systems
- Microsoft Windows Overview
- UNIX Systems
- Linux
- Android

Major achievements:

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

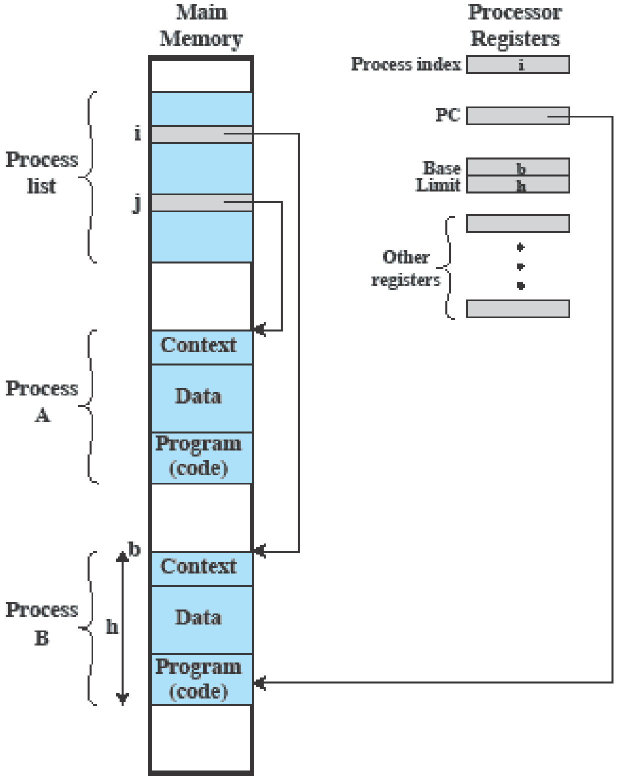
Process

Fundamental to the structure of OS’s

A *process* is:

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

Process Management



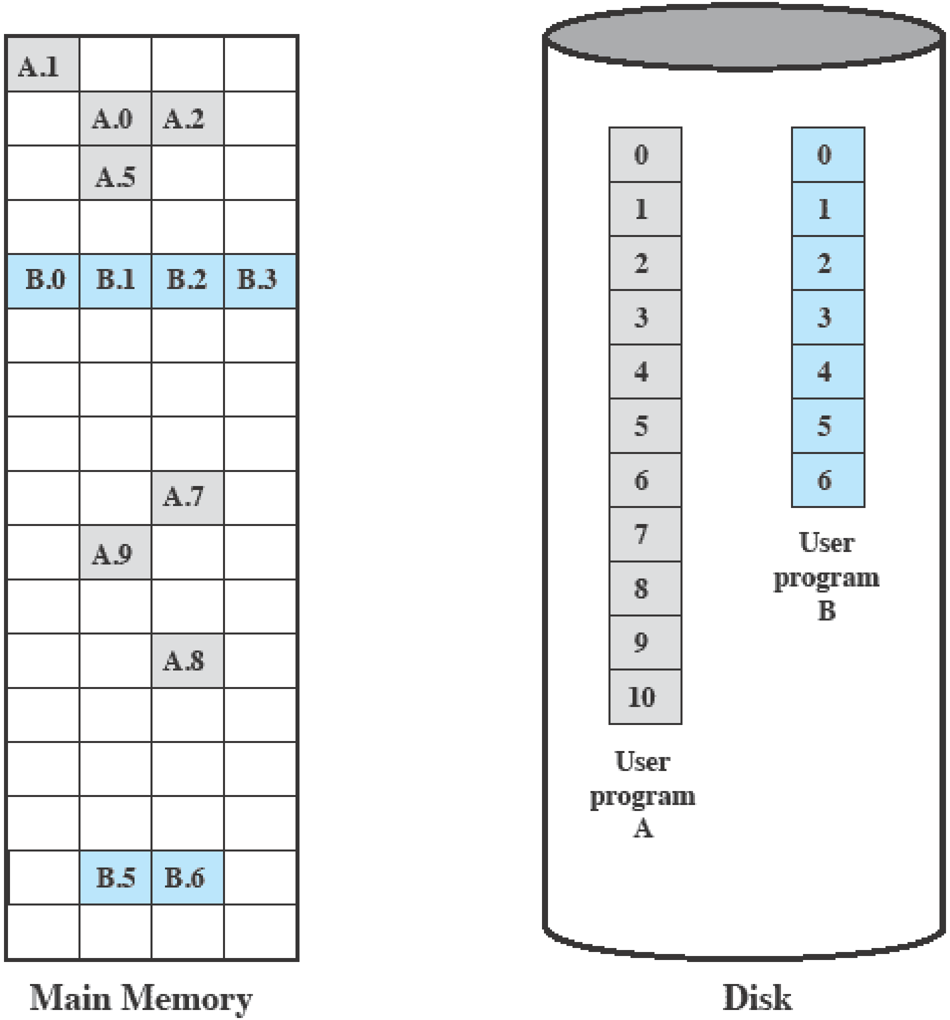
Memory Management

- The OS has 5 principal storage management responsibilities
  - Process isolation
  - Automatic allocation and management
  - Support of modular programming
  - Protection and access control
  - Long-term storage

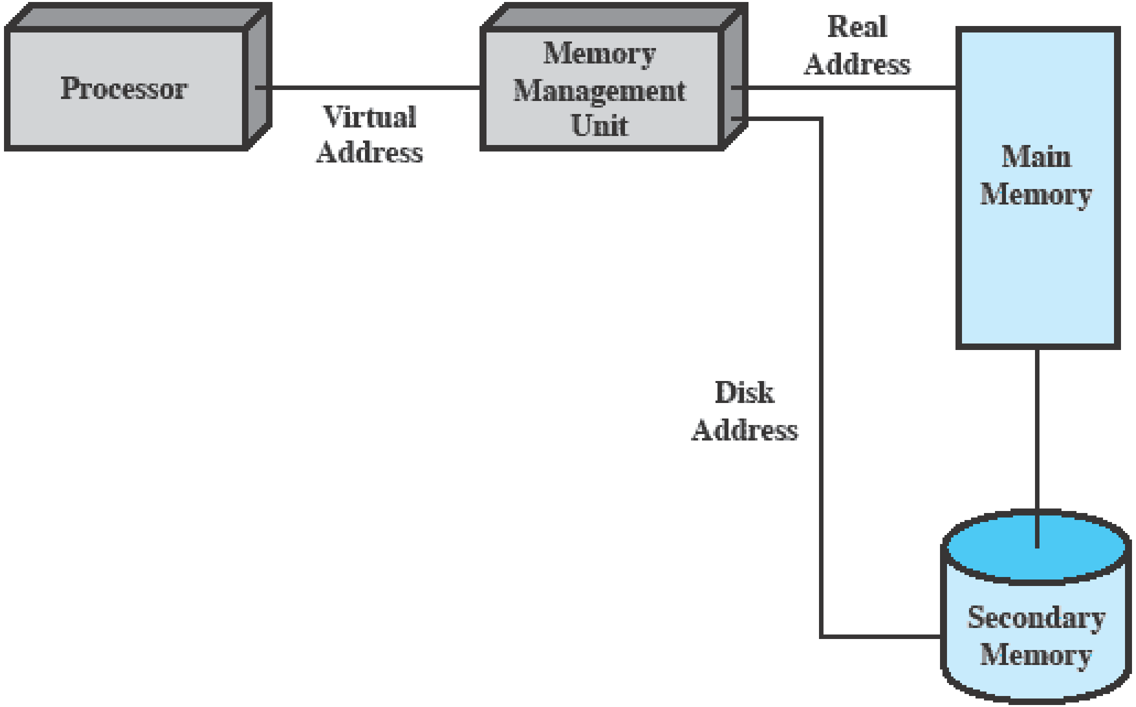
How to solve the physical memory boundary?

Virtual Memory

Paging



Virtual Memory Addressing



What order are the different processes allowed control of the processor?

Key Elements of an Operating System

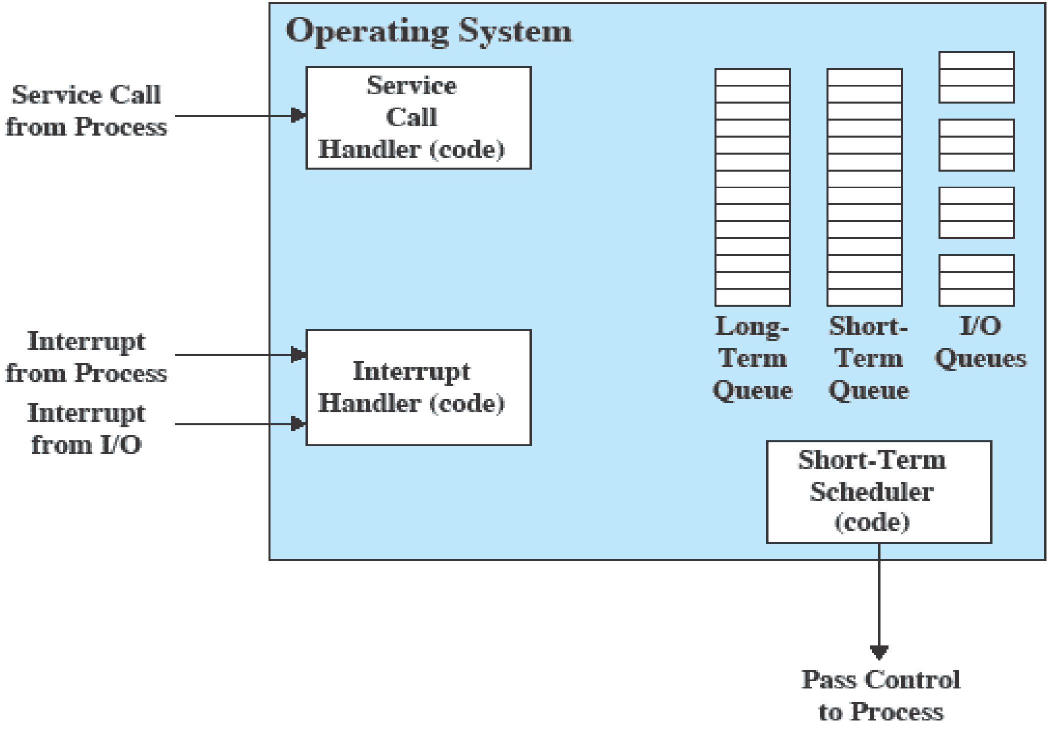


Figure 2.11 Key Elements of an Operating System for Multiprogramming

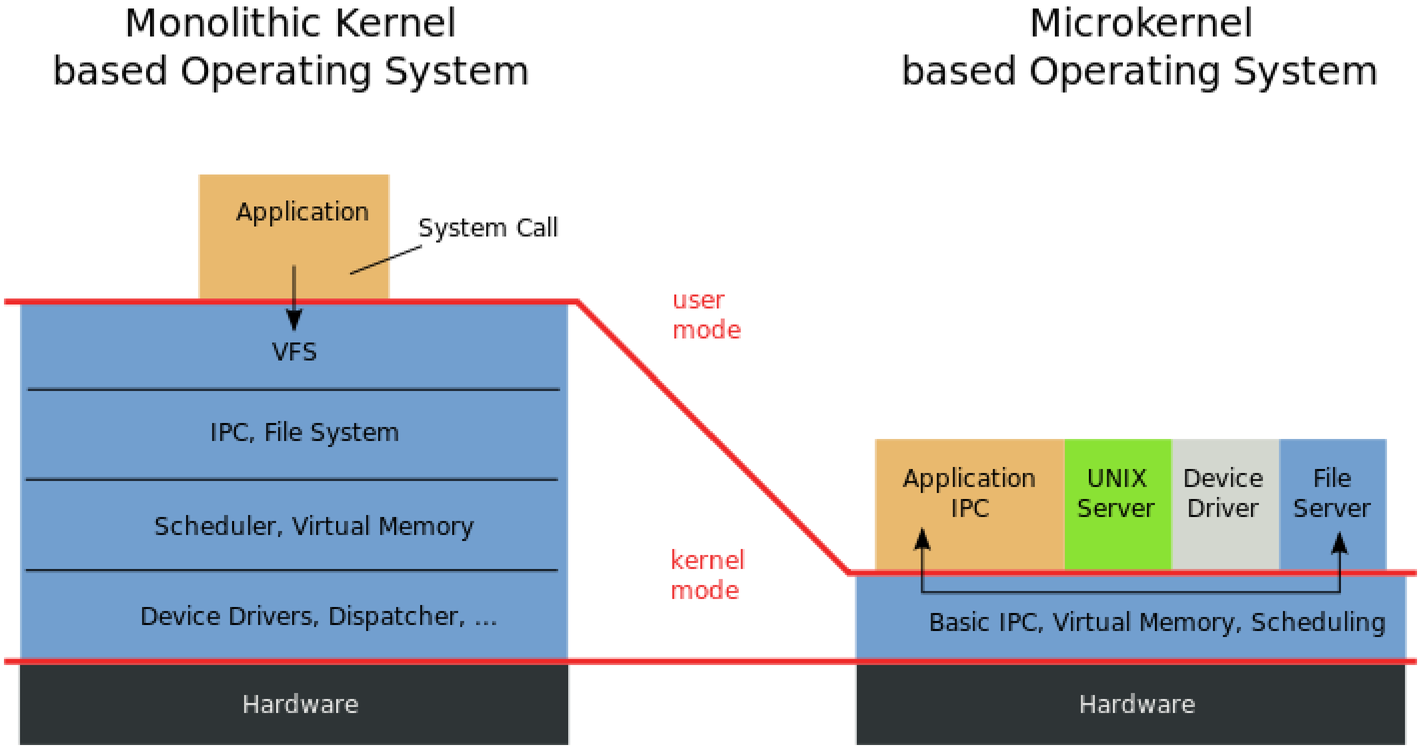
Roadmap

- Operating System Objectives/Functions
- The Evolution of Operating Systems
- Major Achievements
- **Developments Leading to Modern Operating Systems**
- Microsoft Windows Overview
- UNIX Systems
- Linux
- Android

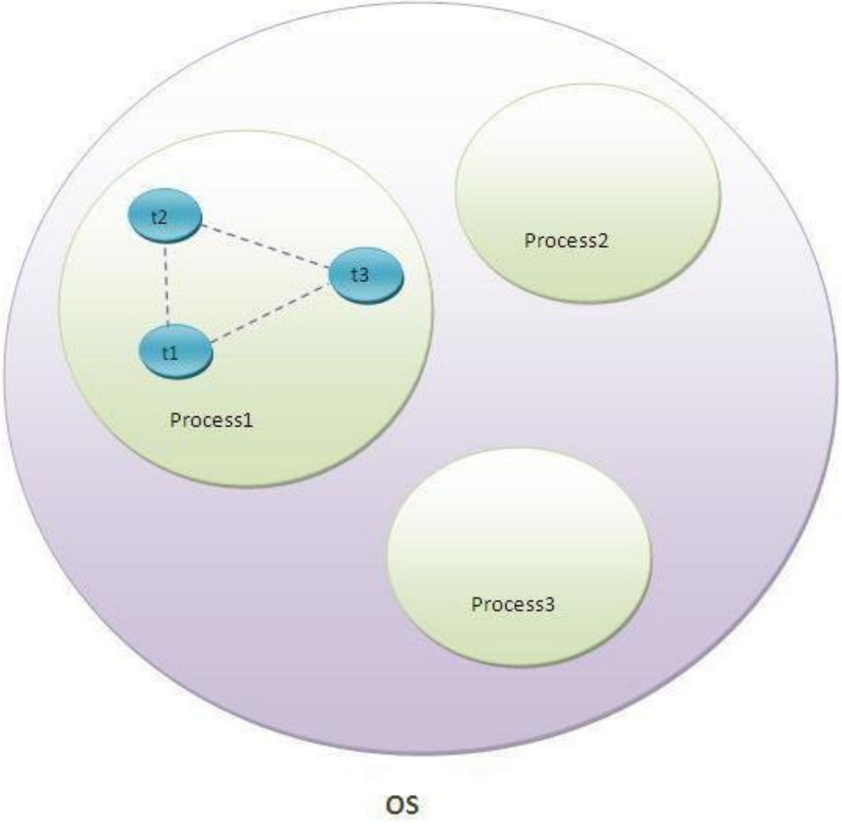
Different Architectural Approaches

- Various approaches have been tried, categories include:
  - Microkernel architecture
  - Multithreading
  - Symmetric multiprocessing
  - Distributed operating systems
  - Object-oriented design

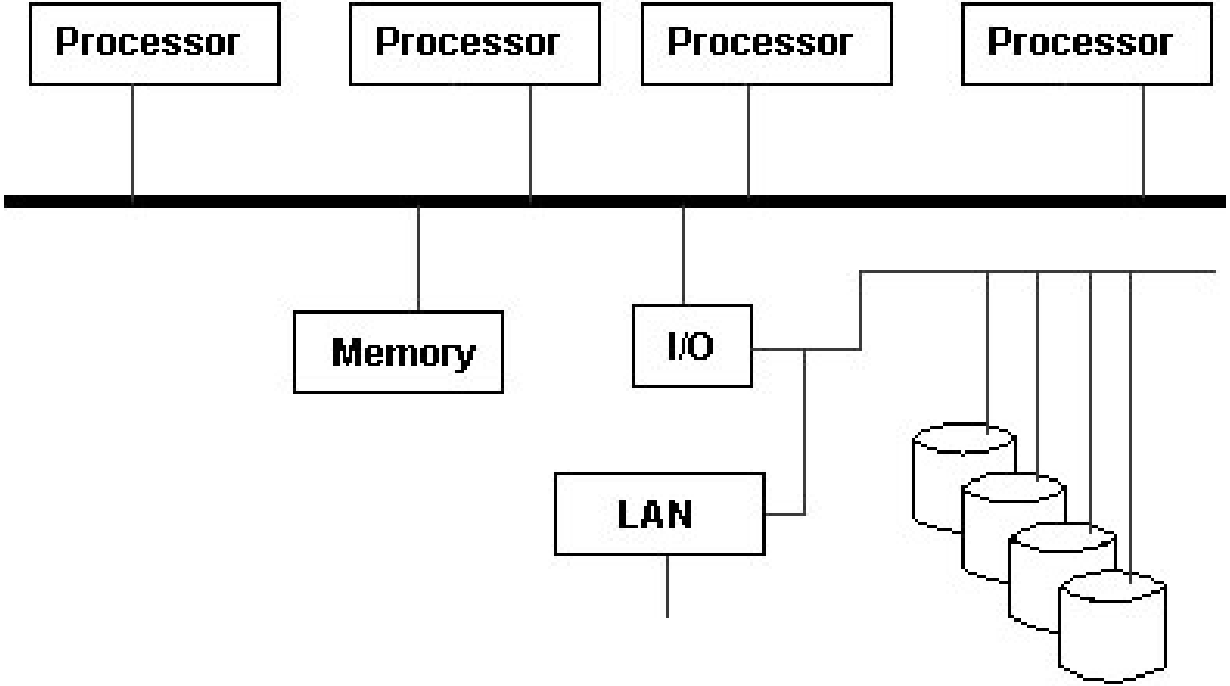
Monolithic or Micro kernel?



Processes and Threads (multi threading)



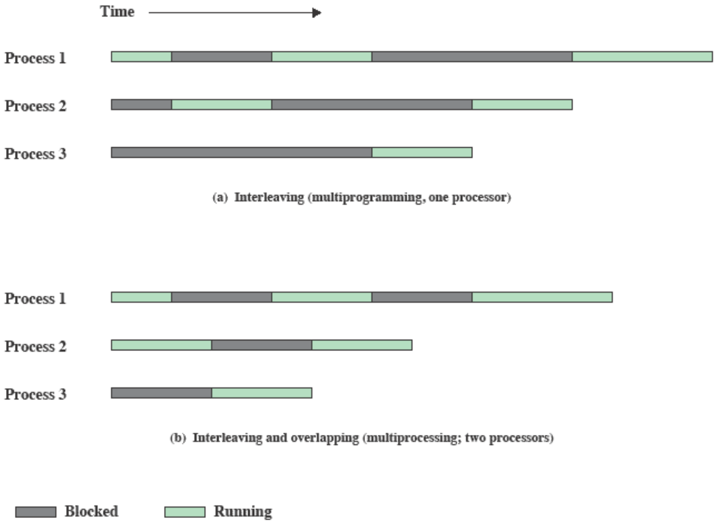
Symetric Multi Processing (SMP)



SMP Advantages

- Performance
  - Allowing parallel processing
- Availability
  - Failure of a single process does not halt the system
- Incremental Growth
  - Additional processors can be added.
- Scaling

Multiprogramming and Multiprocessing



(a) Interleaving (multiprogramming, one processor)

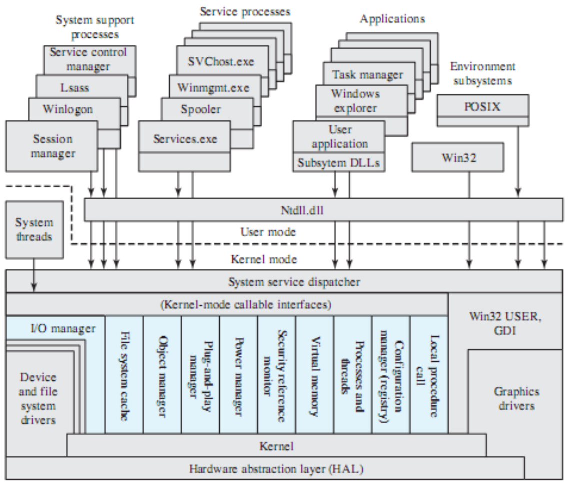
(b) Interleaving and overlapping (multiprocessing; two processors)

Figure 2.12 Multiprogramming and Multiprocessing

Roadmap

- Operating System Objectives/Functions
- The Evolution of Operating Systems
- Major Achievements
- Developments Leading to Modern Operating Systems
- **Microsoft Windows Overview**
- UNIX Systems
- Linux
- Android

Windows Architecture



Lsass = local security authentication server
POSIX = portable operating system interface
GDI = graphics device interface
DLL = dynamic link libraries

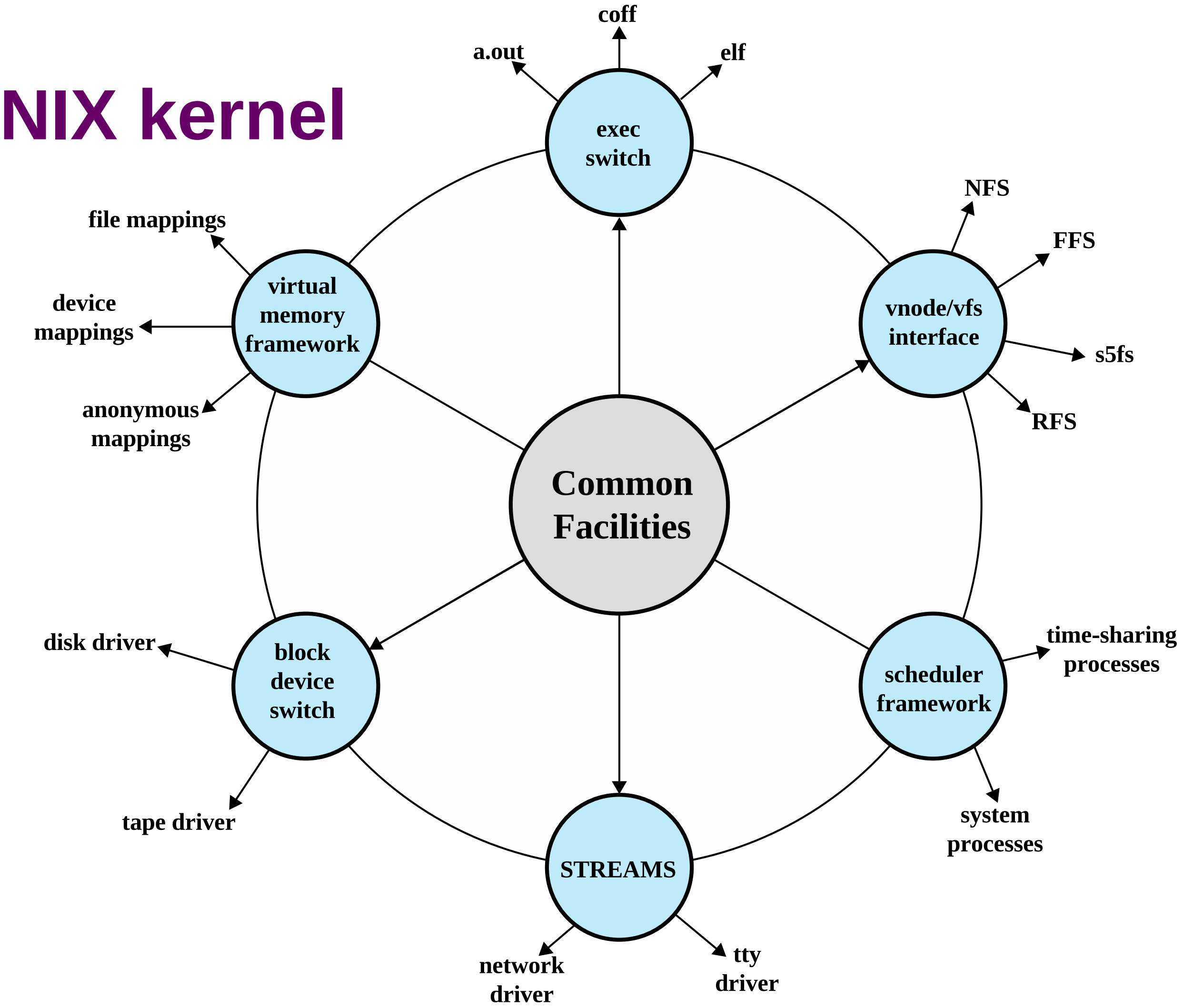
Colored area indicates Executive

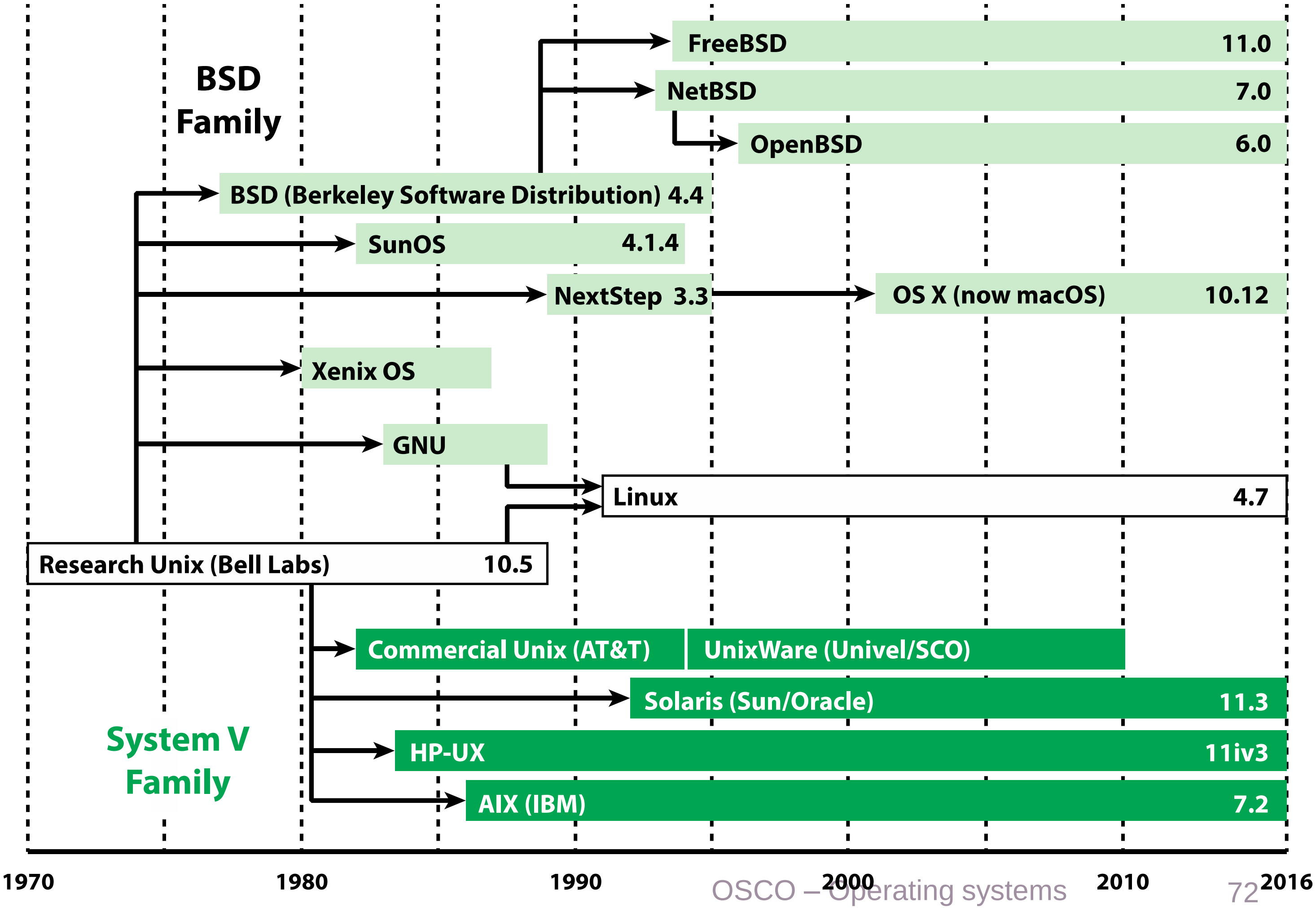
Figure 2.13 **Windows and Windows Vista Architecture [RUSS05]**

Roadmap

- Operating System Objectives/Functions
- The Evolution of Operating Systems
- Major Achievements
- Developments Leading to Modern Operating Systems
- Microsoft Windows Overview
- **UNIX Systems**
- Linux
- Android

Modern UNIX kernel




BSD
Family
FreeBSD
11.0
NetBSD
7.0
OpenBSD
6.0
BSD (Berkeley Software Distribution) 4.4
SunOS
4.1.4
NextStep 3.3
OS X (now macOS)
10.12
Xenix OS
GNU
Linux
4.7
Research Unix (Bell Labs)
10.5
Commercial Unix (AT&T)
UnixWare (Univel/SCO)
Solaris (Sun/Oracle)
11.3
System V
Family
HP-UX
11iv3
AIX (IBM)
7.2
1970
1980
1990
2000
2010
2016

Roadmap

- Operating System Objectives/Functions
- The Evolution of Operating Systems
- Major Achievements
- Developments Leading to Modern Operating Systems
- Microsoft Windows Overview
- UNIX Systems
- **Linux**
- Android

Modular Structure

- Although monolithic, the kernel is structured as a collection of modules
  - Loadable modules
  - An object file which can be linked and unlinked at run time
- Characteristics:
  - Dynamic Linking
  - Stackable modules

Linux Kernel Modules

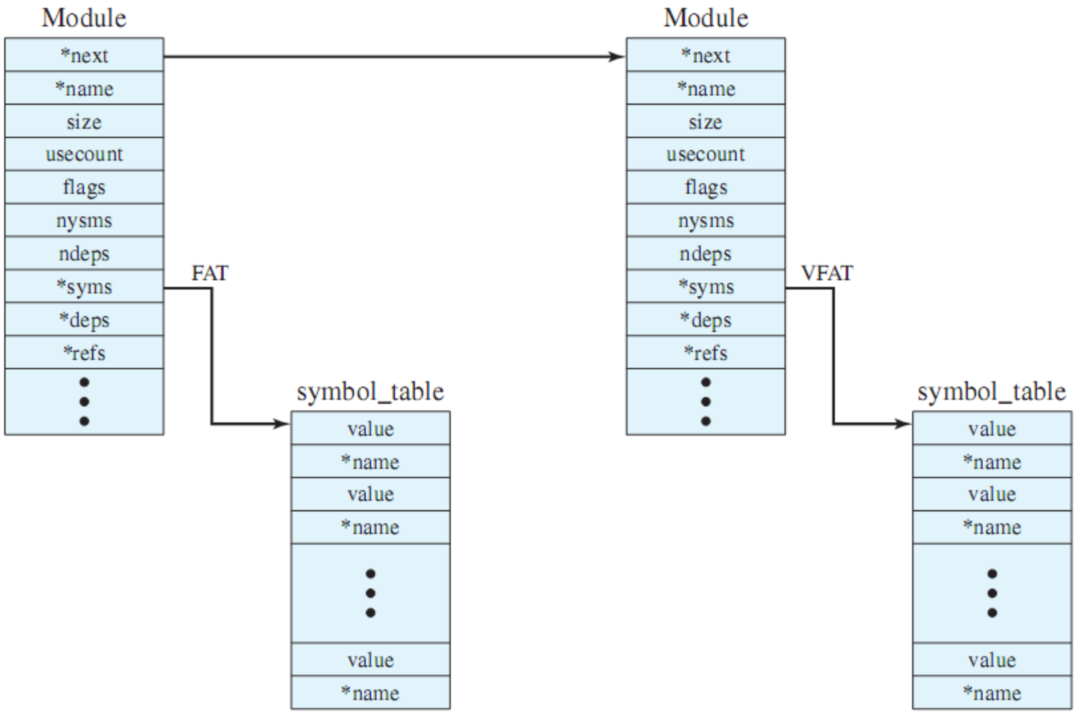
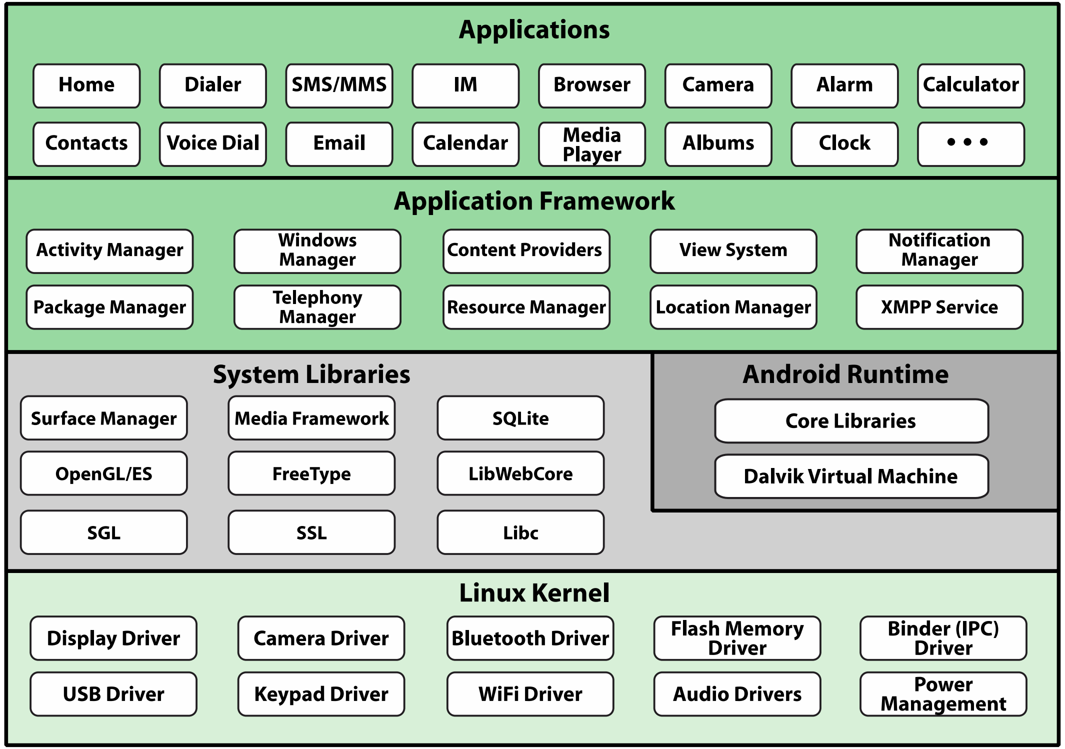


Figure 2.17 **Example List of Linux Kernel Modules**

Roadmap

- Operating System Objectives/Functions
- The Evolution of Operating Systems
- Major Achievements
- Developments Leading to Modern Operating Systems
- Microsoft Windows Overview
- UNIX Systems
- Linux
- **Android**

Android



Applications

Home, Dialer, SMS/MMS, IM, Browser, Camera, Alarm, Calculator

Contacts, Voice Dial, Email, Calendar, Media Player, Albums, Clock, • • •

Application Framework

Activity Manager, Windows Manager, Content Providers, View System, Notification Manager

Package Manager, Telephony Manager, Resource Manager, Location Manager, XMPP Service

System Libraries

Surface Manager, Media Framework, SQLite

OpenGL/ES, FreeType, LibWebCore

SGL, SSL, Libc

Android Runtime

Core Libraries

Dalvik Virtual Machine

Linux Kernel

Display Driver, Camera Driver, Bluetooth Driver, Flash Memory Driver, Binder (IPC) Driver

USB Driver, Keypad Driver, WiFi Driver, Audio Drivers, Power Management

Implementation:

🟩 **Applications, Application Framework: Java**

⬜ ▩ **System Libraries, Android Runtime: C and C++**

▢ **Linux Kernel: C**

› FOR SOCIETY

Random selection
&
Practical assignment explanation

Questions?