دلیل وجود چندین مدل مختلف برای به اشتراک گذاری مموری چیست؟ هرکدام چه مزایا و معایبی دارند؟ هریک از آنها را از جهات سادگی، کارایی و دیگر موارد با یکدیگر مقایسه کنید.

1. **Shared Memory Model :**
   In this IPC model, a shared memory region is established which is used by the processes for data communication. This memory region is present in the address space of the process which creates the shared memory segment.The processes who want to communicate with this process should attach this memory segment into their address space.Shared memory region is used for communication.It is used for communication between processes on a single processor or multiprocessor systems where the communicating processes reside on the same machine as the communicating processes share a common address space.The code for reading and writing the data from the shared memory should be written explicitly by the Application programmer.It provides maximum speed of computation as communication is done through shared memory so system calls are made only to establish the shared memory.Here the processes need to ensure that they are not writing to the same location simultaneously.Faster communication strategy.

2. Message Passing Model :
In this model, the processes communicate with each other by exchanging messages. For this purpose a communication link must exist between the processes and it must facilitate at least two operations send (message) and receive (message). Size of messages may be variable or fixed.Message passing facility is used for communication.It is typically used in a distributed environment where communicating processes reside on remote machines connected through a network.No such code required here as the message passing facility provides mechanism for communication and synchronization of actions performed by the communicating processes.It is time consuming as message passing is implemented through kernel intervention (system calls). It is useful for sharing small amounts of data as conflicts need not to be resolved.Relatively slower communication strategy.
geeksforgeeks.org

تحقیق کنید آیا میتوان Mutex را بین چند پردازه به اشتراک گذاشت؟ توضیح دهید
the common way of using a process-shared mutex is the following: allocate a block of shared memory, initialize a pthread mutex on the shared memory block, use it.
stackoverflow.com

آیا میتوان با گرفتن فضایی از فضای Heap، آن را بین پردازه های مختلف به اشتراک گذاشت؟ اگر بله، توضیح دهید چرا و یک نمونه کد برای آن بیاورید. درغیراین صورت، با ذکر دلیل راهکار جایگزین پیشنهاد دهید.

Sharing heaps across processes
Shared heaps are specified with the D3D12_HEAP_FLAG_SHARED member of the **D3D12_HEAP_FLAGS** enum.
Shared heaps are not supported on CPU-accessible heaps: D3D12_HEAP_TYPE_UPLOAD, D3D12_HEAP_TYPE_READBACK, and D3D12_HEAP_TYPE_CUSTOM without D3D12_CPU_PAGE_PROPERTY_NOT_AVAILABLE.
Precluding a non-deterministic choice of undefined texture layout can significantly impair deferred rendering scenarios on some GPUs, so it is not the default behavior for placed and committed resources. Deferred rendering is impaired on some GPU architectures because deterministic texture layouts decrease the effective memory bandwidth achieved when rendering simultaneously to multiple render target textures of the same format and size. GPU architectures are evolving away from leveraging non-deterministic texture layouts in order to support standardized swizzle patterns and standardized layouts efficiently for deferred rendering.
Shared heaps come with other minor costs as well: Shared heap data cannot be recycled as flexibly as in-process heaps due to information disclosure concerns, so physical memory is zero'ed more often.There is a minor additional CPU overhead and increased system memory usage during creation and destruction of shared heaps.
docs.microsoft.com