Department of Computer Science and Engineering

FACULTY OF ENGINEERING AND TECHNOLOGY
UNIVERSITY OF LUCKNOW
LUCKNOW

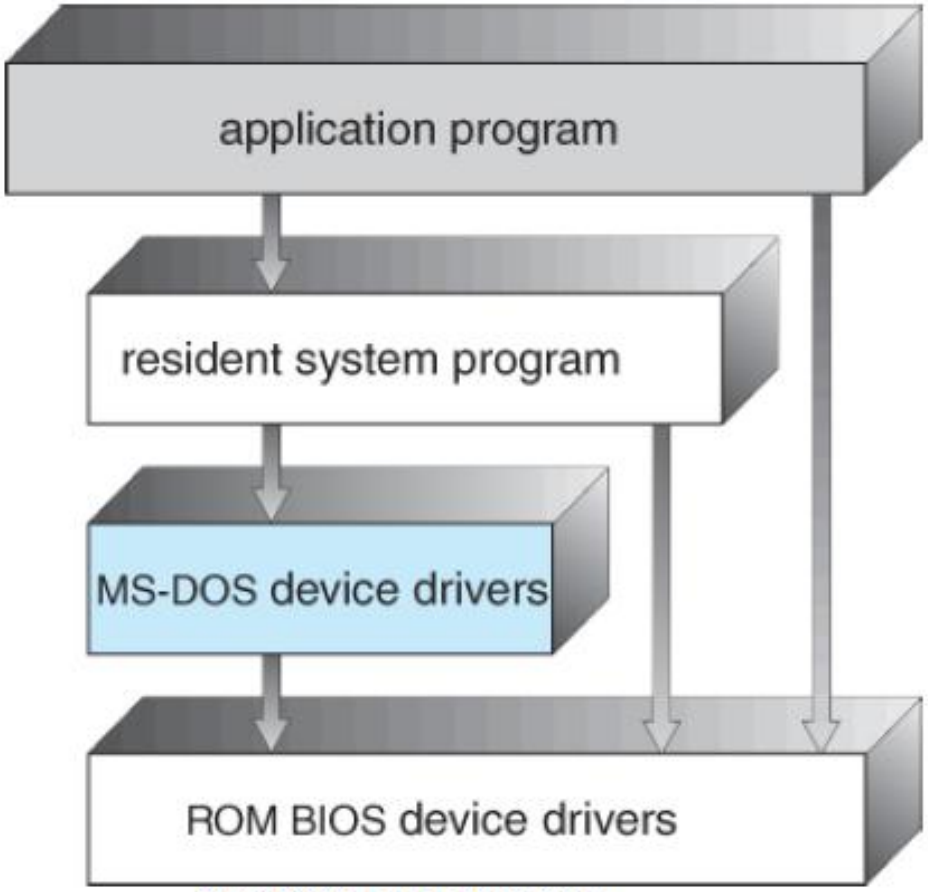
CS-501

Dr. Zeeshan Ali Siddiqui
Assistant Professor
Deptt. of C.S.E.

STRUCTURES

1. Simple Structure

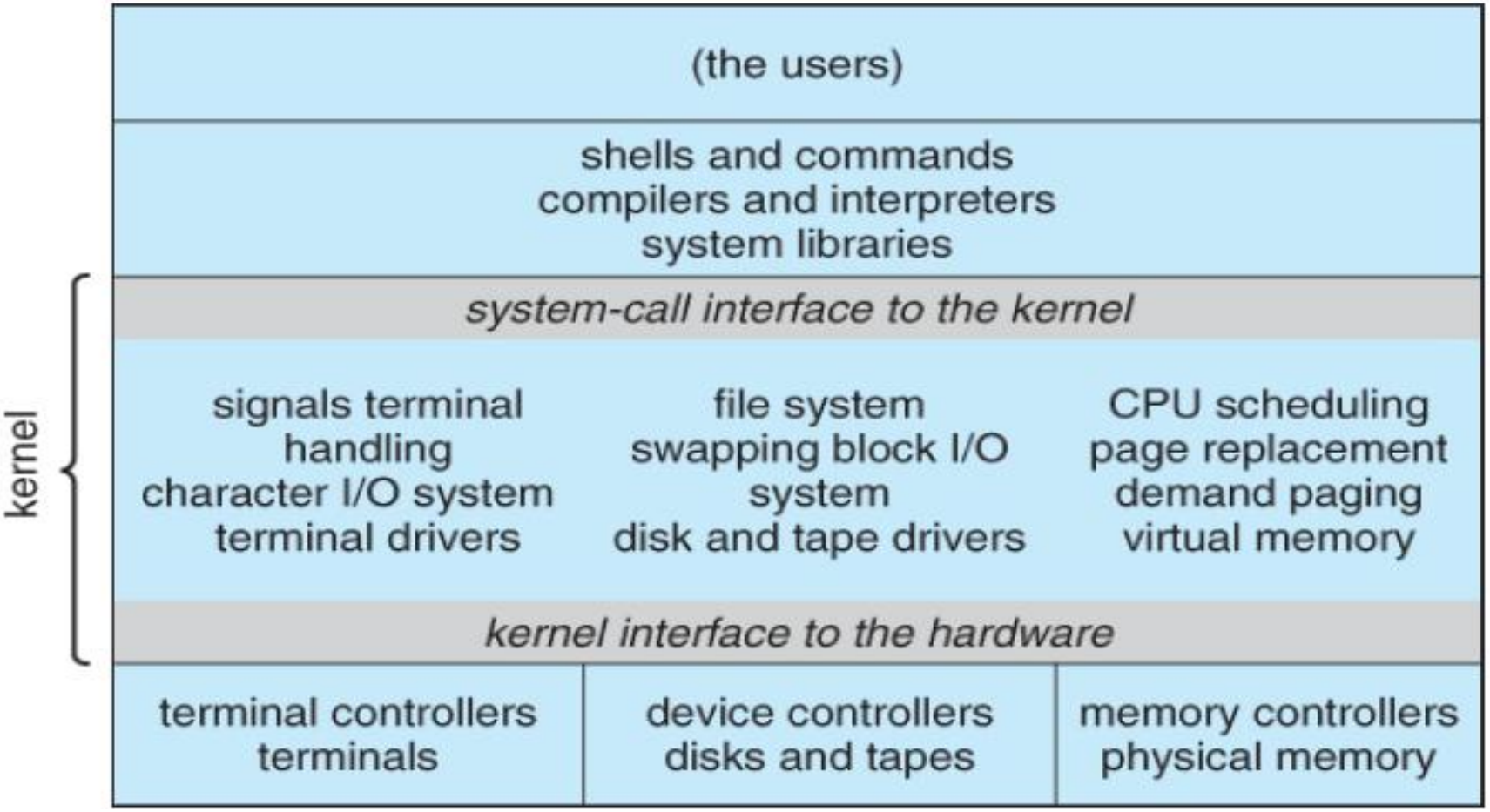
- It does not break the system into subsystems, and has no distinction between user and kernel modes, allowing all programs *direct access* to the underlying *hardware*.



MS-DOS layer structure

Eg: Traditional UNIX system structure

- The original UNIX OS used a *simple layered approach*, but almost all the OS was in one big layer, not really breaking the OS down into layered subsystems:

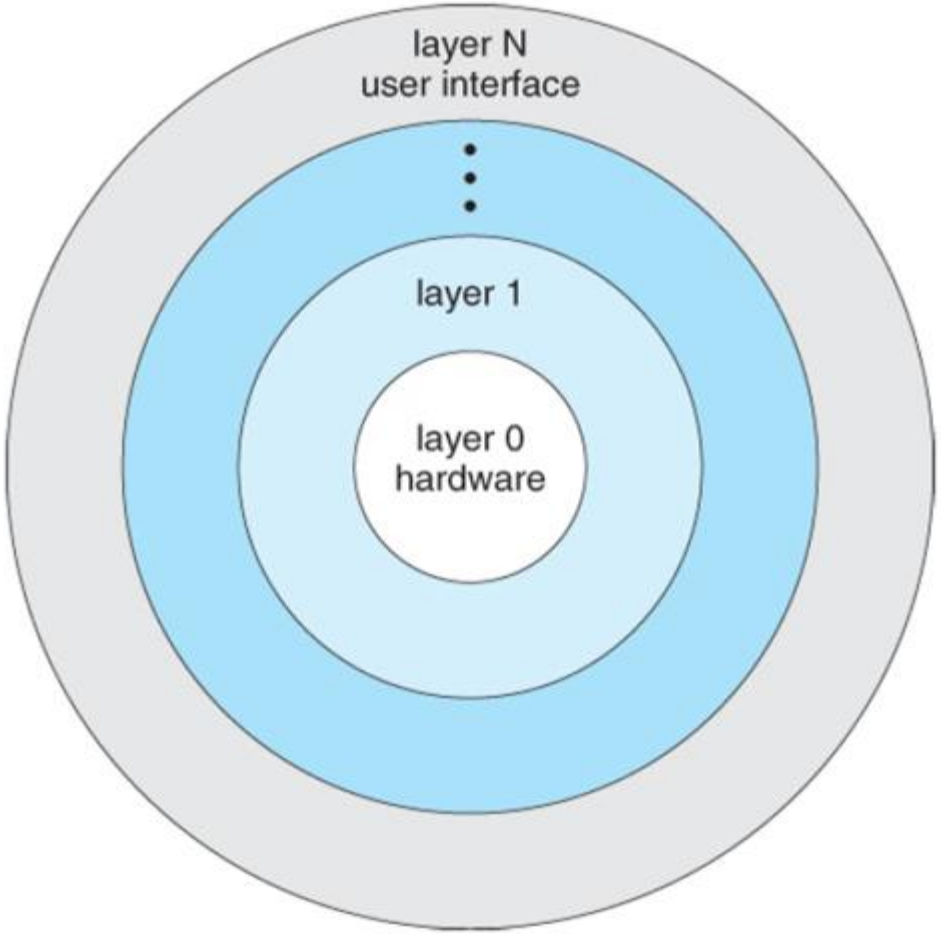


2. Layered Approach

- This approach allows each layer to be developed and debugged *independently*, with the assumption that all lower layers have already been debugged and are trusted to deliver proper services.

- Layered approaches can also be *less efficient*, as a request for service from a higher layer has to filter through all lower layers before it reaches the hardware, possibly with significant processing at each step.

A layered operating system



3. Microkernels 1/2

- 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, 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.

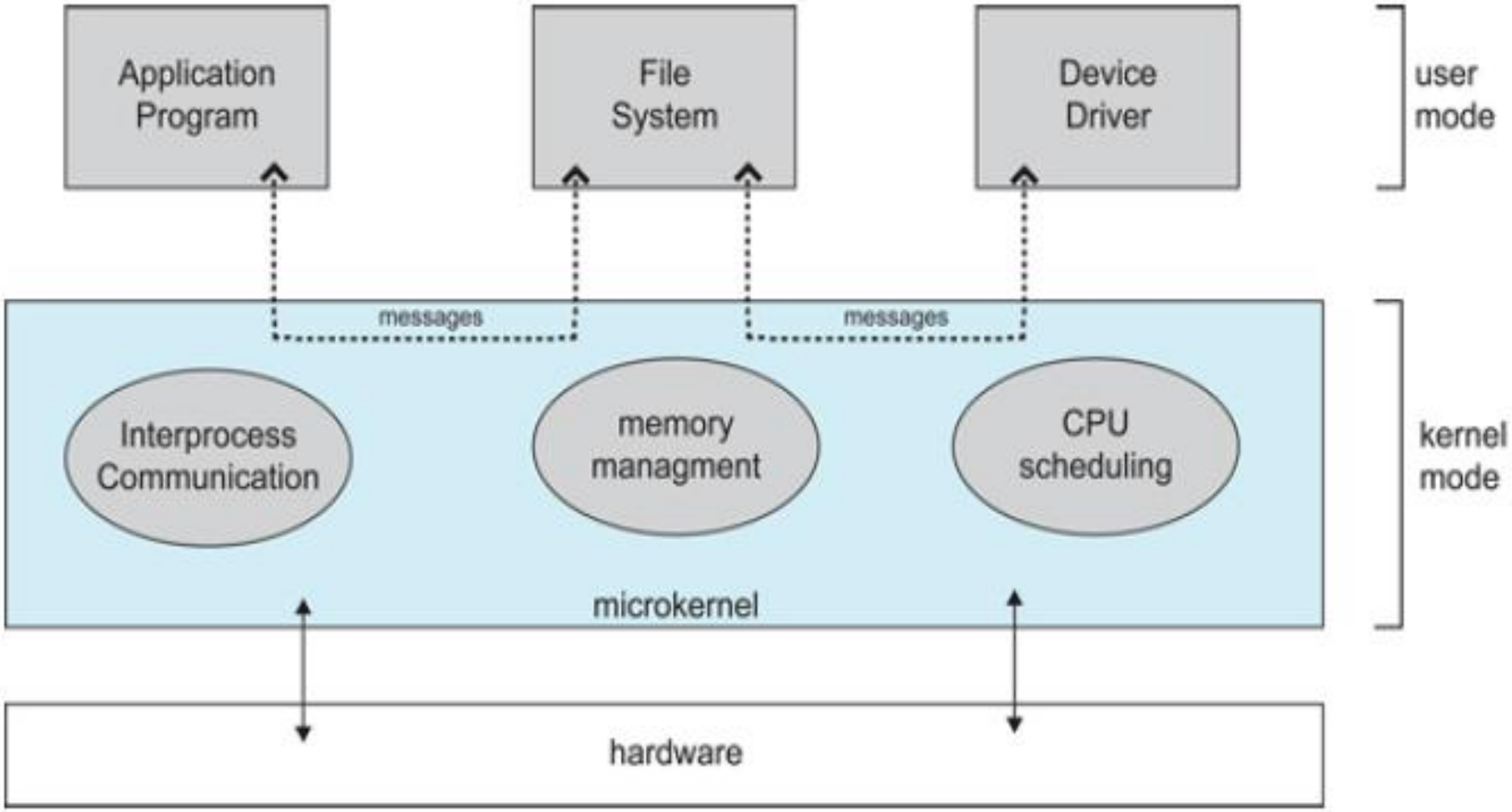
3. Microkernels 2/2

- 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.

- Another microkernel example is QNX, a real-time OS for embedded systems.

Architecture of a typical microkernel



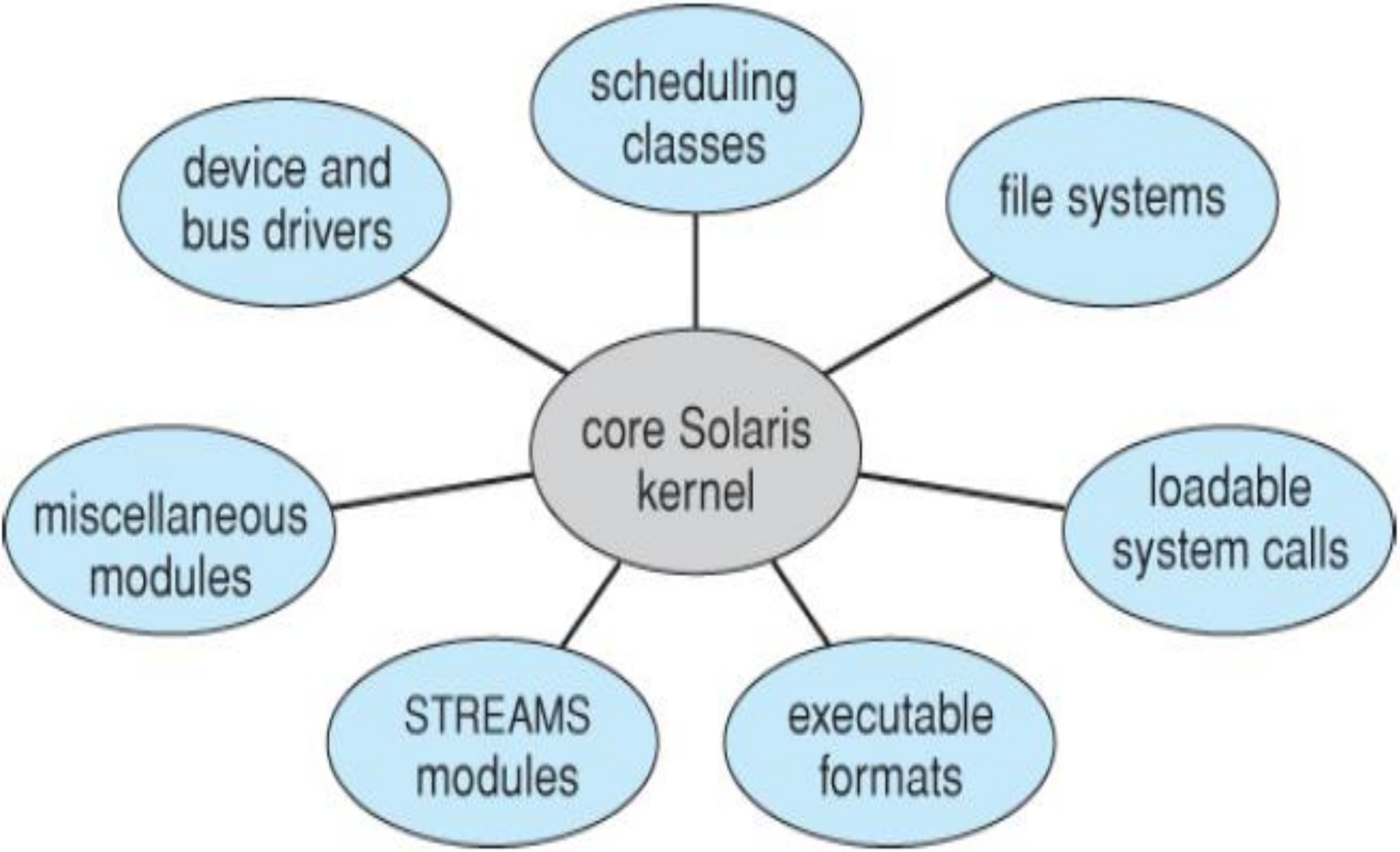
4. Modules

- 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.

Solaris loadable modules



5. Hybrid Systems

- Mac OS X
- iOS
- Android

Exercise

1. What do you mean by Kernel?

2. What is Kernel? Describe various operations performed by Kernel.

3. What is the main advantage of the layered approach to system design? What are the disadvantages of the layered approach?

4. Write about monolithic kernel, layered, and microkernel structures of operating systems.

5. Explain briefly Layered OS structure with neat sketch.

References

1. Silberschatz, Galvin and Gagne, “Operating Systems Concepts”, Wiley.

2. William Stallings, “Operating Systems: Internals and Design Principles”, 6th Edition, Pearson Education.

3. D M Dhamdhere, “Operating Systems: A Concept based Approach”, 2nd Edition, TMH.

Thank You.