M.S. Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)
Department of Computer Science and Engineering

Course Name: Operating Systems
Course Code: CS51
Credits: 3:1:0

Term: September – December 2020

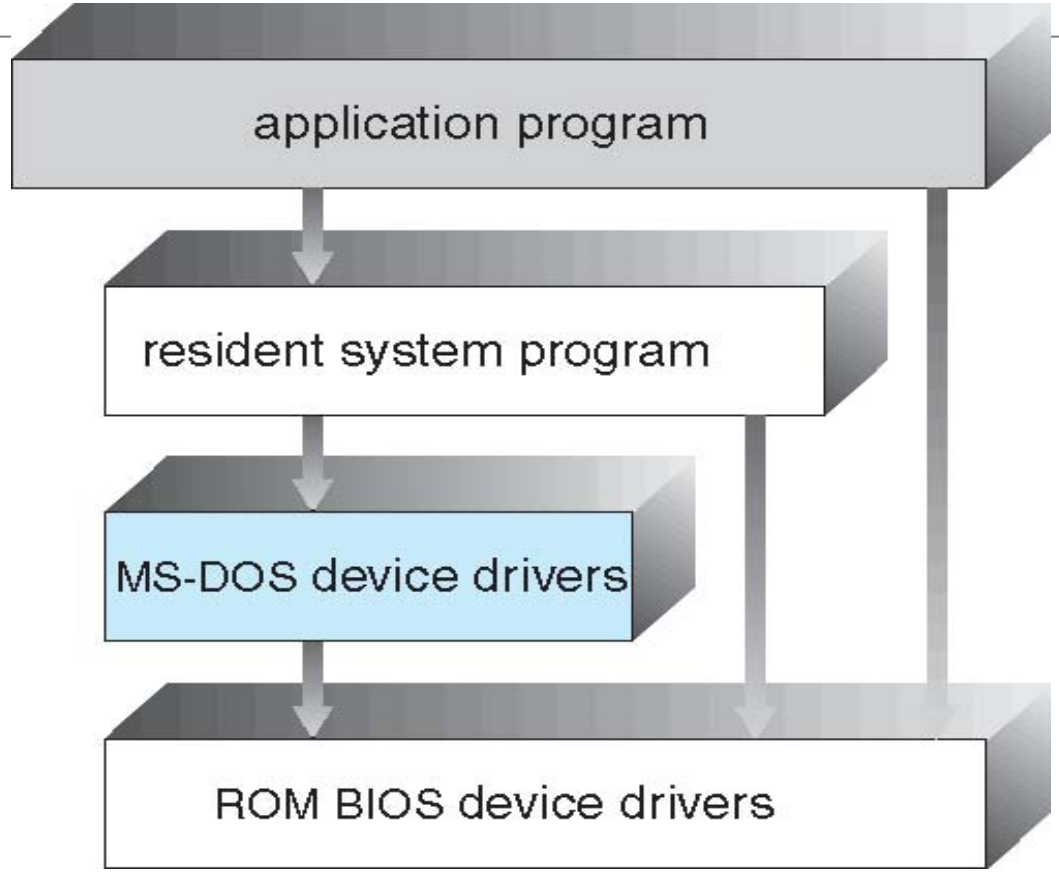
Faculty:
Chandrika Prasad
Vandana S Sardar

OS Structure:

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

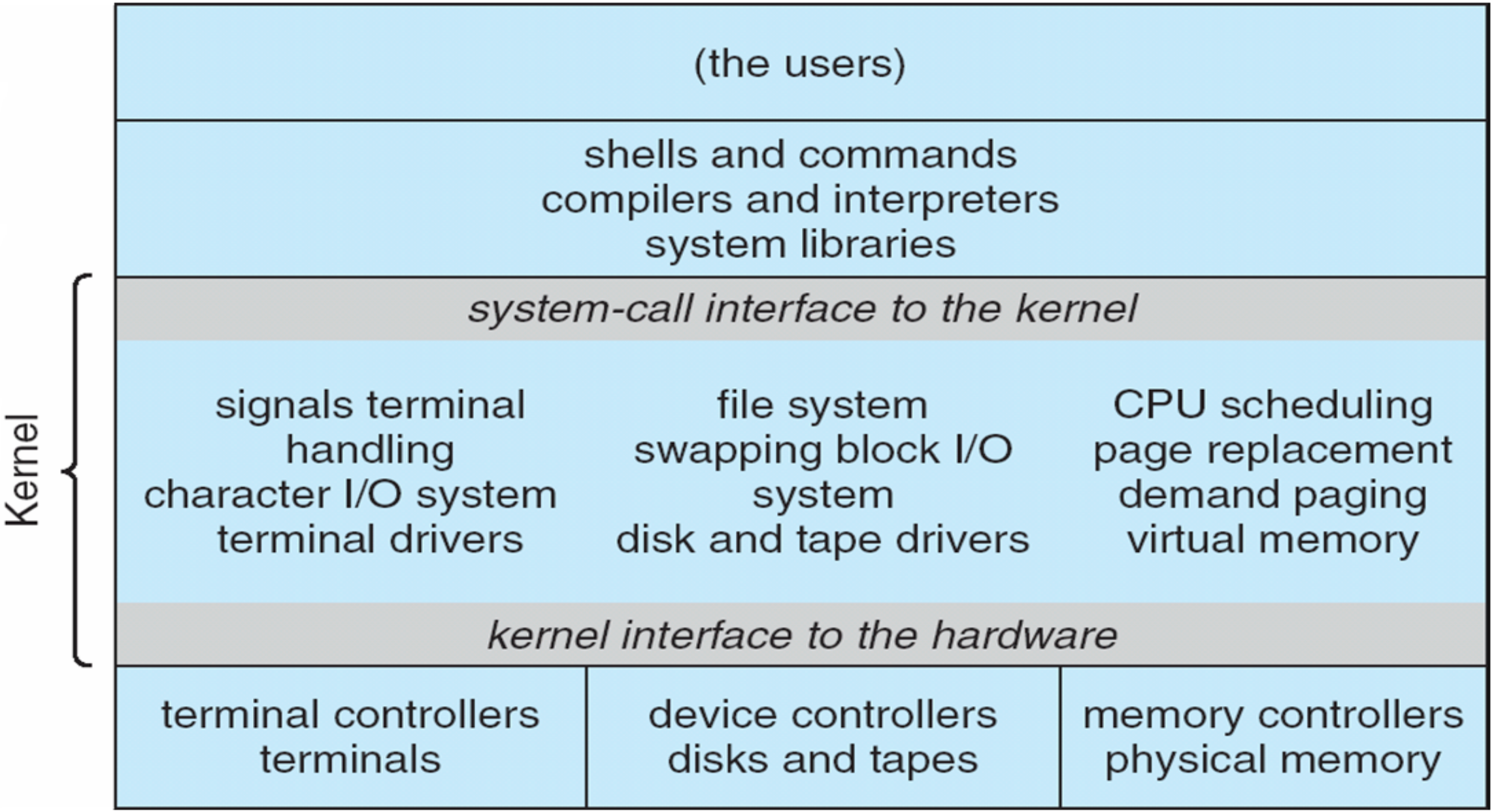
Not divided into modules

Although MS-DOS has some structure, its interfaces and levels of functionality are not well separated



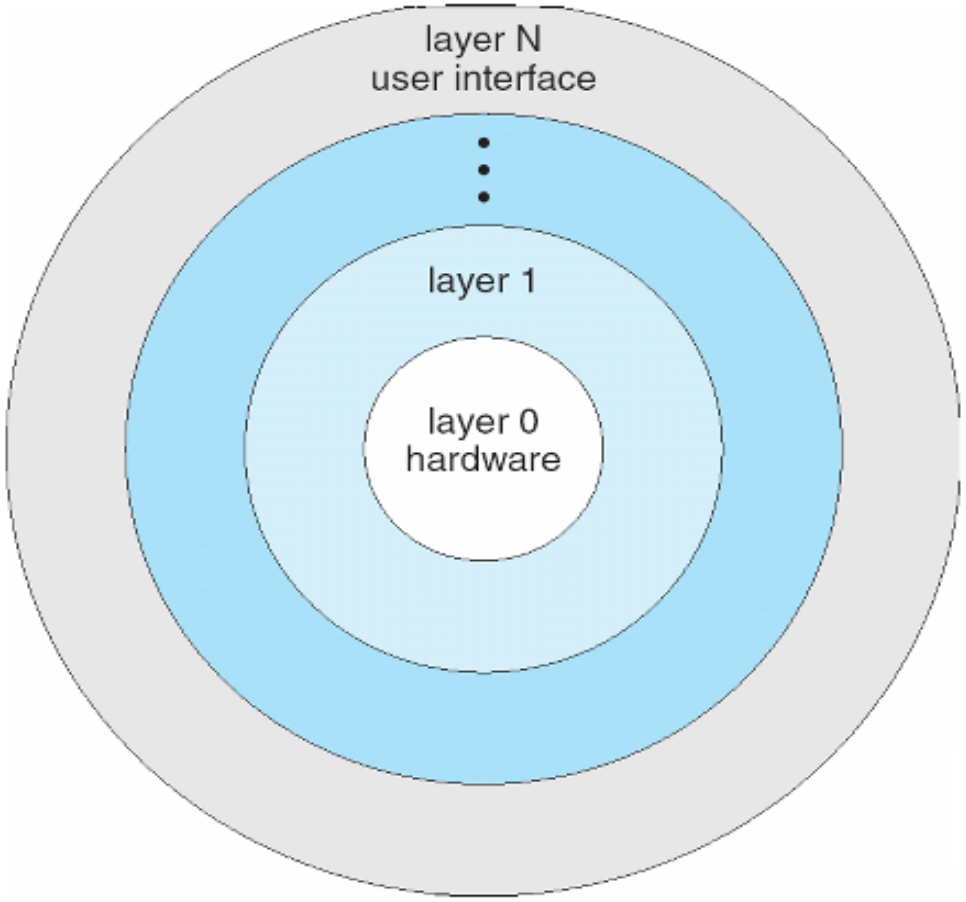
Traditional UNIX System Structure:

- UNIX – limited by hardware functionality, the original UNIX operating system had limited structuring. The UNIX OS consists of two separable parts
  - Systems programs
  - The kernel
- Consists of everything below the system-call interface and above the physical hardware
- Provides the file system, CPU scheduling, memory management, and other operating-system functions; a large number of functions for one level


(the users)
shells and commands
compilers and interpreters
system libraries
system-call interface to the kernel
signals terminal handling
character I/O system
terminal drivers
file system
swapping block I/O system
disk and tape drivers
CPU scheduling
page replacement
demand paging
virtual memory
kernel interface to the hardware
terminal controllers
terminals
device controllers
disks and tapes
memory controllers
physical memory
Kernel

Layered Approach:

- The operating system is divided into a number of layers (levels), each built on top of lower layers.
- The bottom layer (layer 0), is the hardware; the highest (layer N) is the user interface.
- With modularity, layers are selected such that each uses functions (operations) and services of only lower-level layers



RAMAIAH Institute of Technology

Microkernel System Structure

Moves as much from the kernel into user space

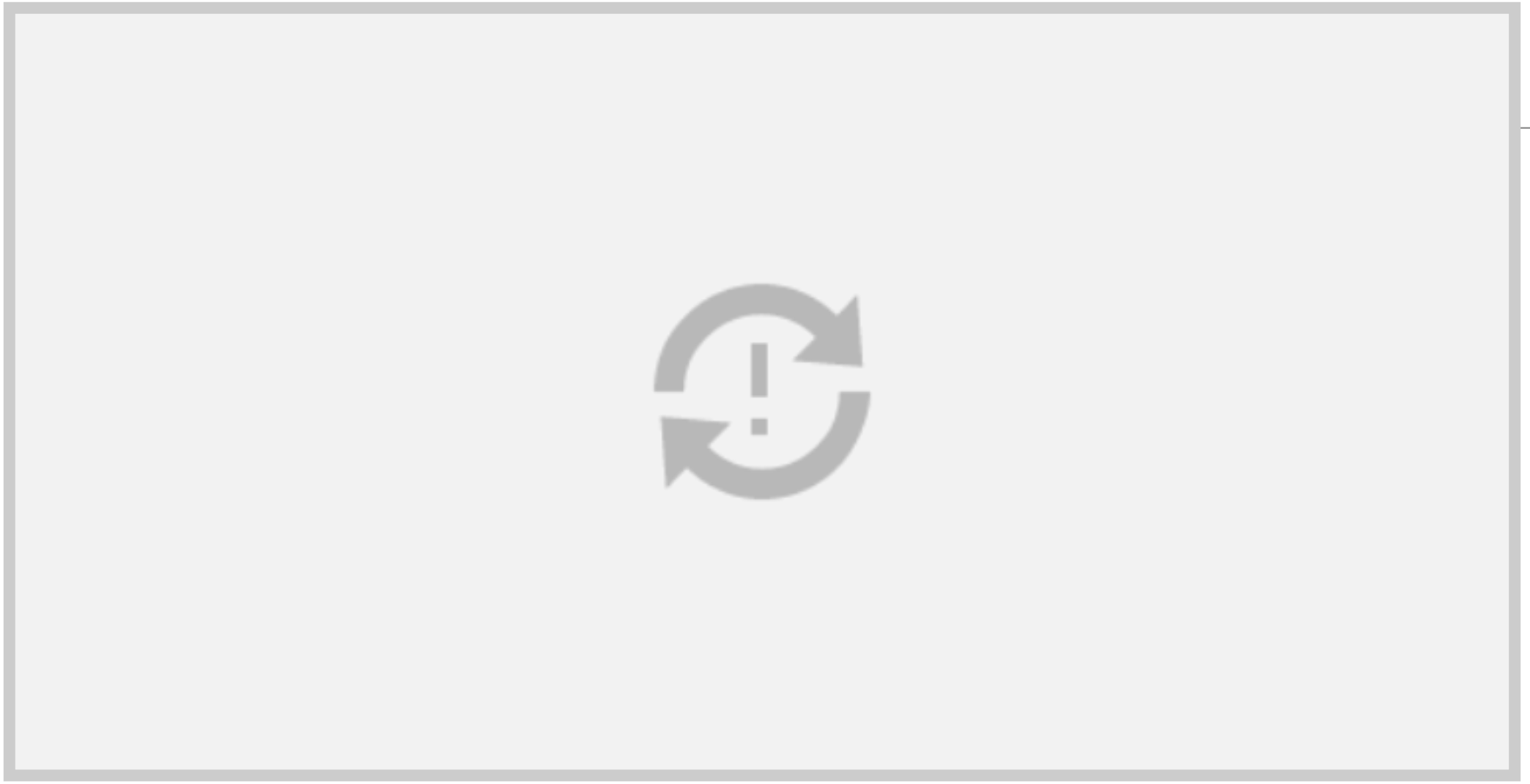
Mach example of **microkernel**

- Mac OS X kernel (**Darwin**) partly based on Mach

Communication takes place between user modules using **message passing**

- Benefits:
  - Easier to extend a microkernel
  - Easier to port the operating system to new architectures
  - More reliable (less code is running in kernel mode)
  - More secure

- Detriments:
  - Performance overhead of user space to kernel space communication



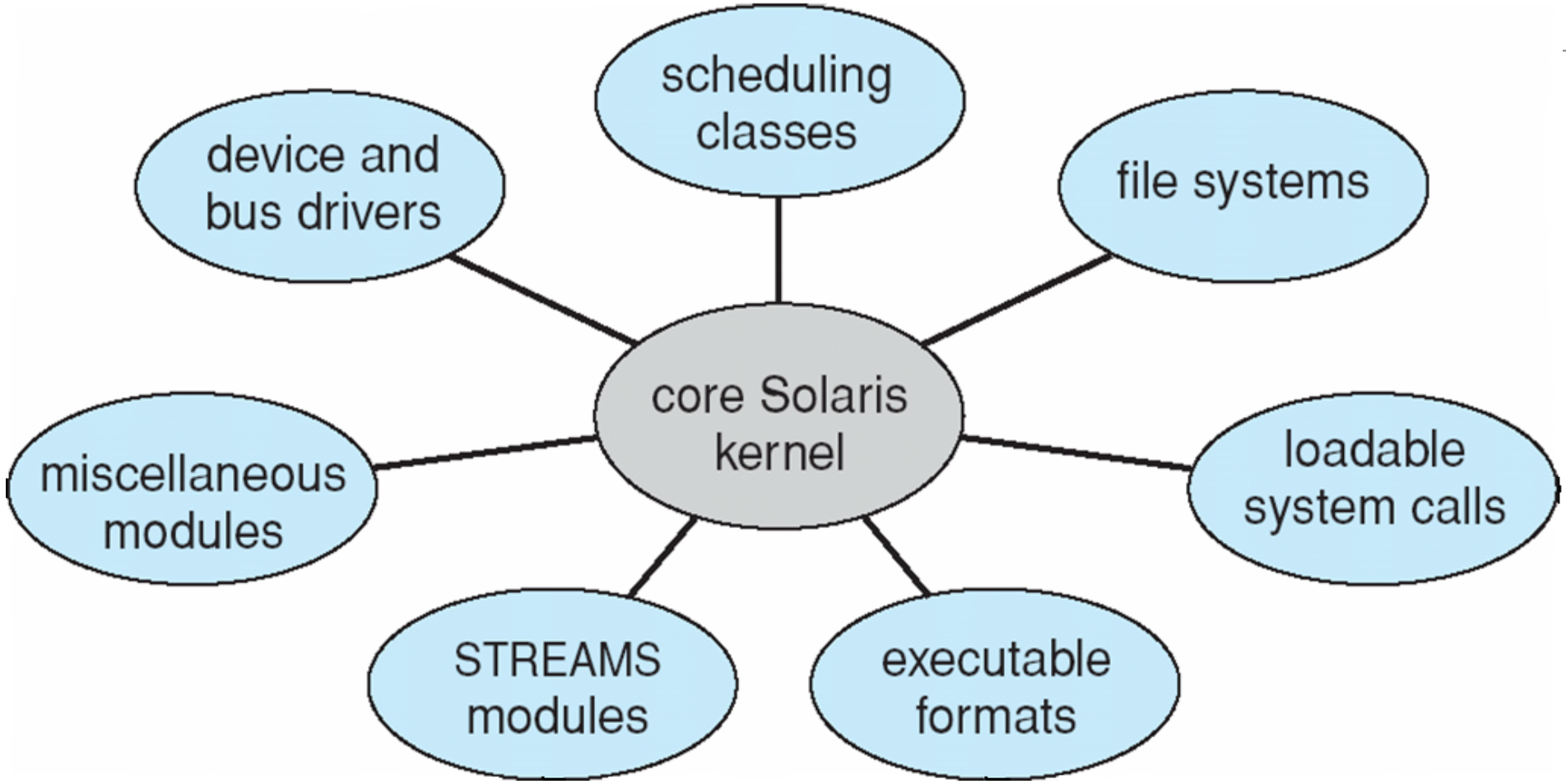
RAMAIAH Institute of Technology

Modules

- Many modern operating systems implement **loadable kernel modules**
  - Uses object-oriented approach
  - Each core component is separate
  - Each talks to the others over known interfaces
  - Each is loadable as needed within the kernel
- Overall, similar to layers but with more flexible
  - Linux, Solaris, etc

RAMAIAH Institute of Technology

Solar Modular Approach



RAMAIAH Institute of Technology

OS Tutorial-2

1. What is the purpose of system calls?
2. What are the five major activities of an operating system with regard to process management?
3. What is the purpose of the command interpreter? Why is it usually separate from the kernel?
4. What system calls have to be executed by a command interpreter or shell in order to start a new process?
5. What is the purpose of system programs?
6. What is the main advantage of the layered approach to system design? What are the disadvantages of using the layered approach?

7) Why do some systems store the operating system in firmware, while others store it on disk?

8) Discuss the structure of Mac OS X with a diagram

9) What is the main difficulty that a programmer must overcome in writing an operating system for a real-time environment?

10) Which of the following instructions should be privileged?

a. Set value of timer.

b. Read the clock.

c. Clear memory.

d. Issue a trap instruction.

e. Turn off interrupts.

f. Modify entries in device-status table.

g. Switch from user to kernel mode. h. Access I/O device.

Thank you