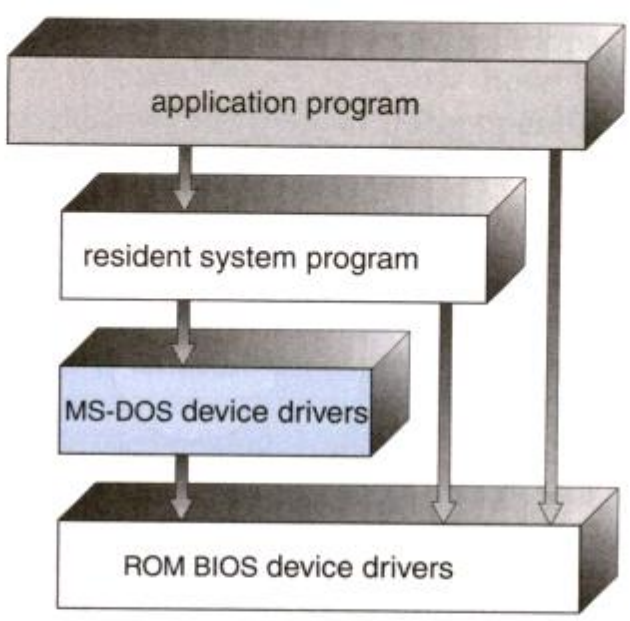
Govt Mahila Engineering College Ajmer

Department of Computer Engineering

Midterm I Exam Session 2017-18 Subject:Advanced Operating System Sem VI(IT) MM.20

Q1) Explain operating system structure,policies and mechansisms. 5

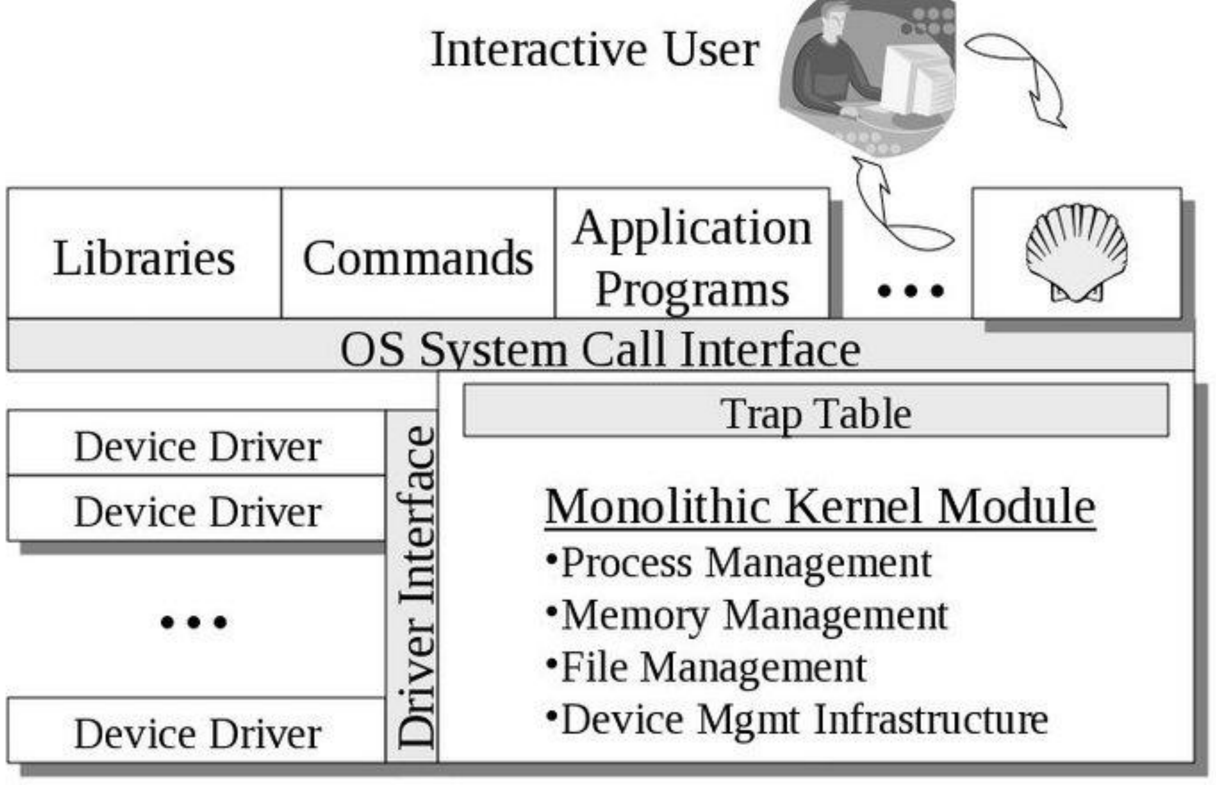
Solution: Simple Structure



- Operating systems such as MS-DOS and the original UNIX did not have well-defined structures.
- There was no CPU Execution Mode (user and kernel), and so errors in applications could cause the whole system to crash.

Monolithic Approach

- Functionality of the OS is invoked with simple function calls within the kernel, which is one large program.
- Device drivers are loaded into the running kernel and become part of the kernel



Policies are ways to choose which activities to perform. Mechanisms are the implementations that enforce policies, and often depend to some extent on the hardware on which the operating system runs. For instance, a processes may be granted resources using the first come, first serve policy. This policy may be implemented using a queue of requests. Often the kernel provides mechanisms that are used to implement policies in servers.

Policy vs mechanism OS examples

- Granting a resource to a process using first come first serve algorithm (policy). This policy can be implemented using a queue (mechanism).
- Thread scheduling or answering the question "which thread should be given the chance to run next?" is a policy. For example, is it priority based ? or just round robin ?. Implementing context switching is the corresponding mechanism.
- In virtual memory, keeping track of free and occupied pages in memory is a mechanism. Deciding what to do when a page fault occurs is a policy

Separation of mechanism and policy

Separation of policy and mechanism is a design principe to achieve flexibility. In other words, adopting a certain mechanism should not restrict existing policies. The idea behind this concept is to have the least amount of implementation changes if we decide to change the way a particular feature is used. We can also look at it from the other side. For example, if a certain implementation needs to be changed (ex. improve efficiency). This must not greatly influence the way it is used. In the login example mentioned earlier (logging to a website) switching from a user name password pair to Facebook account should not prevent a user from logging in to the website.

Q2) Briefly outline the functionality of Monolithic Structure. 5

A **monolithic kernel** is an operating system architecture where the entire operating system is working in kernel space and is alone in supervisor mode. The monolithic model differs from other operating system architectures (such as the microkernel architecture) in that it alone defines a high-level virtual interface over computer hardware. A set of primitives or system calls implement all operating system services such as process management, concurrency, and memory management. Device drivers can be added to the kernel as modules

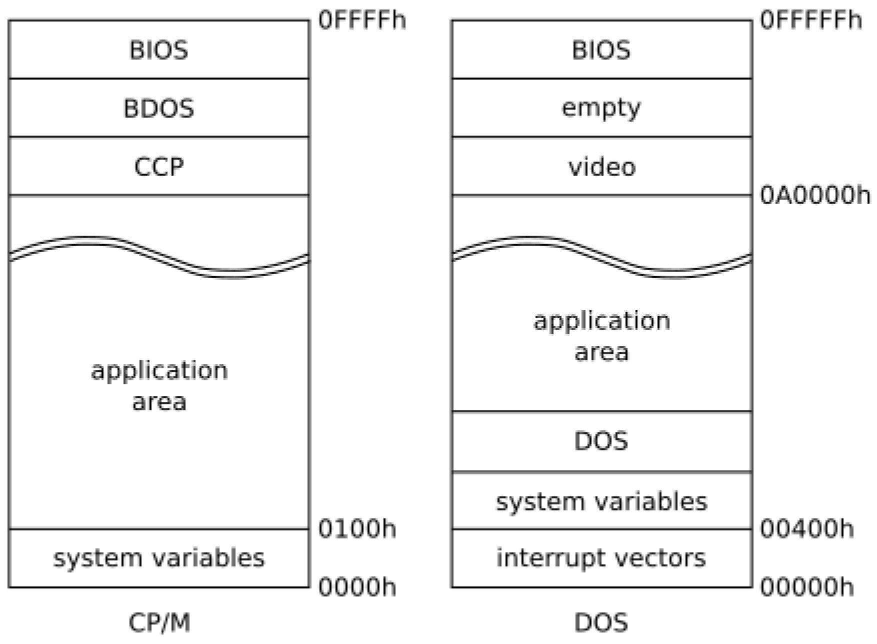


Figure Simple Monolithic Operating Systems Example

Q3) Comment on the advantages and disadvantages of Layered System. 5

Advantages of layered operating systems are:

1. It is decomposable and therefore effects separation of concerns and different abstraction levels
2. It allows good maintenance, where you can make changes without affecting layer interfaces

Disadvantages of layered operating systems are:

1. It is difficult to exactly assign of functionalities to the correct and appropriate layer
2. Because of having too many layers, performance of the system is degraded

Q4) What is paravirtualization. 5

In computing, **paravirtualization** is a virtualization technique that presents to virtual machines a software interface, which is similar yet not identical to the underlying hardware-software interface.

The intent of the modified interface is to reduce the portion of the guest's execution time spent performing operations which are substantially more difficult to run in a virtual environment compared to a non-virtualized environment. The paravirtualization provides specially defined 'hooks' to allow the guest(s) and host to request and acknowledge these tasks, which would otherwise be executed in the virtual domain (where execution performance is worse). A successful paravirtualized platform may allow the virtual machine monitor (VMM) to be simpler (by relocating execution of critical tasks from the virtual domain to the host domain), and/or reduce the overall performance degradation of machine-execution inside the virtual-guest.

Paravirtualization requires the guest operating system to be explicitly ported for the para-API — a conventional OS distribution that is not paravirtualization-aware cannot be run on top of a paravirtualizing VMM. However, even in cases where the operating system cannot be modified, components may be available that enable many of the significant performance advantages of paravirtualization. For example, the Xen Windows GPLPV project provides a kit of paravirtualization-aware device drivers, licensed under the terms of the GPL, that are intended to be installed into a Microsoft Windows virtual-guest running on the Xen hypervisor.