Module 1

Introduction to OS

Oswald c, SCOPE
VIT Chennai

Standard Text Books

- Silberschatz, Gagne, Galvin: Operating System Concepts, 6th Edition(most preferred)
- Operating Systems: Internals and Design Principles – 8th Edition or above by William Stallings
- Remzi H. Arpaci-Dusseau, Andrea C. Arpaci-Dusseau - Operating Systems_ Three Easy Pieces
- Ramez Elmasri, A Carrick, David Levine - Operating Systems_ A Spiral Approach (2009, McGraw-Hill Science_Engineering_Math)
- Orielly Series related to Pthreads, Linux (Rober Love), etc.
- Other online resources will be shared through the course

Operating Systems - Introduction

What is the need for an Operating System ?

Need ; Functionalities ; Types

✓Interface or Intermediary between User and Computer H/W

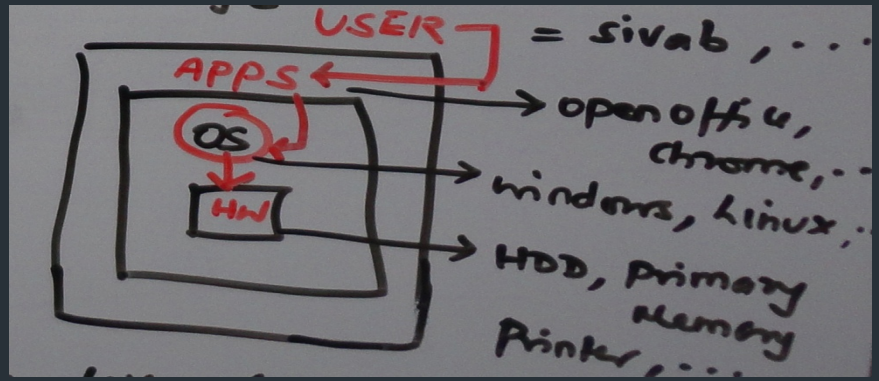
✓Software or Collection of Programs to manage the User and HW

✓Another view – as a Resource Manager to manage the various HW and Software Resources

Primary Responsibility

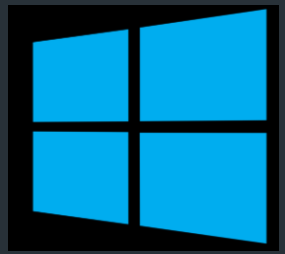
Ease of Use / User Convenience ; Efficient Management of Resources

OS is like a Government!! – No useful function by itself but sets up environment for other applications / end users to achieve their task!



Introduction to OS

- Operating System(OS) manages computer hardware.
- Application Program runs on OS
- Computer User interacts with the OS which in turn interacts with the hardware.
- Variety of OS depending on the tasks. Example:-
  - Mainframe OS
    - Optimize hardware utilization
  - Computer standard OS
    - Standard Application, Games, etc.
  - Handheld OS -> Apps, etc.

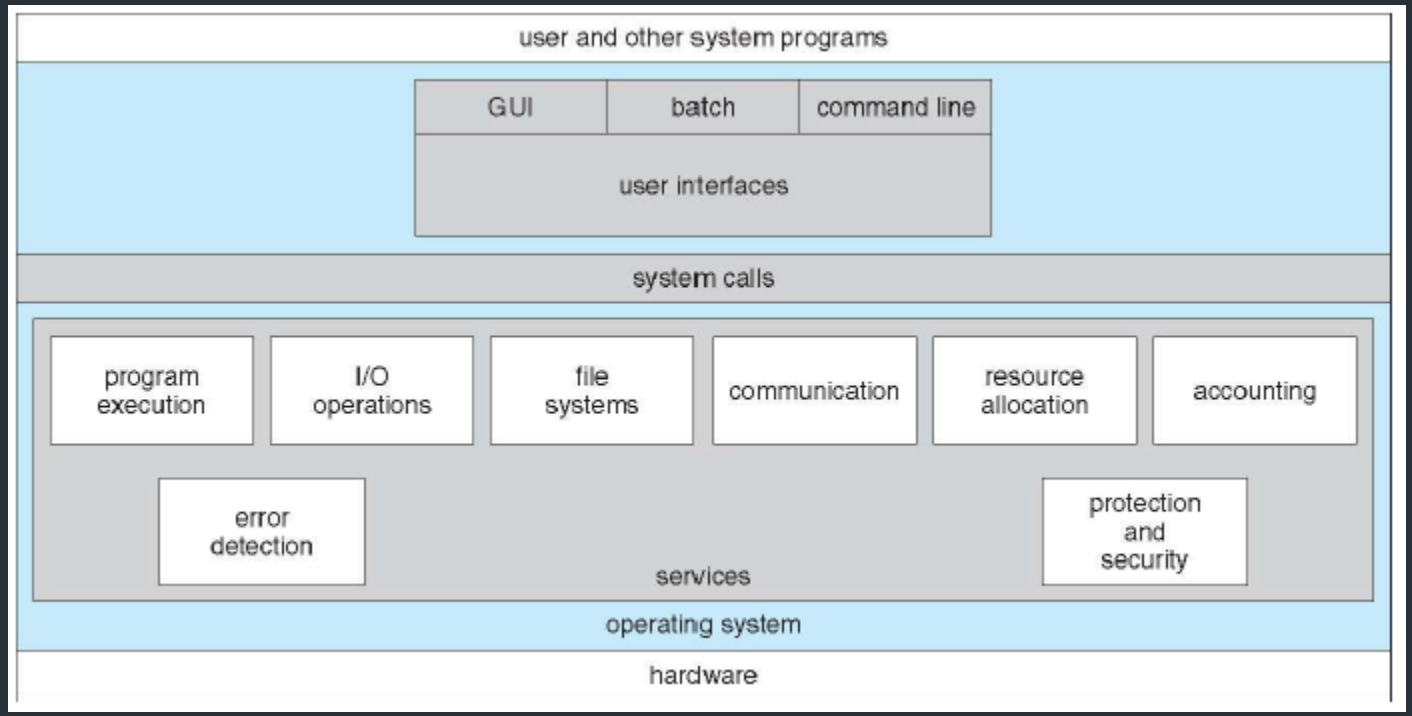



iOS

OS as a base for Application Programs

- Application program runs on a platform and that platform is an Operating systems.
- OS plays an important role to determine which application you need, because some applications may exists only in some OS.
- Example:
  - Words in windows
  - Libre office in linux

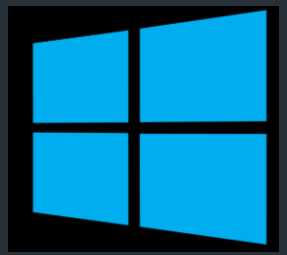
Schematic of Operating System Services



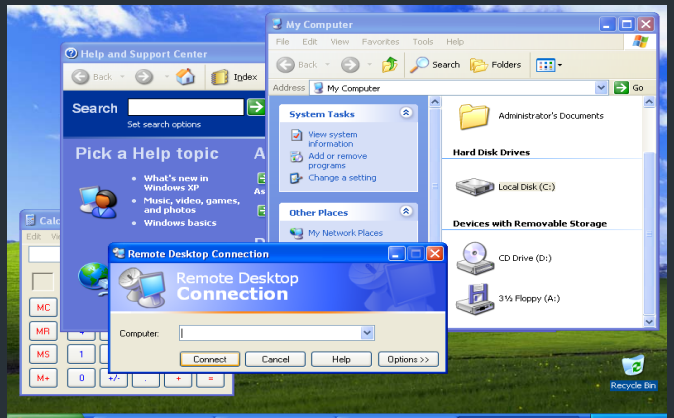
Silberschatz, Gagne, Galvin: Operating System Concepts, 6th Edition

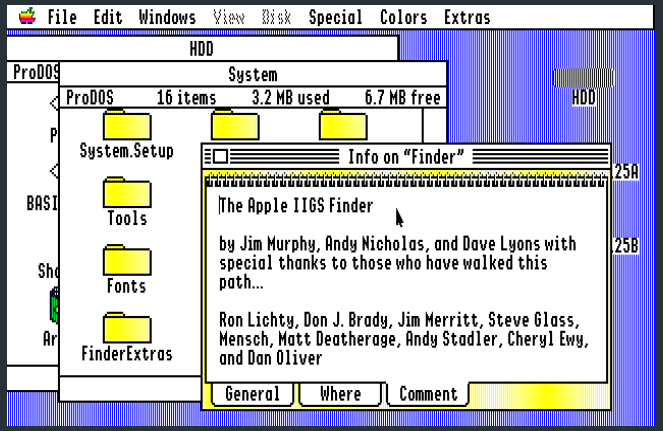
Different types of OS

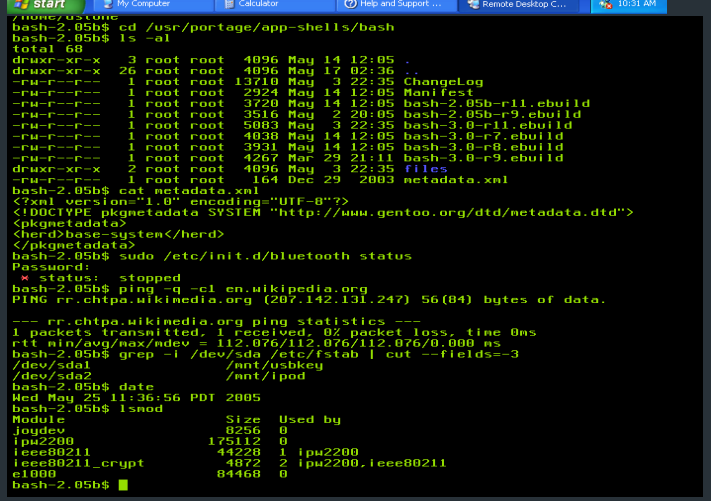
- Microsoft Windows
- Mainframe
- DOS
- OS/2
- Linux - Example Ubuntu
- Mac OS
- AmigaOS

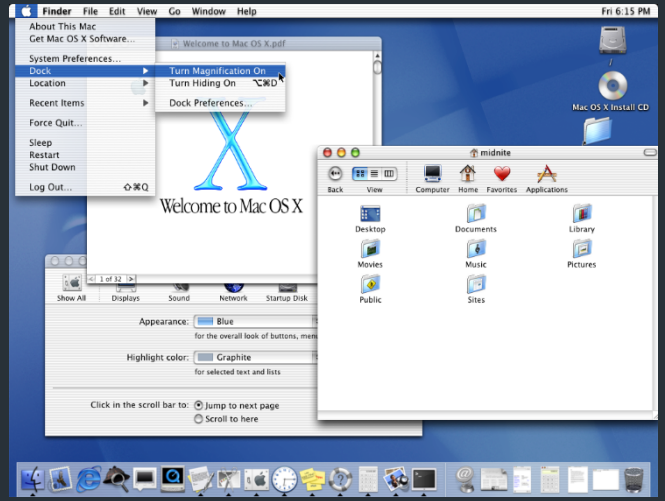


Types Of OS


My Computer
Help and Support Center
Search
Pick a Help topic
Remote Desktop Connection
Remote Desktop Connection
Computer:
Connect
Cancel
Help
Options >>
start
My Computer
Calculator
Help and Support ...
Remote Desktop C...
10:31 AM


File Edit Windows View Disk Special Colors Extras
HDD
System
ProDOS 16 items 3.2 MB used 6.7 MB free
System.Setup
Tools
Fonts
FinderExtras
Info on "Finder"
The Apple IIGS Finder
by Jim Murphy, Andy Nicholas, and Dave Lyons with special thanks to those who have walked this path...
Ron Lichty, Don J. Brady, Jim Merritt, Steve Glass, Mensch, Matt Deatherage, Andy Stadler, Cheryl Ewy, and Dan Oliver
General
Where
Comment


bash-2.05b$ cd /usr/portage/app-shells/bash
bash-2.05b$ ls -al
total 68
drwxr-xr-x 3 root root 4096 May 14 12:05 .
drwxr-xr-x 26 root root 4096 May 17 02:36 ..
-rw-r--r-- 1 root root 13710 May 3 22:35 ChangeLog
-rw-r--r-- 1 root root 2924 May 14 12:05 Manifest
-rw-r--r-- 1 root root 3720 May 14 12:05 bash-2.05b-r11.ebuild
-rw-r--r-- 1 root root 3516 May 2 20:05 bash-2.05b-r9.ebuild
-rw-r--r-- 1 root root 5083 May 3 22:35 bash-3.0-r11.ebuild
-rw-r--r-- 1 root root 4038 May 14 12:05 bash-3.0-r7.ebuild
-rw-r--r-- 1 root root 3931 May 14 12:05 bash-3.0-r8.ebuild
-rw-r--r-- 1 root root 4267 Mar 29 21:11 bash-3.0-r9.ebuild
drwxr-xr-x 2 root root 4096 May 3 22:35 files
-rw-r--r-- 1 root root 164 Dec 29 2003 metadata.xml
bash-2.05b$ cat metadata.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pkgmetadata SYSTEM "http://www.gentoo.org/dtd/metadata.dtd">
<pkgmetadata>
<herd>base-system</herd>
</pkgmetadata>
bash-2.05b$ sudo /etc/init.d/bluetooth status
Password:
* status: stopped
bash-2.05b$ ping -q -c1 en.wikipedia.org
PING rr.chtpa.wikimedia.org (207.142.131.247) 56(84) bytes of data.
--- rr.chtpa.wikimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 112.076/112.076/112.076/0.000 ms
bash-2.05b$ grep -i /dev/sda /etc/fstab | cut --fields=-3
/dev/sda1 /mnt/usbkey
/dev/sda2 /mnt/ipod
bash-2.05b$ date
Wed May 25 11:36:56 PDT 2005
bash-2.05b$ lsmod
Module Size Used by
joydev 8256 0
ipw2200 175112 0
ieee80211 44228 1 ipw2200
ieee80211_crypt 4872 2 ipw2200,ieee80211
e1000 84468 0
bash-2.05b$


Finder File Edit View Go Window Help
About This Mac
Get Mac OS X Software...
System Preferences...
Dock
Location
Recent Items
Force Quit...
Sleep
Restart
Shut Down
Log Out...
Turn Magnification On
Turn Hiding On ⌥⌘D
Dock Preferences...
Welcome to Mac OS X
midnite
Desktop
Documents
Library
Movies
Music
Pictures
Public
Sites
Appearance:
Highlight color:
Graphite
Click in the scroll bar to:
Jump to next page
Scroll to here

OS Services

- **User Interface:** There are different kinds, like touchscreen, GUI, and command-line.
- **Program Execution:** (Execute programs for users)
- **I/O operations:** It is much too difficult for users to operate the I/O hardware correctly without help.
- **File System Manipulation:** The OS helps us store, organize, manage, and protect our information.
- **Communications:** Users need their processes to exchange information. OSs help. The two main ways to do it are *with shared memory* and *by message passing*.
- **Error Detection:** An OS continually checks to see if something is going wrong. The OS is programmed to take appropriate action.

Program execution

- OS handles many activities, that are encapsulated as a process.
- Process refer to a full execution that includes:-
  - code to execute,
  - data to manipulate,
  - registers,
  - OS resources in use.
- When Program is executing the OS manages the following:
  - Loads a program into memory.
  - Executes the program.
  - Handles program's execution.
  - Provides a mechanism for process synchronization.
  - Provides a mechanism for process communication.
  - Provides a mechanism for deadlock handling.

Other Important OS Services

- Other tasks include:
  - **Resource Allocation**
  - **Logging:**
    - Records for accounting, fault detection, failure, protection, maintenance, update, security, etc.
  - **Protection and Security**

OS Design Issues

- Depends on the type of hardware and system.
- Perspective:
  - User–
    - convenient to use,
    - easy to learn,
    - reliable,
    - safe,
    - secure, and
    - fast
  - System goals–
    - easy to design,
    - implement,
    - maintain,
    - flexible,
    - reliable,
    - error-free,
    - secure, and
    - efficient

Design Goals

- **Design Goals:**
  - system that is convenient,
  - reliable,
  - safe, and
  - fast.
- **Mechanisms and Policies:**
  - *Policy* refers to *what* we decide to do. *Mechanism* refers to *how* we do what we do.
  - Policies change over time, so the computing system tends to be more flexible if we build in mechanisms that can support a range of policies.

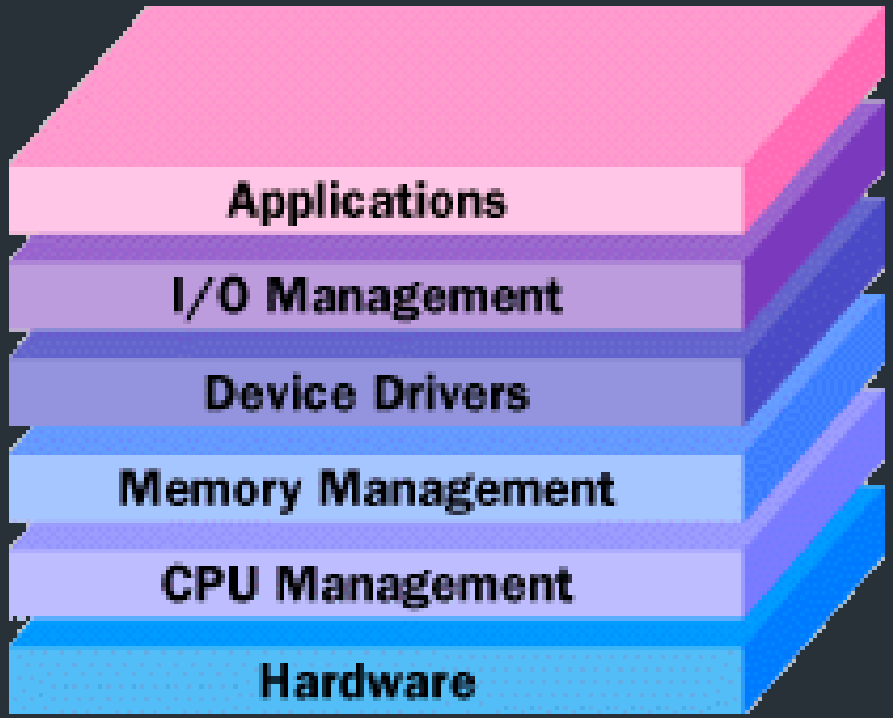
Implementation

- **Implementation: The** *implementation* of the operating system, that is the manner in which the ideas of the design are written in programming language(s).
  - Assembly
  - High Level Language
- Earlier assembly could make the code run faster but nowadays high-level are translated to equivalently good assembly code.
- Instead performance of OS will increase if selection data structure and algorithms are done rather than proper assembly code.

Operating System Structure

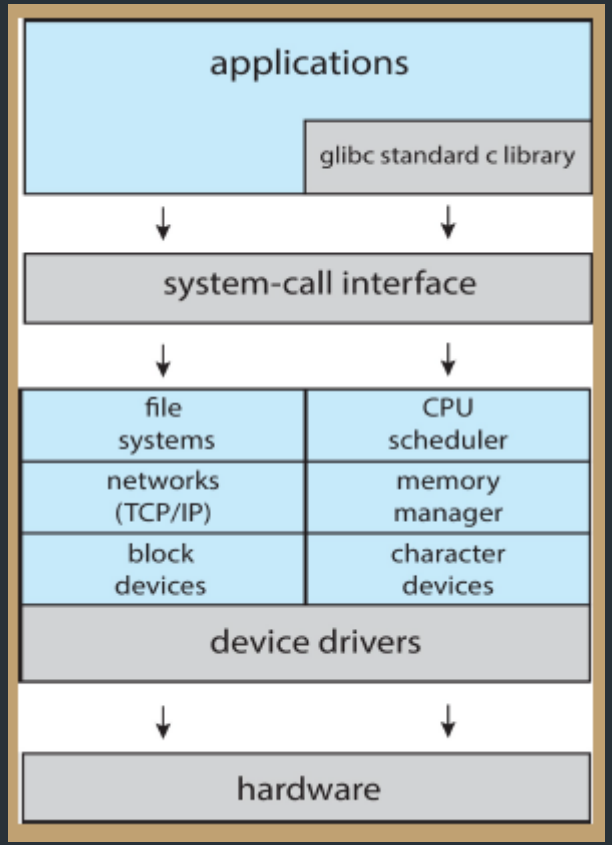
- Monolithic,
- Layered,
- Modular,
- Micro-kernel models

Operating-System Structure



Monolithic

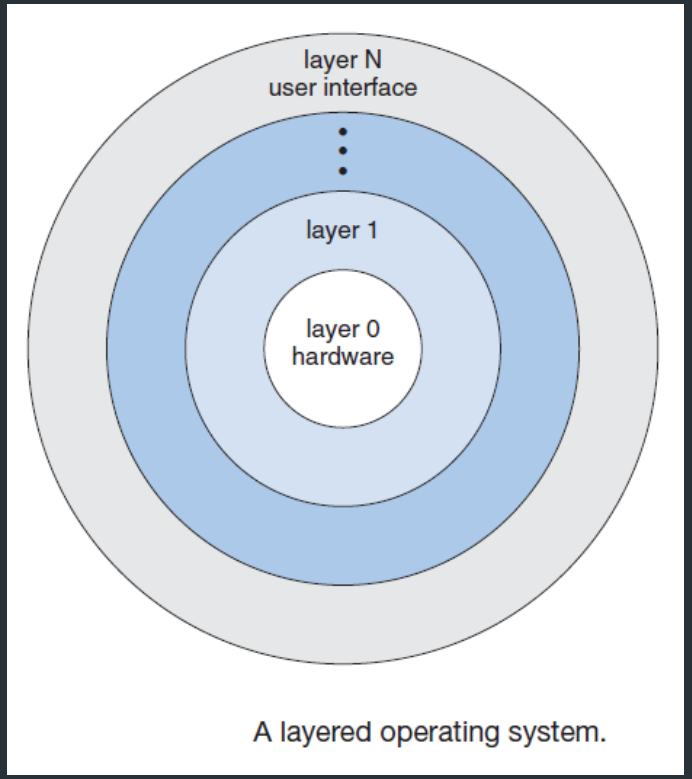
- Monolithic Kernel
  - value on speed and efficiency.
  - Monolithic is a single static binary file.
  - It executes in a single address space.



Silberschatz, Gagne, Galvin: Operating System Concepts, 6th Edition

Layered

- Division into number of layers as shown in figure.
- Innermost layer is hardware
- Outermost layer is interface
- Accordingly as we move from innermost to outermost layer we observe that we are moving from the perspective of hardware to software with each interlinkage layer.
- Simple
- Easy to Debug
- Easy to verify

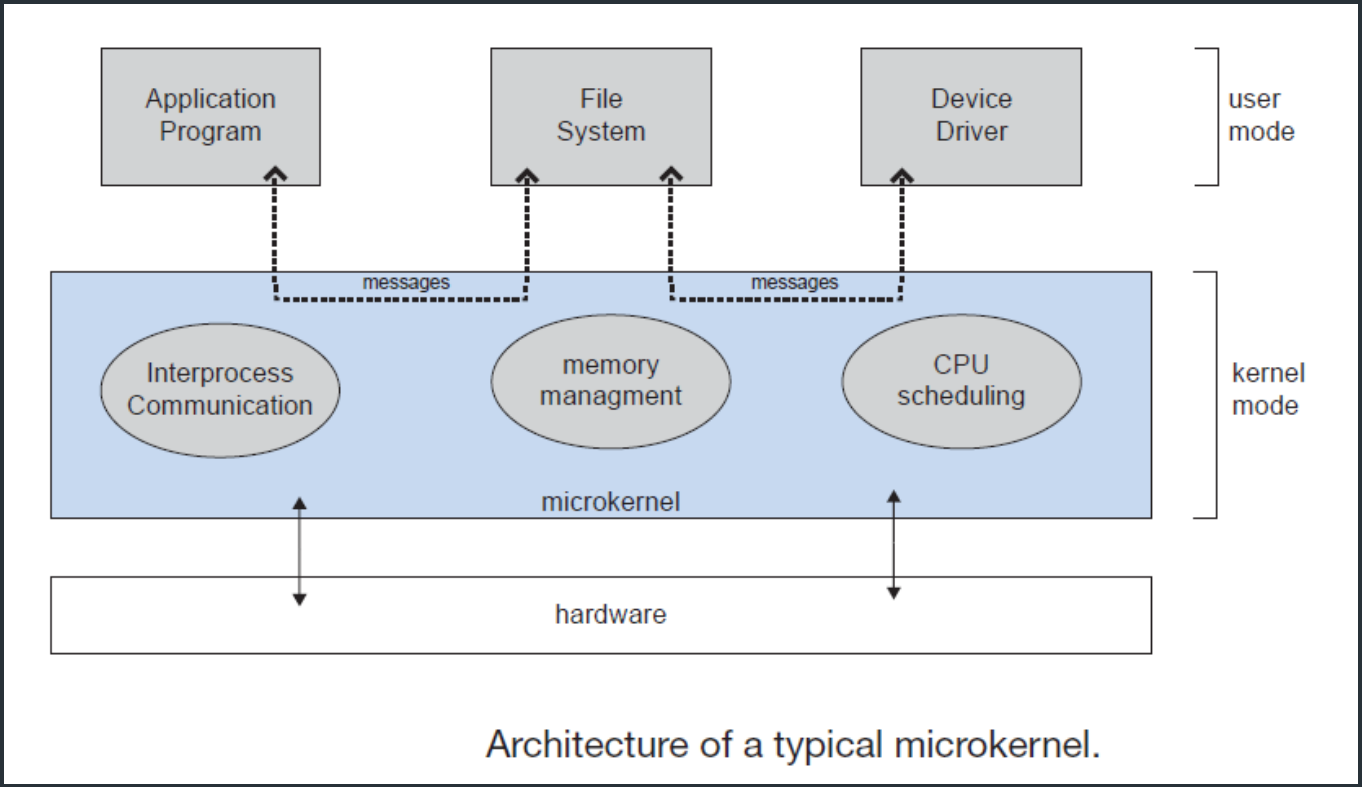


A layered operating system.

Microkernels:

- Keep only necessary component in kernel. Others are implemented as programs (system or user level).
- Resulting in a kernel smaller in size.
- Minimal process management
- Minimal memory management
- Main role is that it facilitates communication between the client program and the various services that are running in user space.

Microkernel

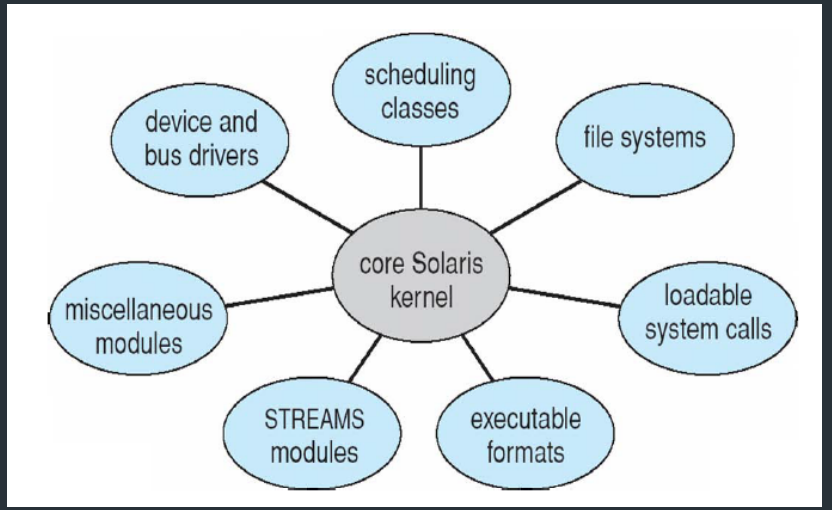


Architecture of a typical microkernel.

Silberschatz, Gagne, Galvin: Operating System Concepts, 6th Edition

Modular

- Divided into different module.
- Typically employs:
  - dynamic loadable kernel module (LKM). i.e. Different modules communicate through kernel (core part).
- LKM may be loaded during boot or when required, and can be deleted also.
- An example would be a device driver support module loaded when a new device is plugged into the computer, and when the device is unplugged, the module is deleted because it is not needed any more.



Silberschatz, Gagne, Galvin: Operating System Concepts, 6th Edition

Abstraction

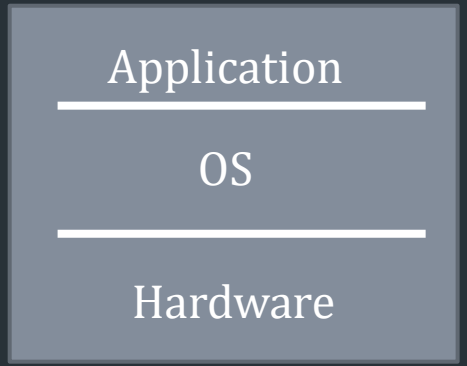
- OS acts as an intermediary between a user and the hardware
- Interface for the user is provided by the OS. This interface is how a user use the service.
- An abstraction is a software that hides lower level details and provides a set of higher-level functions. An operating system transforms the physical world of devices, instructions, memory, and time into virtual world that is the result of abstractions built by the operating system.
- Creates an environment for the user

Abstraction

- Processor → Thread
- Memory → Address Space
- Disks, SSDs, … → Files
- Networks → Sockets
- Machines → Processes
- OS as an Illusionist:
  - Remove software/hardware quirks (fight complexity)
  - Optimize for convenience, utilization, reliability, … (help the programmer)
- For any OS area (e.g. file systems, virtual memory, networking, scheduling):
  - What hardware interface to handle? (physical reality)
  - What's software interface to provide? (nicer abstraction)

Abstract Machine Interface

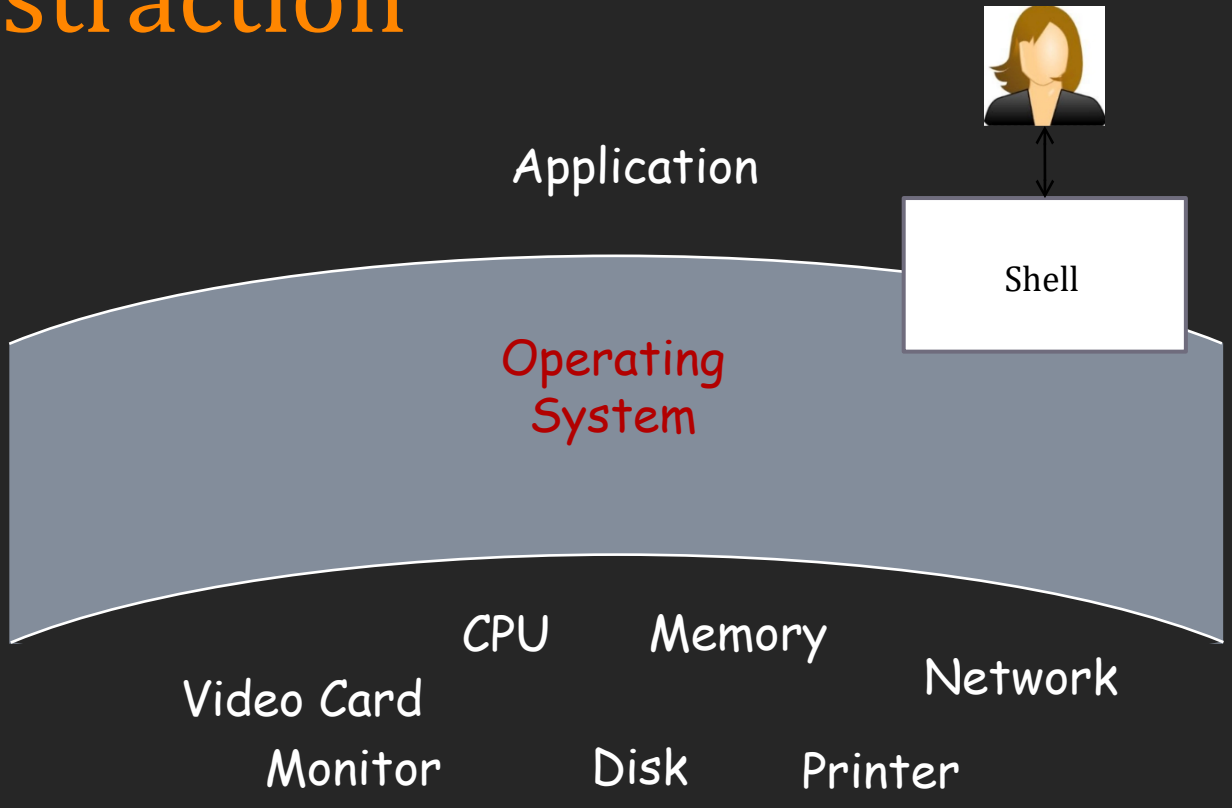
Physical Machine Interface



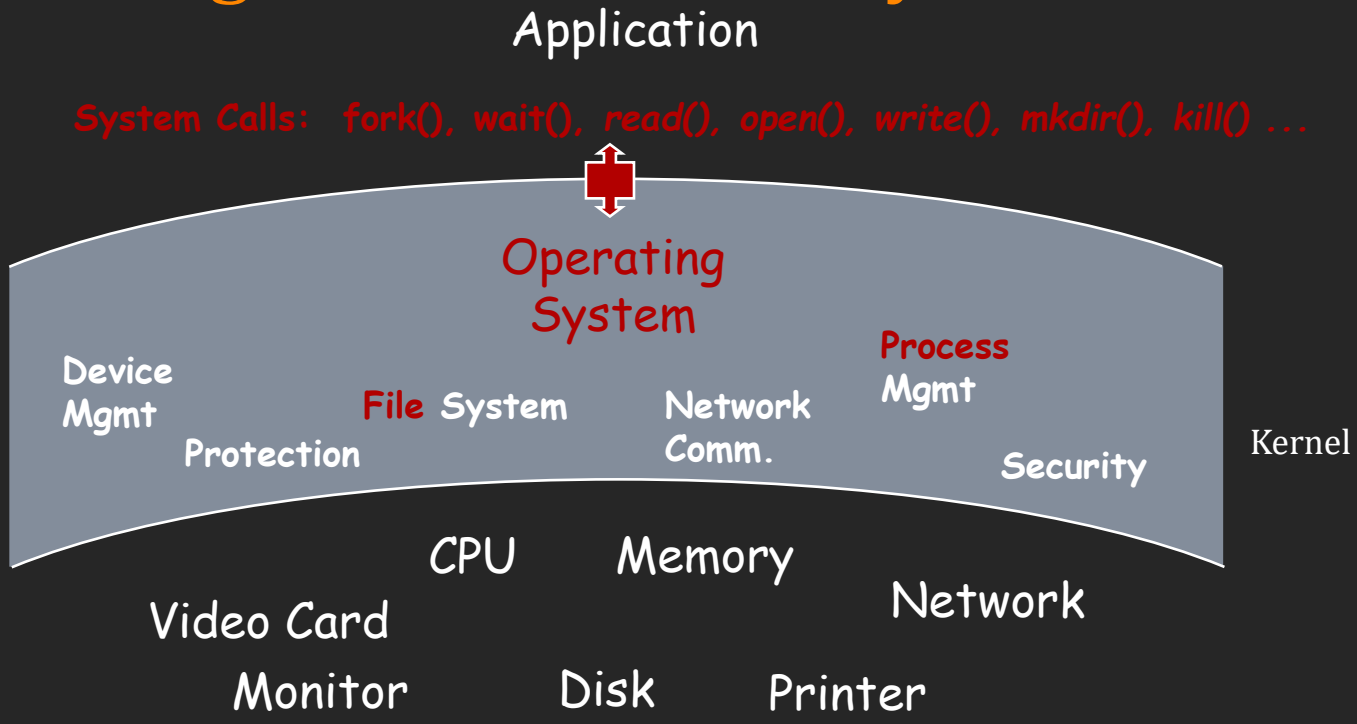
Abstraction

- Abstract Machine
  - Complex details of the hardware are hidden
  - APIs
  - Application development becomes simple
- Command Interpreter
  - Part of a OS that understands and executes commands that are entered interactively by a human being or from a program
  - Shell

Abstraction



Providing abstraction via system calls



Why is abstraction important?

- Without OSs and abstract interfaces, application writers must program all device access directly
  - load device command codes into device registers
  - understand physical characteristics of the devices

- Applications suffer!
  - Very complicated maintenance and upgrading
  - No portability

Concept of Process

- Process
  - Program loaded in memory and in execution.
- Program is a passive whereas process is an active entity

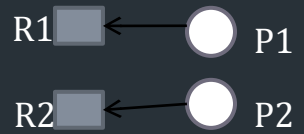
Process Management

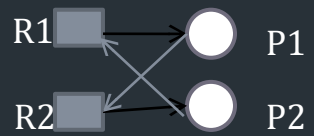
- Process requires resources for completion.
  - CPU time
- Representation of process
  - One **program counter** specifying location of next instruction to execute
  - Data structure (stores information of a process)
- Many processes may be associated with the same program
  - user processes,
  - operating system processes
- Life cycle of a process
  - States
  - Arrival, Computation, I/O, I/O completion, termination

Process Management Activities

The operating system is responsible for the following activities in connection with process management:

- Initiating and Terminating Processes
- To pause and resume processes
- Process scheduling
- Mechanism for:
  - Process synchronization, communication and deadlock handling.





Resource

- OS acts an interface between hardware and software.
- Resources are objects that can be allocated in a computer . Examples:
  - Processors,
  - Devices: Both input and output devices,
  - Memory,
  - Files

Purpose of OS in terms of resource

- Thus, we can restate the purpose of the operating system in terms of resources.
  - The operating system manages resources (resource allocation) and
  - To provides an interface to resources for application programs (resource abstraction).

Influence of Security

- Security must consider external environment of the system, and protect it from:
- unauthorized access.
- malicious modification or destruction
- accidental introduction of inconsistency.
- Easier to protect against accidental than malicious misuse.

Authentication

- User identity most often established through *passwords*,
- can be considered a special case of either keys or
- capabilities.
- Passwords must be kept secret.
  - Frequent change of passwords.
  - Use of “non-guessable” passwords.
  - Log all invalid access attempts.
  - Passwords may also either be encrypted or allowed to be used only once.

Other Security Issues

- Program Threats
  - Trojans
  - Trap Door
- System Threats
  - Worms
  - Viruses
  - Denial of Services

Networking and OS

- A modern OS contains built-in software designed to simplify networking.
- Typical OS software includes an implementation of TCP/IP and related utility programs such as ping and traceroute, along with device drivers and other software to automatically enable the Ethernet or wireless interface for a device.
- The operating systems of mobile devices normally provide programs to enable Wi-Fi, Bluetooth, and other wireless connectivity.

Multimedia OS

- The operating system provides a comfortable environment for the execution of programs, and it ensures effective utilization of the computer hardware.
- The OS offers various services related to the essential resources of a computer: CPU, main memory, storage and all input and output devices.
- In multimedia applications, a lot of data manipulation (e.g. A/D, D/A and format conversion) is required and this involves a lot of data transfer, which consumes many resources.
- The integration of discrete and continuous multimedia data demands additional services from many operating system components.
- The major aspect in this context is real-time processing of continuous media data and synchronization

Demand on the applications

- Soft real-time applications: statistical guarantees
  - Examples: Streaming media, virtual games
- Interactive applications: no absolute performance guarantees, but low average response times
  - Examples: Editors, compilers
- Throughput-intensive Applications: no performance guarantees, but high throughput
  - Examples: http, ftp servers

Any Questions ?

References

- Silberschatz, Gagne, Galvin: Operating System Concepts, 6th Edition
- Remzi H. Arpaci-Dusseau, Andrea C. Arpaci-Dusseau - Operating Systems_ Three Easy Pieces
- Ramez Elmasri, A Carrick, David Levine - Operating Systems_ A Spiral Approach (2009, McGraw-Hill Science_Engineering_Math)
- https://www2.eecs.berkeley.edu/Courses/CS162/

Thank You!