



OPERATING SYSTEMS

Module1_Part2



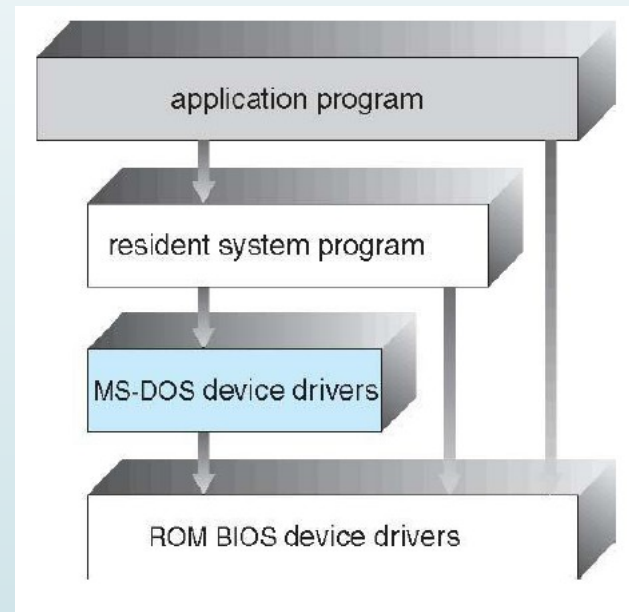
Operating System Structure

- ▮ Simple structure
- ▮ Layered
- ▮ Microkernel
- ▮ Modules

Simple structure

MS-DOS – written to provide the most functionality in the least space

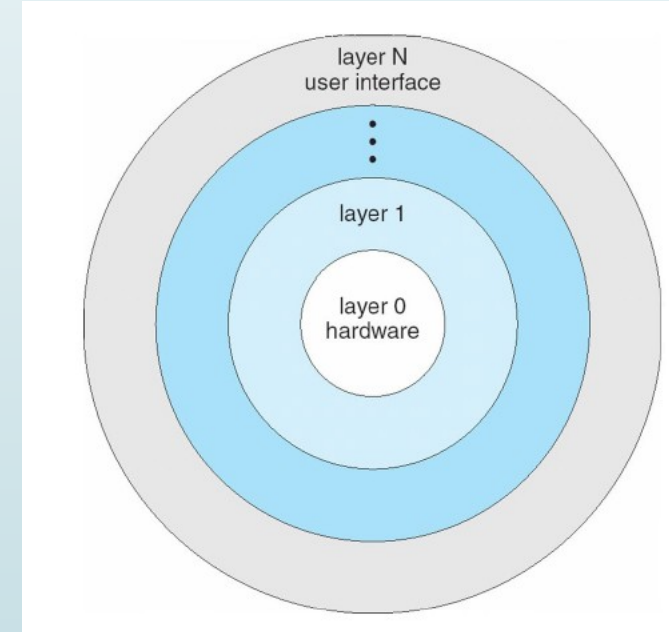
- Not divided into modules
 - Although MS-DOS has some structure, its interfaces and levels of functionality are not well separated
- It leaves base hardware access for application programs (no protection)
- Enormous amount of functionality to be combined into one level. This



Layered Approach

- ▮ **Layered approach:** In the layered approach, the OS is broken into a number of layers (levels) each built on top of lower layers. The bottom layer (layer 0) is the hardware & top most layer (layer N) is the user interface.

The layers are selected such that each uses functions (or operations) & services of only lower layer.





Layered Approach

- ▮ This approach simplifies debugging & system verification, i.e. the first layer can be debugged without concerning the rest of the system. Once the first layer is debugged, its correct functioning is assumed while the 2nd layer is debugged & so on.
- ▮ If an error is found during the debugging of a particular layer, the error must be on that layer because the layers below it are already debugged. Thus the design & implementation of the system are simplified when the system is broken down into layers.
- ▮ Each layer is implemented using only operations provided by lower layers. A layer doesn't need to know how these operations are implemented; it only needs to know what these operations do.
- ▮ The layer approach was first used in THE operating system. It was defined in six layers.



Layered Approach

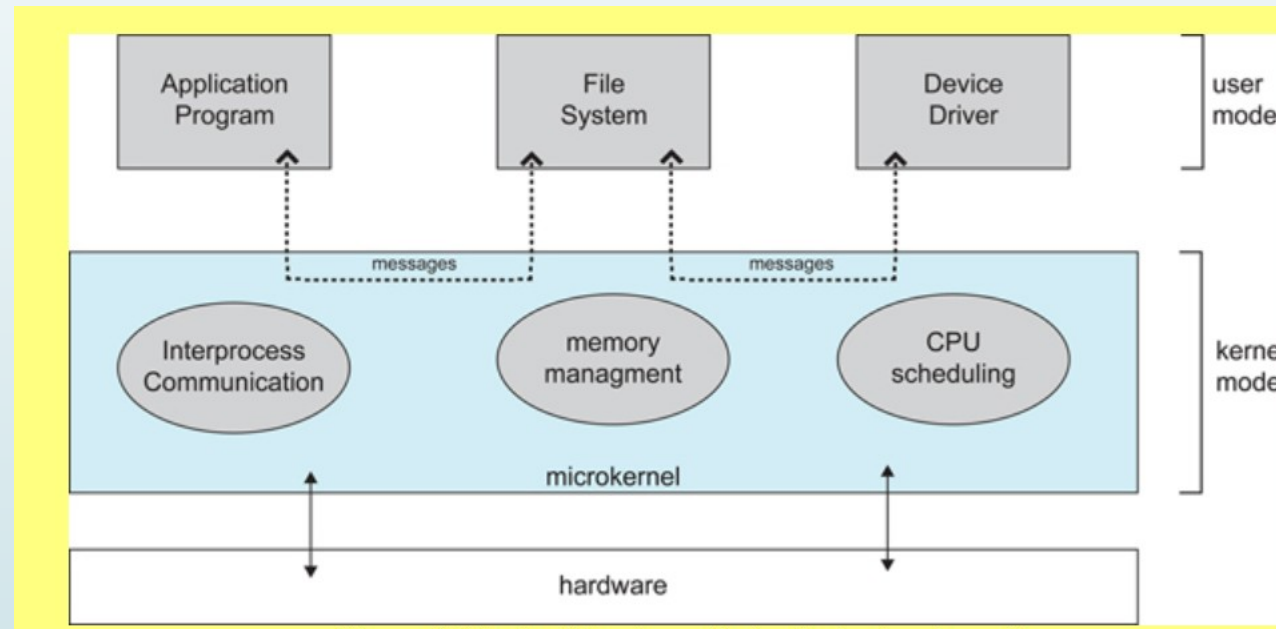
- The main disadvantage of the layered approach is:
- The main difficulty with this approach involves the careful definition of the layers, because a layer can use only those layers below it.
- It is less efficient than a non layered system (Each layer adds overhead to the system call & the net result is a system call that take longer time than on a non layered system).



Microkernels

- The basic idea behind micro kernels is to remove all non-essential services from the kernel, and implement them as system applications instead, thereby making the kernel as small and efficient as possible.
- Most microkernels provide basic process and memory management, and message passing between other services, and not much more.
- Security and protection can be enhanced, as most services are performed in user mode, not kernel mode.
- System expansion can also be easier, because it only involves adding more system applications, not rebuilding a new kernel.
- Mach was the first and most widely known microkernel, and now forms a major component of Mac OSX.
- Windows NT was originally microkernel, but suffered from performance problems relative to Windows 95. NT 4.0 improved performance by moving more services into the kernel, and now XP is back to being more monolithic.

Microkernels



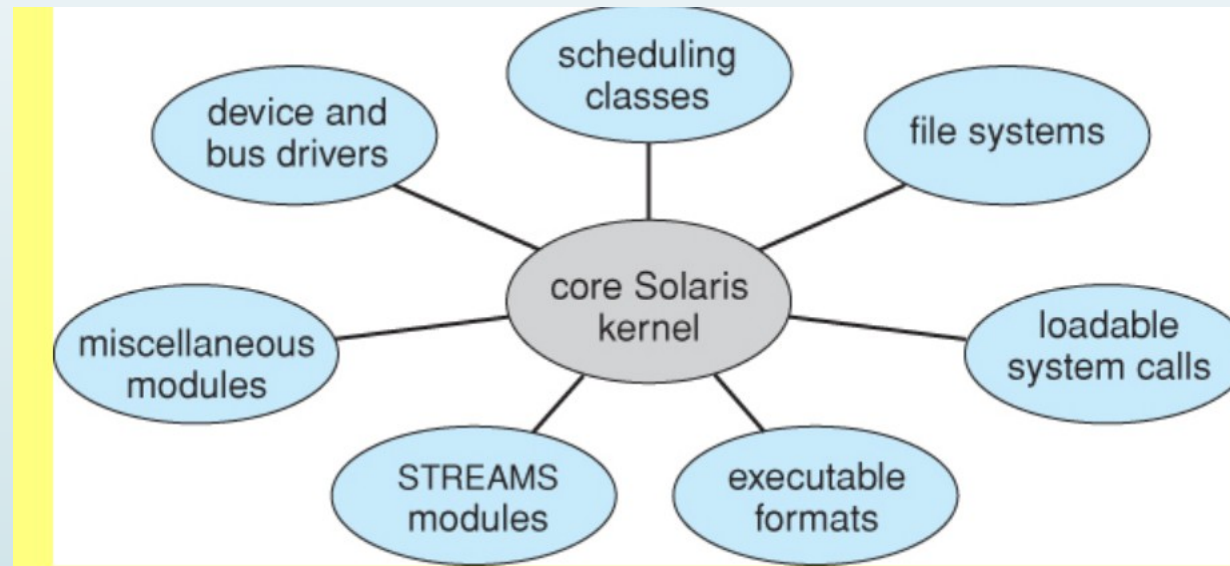
Architecture of a typical microkernel



MODULES

- Modern OS development is object-oriented, with a relatively small core kernel and a set of ***modules*** which can be linked in dynamically.
- Modules are similar to layers, in that each subsystem has clearly defined tasks and interfaces, but any module is free to contact any other module, eliminating the problems of going through multiple intermediary layers.
- The kernel is relatively small in this architecture, similar to microkernels, but the kernel does not have to implement message passing since modules are free to contact each other directly.

MODULES



Solaris loadable modules