MPP:

MPP (massively parallel processing) is the coordinated processing of a program by multiple processors that work on different parts of the program, with each processor using its own operating **system** and memory . Typically, **MPP** processors communicate using some messaging interface.

- ***What is kernel?***

A **kernel** in that context is something from which the rest grows. You could also call a **kernel** a "core", but botanically-speaking a "core" contains seeds (or "**kernels**"). As the rest of the operating systems grows from the **kernel**, the term makes sense to me.

A **kernel** is the central part of an operating system. It manages the operations of the computer and the hardware, most notably memory and CPU time. ... A micro **kernel**, which only contains basic functionality; A monolithic **kernel**, which contains many device drivers.

- ***What is monolithic kernel?***

Monolithic kernel means that the whole operating system runs in **kernel** mode (i.e. highly privileged by the hardware). That is, no part of the OS runs in user mode (lower privilege). Only applications on top of the OS run in user mode.

- ***Which kernel is used in Linux?***

Different Types of Kernels

In general, most kernels fall into one of three types: monolithic, microkernel, and hybrid. **Linux is a monolithic kernel** while **OS X** (**XNU**) and Windows 7 use hybrid kernels.

- ***Why Linux is monolithic kernel?***

Monolithic kernel means that the whole operating system runs in **kernel** mode (i.e. highly privileged by the hardware). That is, no part of the OS runs in user mode (lower privilege). Only applications on top of the OS run in user mode.

- ***Is Windows 10 monolithic kernel?***

Like most Unix systems, **Windows** is a **monolithic** operating system. Because the **kernel** mode protected memory space is shared by the operating system and device driver code.

- ***What is difference between kernel and operating system?***

Operating system is a **system** software. **Kernel** is a part of **operating system**. **Operating system** acts as an interface **between** user and hardware. **Kernel** acts as an interface **between** applications and hardware.

❖ *What are the drawbacks of monolithic kernel?*

One of the major **disadvantage** of **monolithic kernel** is that, if anyone service fails it leads to entire system failure. If user has to add any new service. User needs to modify entire operating system.

Drawbacks of Monolithic Architecture:

- This simple approach has a limitation in size and complexity.
- Application is too large and complex to fully understand and made changes fast and correctly.
- The size of the application can slow down the start-up time.
- You must redeploy the entire application on each update.
- Impact of a change is usually not very well understood which leads to do extensive manual testing.
- Continuous deployment is difficult.
- Monolithic applications can also be difficult to scale when different modules have conflicting resource requirements.
- Another problem with monolithic applications is reliability. Bug in any module (e.g. memory leak) can potentially bring down the entire process. Moreover, since all instances of the application are identical, that bug will impact the availability of the entire application.
- Monolithic applications has a barrier to adopting new technologies. Since changes in frameworks or languages will affect an entire application it is extremely expensive in both time and cost.

❖ *Is Unix monolithic?*

Unix is a **monolithic** kernel because it all the functionality is compiled into one big chunk of code, including substantial implementations for networking, file systems, and devices. Unix is the most influential operating system ever. Micro kernels, for instance, attempt to eliminated networking and file systems from the kernel, loading them in as auxiliary software elements after boot. Ironically, micro kernels are much larger and more complicated than the Unix monolithic kernel, hence have not yet displaced the Unix kernel.

❖ ***Is Unix a kernel or OS?***

UNIX is an **OS**. There is no specific **UNIX kernel** which is available separately since **UNIX OS** was released with the **kernel**, shell and **OS** utilities. **UNIX** developers developed the whole **OS** as one entity (though it had distinct parts but never to be used alone by some other **OS** or to be distributed separately).

❖ ***What is microkernel OS?***

In **computer** science, a **microkernel** (often abbreviated as μ-kernel) is the near-minimum amount of software that can provide the mechanisms needed to implement an **operating system** (**OS**). These mechanisms include low-level address space management, thread management, and inter-process communication (IPC).

Layered Operating System: 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.

❖ ***What are the layers of operating system?***

Layers in Layered Operating System

- Hardware. This layer interacts with the system hardware and coordinates with all the peripheral devices used such as printer, mouse, keyboard, scanner etc.
- CPU Scheduling.
- Memory Management.
- Process Management.
- I/O Buffer.
- User Programs.

❖ ***What are the advantages of layered structure over monolithic structure?***

The main **advantage** of the **layered** approach is simplicity of **construction** and debugging. The **layers** are selected so that each uses functions (operations) and services of only lower-level **layers**. This approach simplifies debugging and system verification.

With the layered approach, the bottom layer is the **hardware**, while the highest layer is the user interface. The main advantage is **simplicity** of construction and debugging. The main difficulty is defining the various layers. The main **disadvantage** is that the OS tends to be less efficient than other implementations.

- ***What are the drawbacks for layers?***

Drawbacks of a Layered Architecture:

Lack of inbuilt scalability: The principles of **layered** architecture hinder the growth of your project as it does not help to scale your project.

Hidden use cases: It is difficult to determine the use cases of your project by simply checking the code organization.