L02a: OS Structure Overview

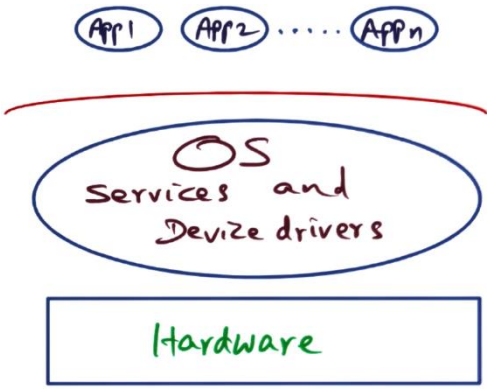
Introduction:

- An operating system has two main responsibilities:
  1. Protect the integrity of the hardware resources.
  2. Provide services to the applications.
- If we think about these two roles, a couple of questions must come to mind:
  1. Which components of the OS will run in a privileged mode (Access to HW)?
  2. Can we make the OS flexible so that it provides personalized services to the applications?
  3. What is the effect of this flexibility on performance and safety?
- OS Services:
  1. Process/thread management and scheduling.
  2. Memory management.
  3. Inter-process communication.
  4. File system.
  5. Access to I/O devices.
  6. Access to the network.

OS Structure:

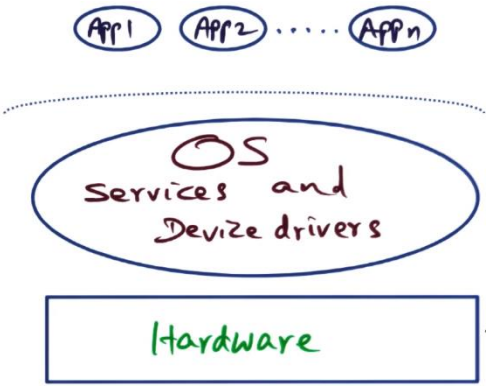
- The OS structure is the way the OS is organized with respect to
  1. The applications that it serves.
  2. The underlying HW that it manages.
- The OS structure affects the following:
  - Protection: The user/system and user/user relationships.
  - Performance: Time taken to perform services.
  - Flexibility: To what degree the OS is extensible.
  - Scalability: Performance improves with more HW resources.
  - Agility: Adapting to changes of resources or application needs.
  - Responsiveness: Time to react to external events.

Monolithic Structure:



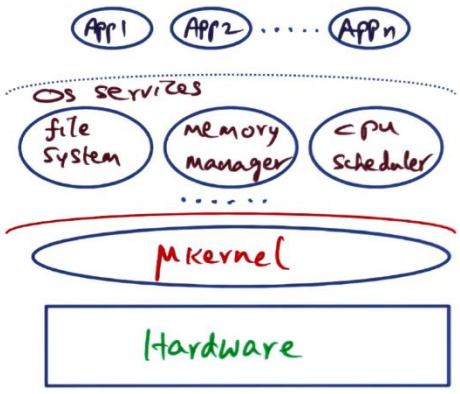
- In a monolithic OS, all the OS services (Files system, CPU Scheduling, Virtual Memory Management, etc.) as well as the OS core functionalities (IPC, Address Space, etc.) are located **within** the kernel. Everything is under the same address space.
- Each application has its own address space, which is separate from the OS address space. This ensures that applications cannot affect each other, and a malfunction of a specific application cannot affect the OS itself → **Higher protection.**
- Whenever an application needs a service from the OS, we have to switch between the application's address space to the OS's address space.
- A Monolithic OS is not customizable → **Lower flexibility.**

MS-DOS Structure:



- The only difference between the Monolithic structure and MS-DOS is that the separation between the application and the OS is less strict in MS-DOS. The application and the OS share the same address space → **Higher performance.**
- In MS-DOS, access to system services is like a procedure call → **Lower protection.**
- An application can access the OS system very quickly. On the other hand, this compromises the integrity of the OS.

Microkernel-based Structure:



- In a Microkernel-based structure, the OS core functionalities (µKernel) will have its own address space, running in a privileged mode.
- The OS services will reside in another address space, separate from the µKernel, and separate from each other.
- Each application has its own address space, which is separate from the OS address space.
- The µKernel will have mechanisms to allow HW access.
- Allow for customization of services because they are implemented as service processes, we can have different replicas of the same service → **Higher flexibility.**

- In a Microkernel-based structure, the application will have to go through the µKernel to make a system call (through an IPC calls). This adds the costs of border crossings (going across different address space), in addition to copying data between address spaces → **Lower performance.**
- The performance issue in µKernel can be mitigated be careful implementation (L3-µKernel).

Feature	Monolithic OS	MS-DOS OS	µKernel OS
Extensibility		✓	✓
Protection	✓		✓
Performance	✓	✓	