**Address space**

The address space term is an overload term that can have different meanings in different contexts.

The range of virtual address that the operating system assigns to a user or separately running programs is called an **address space**. This is the area of contiguous virtual addresses available for executing instructions and storing data.

The physical address space refers to the way the RAM and device memories are visible on the memory bus. For example, on 32bit Intel architecture, it is common to have the RAM mapped into the lower physical address space while the graphics card memory is mapped high in the physical address space.

The range of virtual address in an address space starts at zero and can extend to the highest address permitted by the operating system architecture.

The OS provides each use with a unique address space and maintains the distinction between the programs and data belonging to each address space. Within each address space, the user can start multiple tasks, using task control blocks or TCBs that allow multiprogramming.

The virtual address space (or sometimes just address space) refers to the way the CPU sees the memory when the virtual memory module is activated (sometime called protected mode or paging enabled). The kernel is responsible of setting up a mapping that creates a virtual address space in which areas of this space are mapped to certain physical memory areas.

Related to the virtual address space there are **two** other terms that are often used: process (address) space and kernel (address) space.

The process space is (part of) the virtual address space associated with a process. It is the "memory view" of processes. It is a continuous area that starts at zero. Where the process's address space ends depends on the implementation and architecture.

The kernel space is the "memory view" of the code that runs in kernel mode.

Address space may also denote a range of physical or virtual addresses which can be accessed by a processor. These addresses are also reserved for a process. Each address specifies an entity’s location as a unique identifier of single entities. Also, each computer device or the component is assigned an address space which is a part of the address space of the processor.

The address that is generated by the CPU is commonly referred to as the **Logical Address**. It is basically a virtual address. The logical address is basically the address of an instruction or data used by any program.

The set of all logical addresses that are generated by any program is referred to as**Logical Address Space.**

The address that is loaded into the**memory-address** **register** of the memory is commonly referred to as a **Physical address**. A physical address cannot be accessed by the user directly but the user can calculate the physical address with the help of a **Logical address.**

The user's program mainly generates the logical address and the user thinks that the program is running in this logical address but the program mainly needs physical memory in order to complete its execution.

The set of all**physical addresses** corresponding to the**Logical addresses** is commonly known as**Physical Address Space.**

***Difference between Logical Address and Physical Address***

Let us now cover the differences between the Logical addresses and Physical addresses in the Operating System

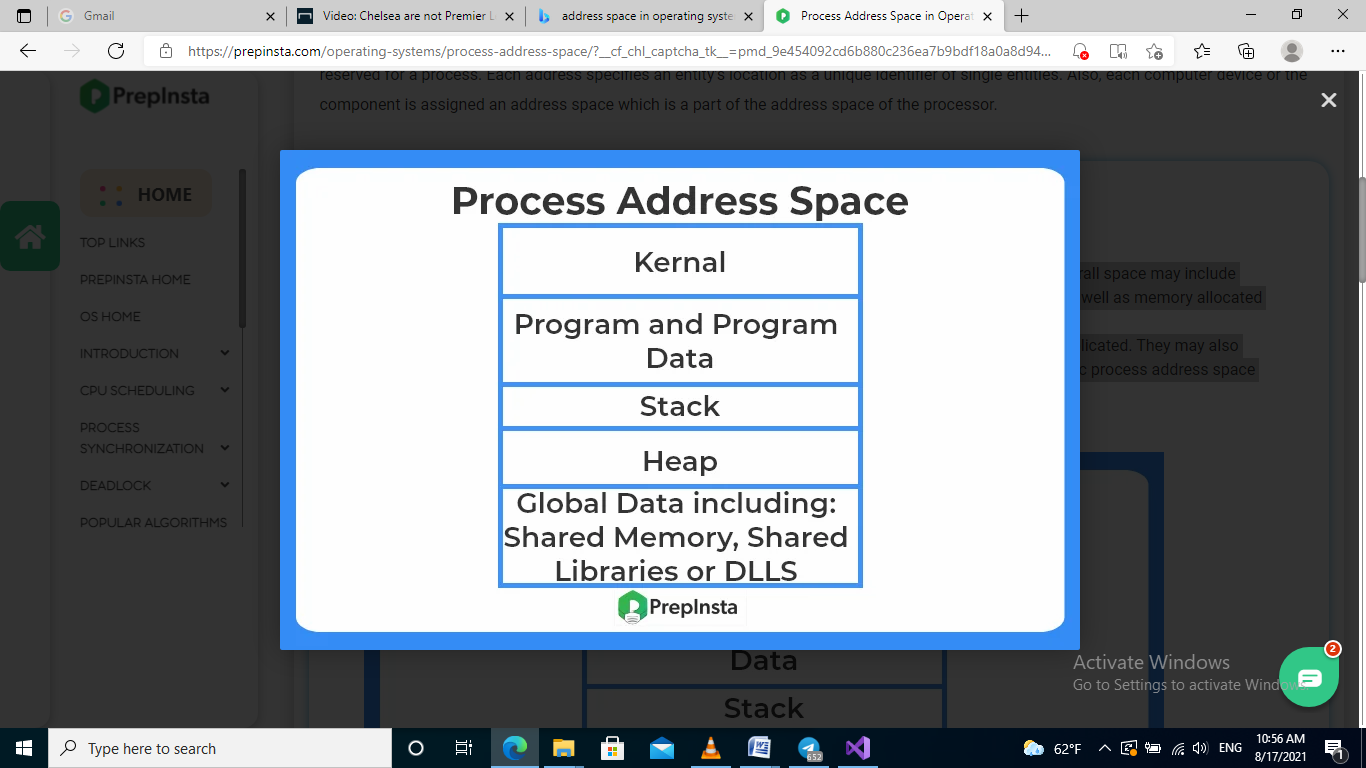
| **No** | **Logical Address** | **Physical Address** |
| --- | --- | --- |
| **1.** | Users can access the **logical address** of the Program. | User can never access the **physical address** of the Program |
| **2.** | The logical address is generated by the CPU. | The physical address is located in the memory unit. |
| **3.** | The user can access the physical address with the help of a logical address. | A physical address can be accessed by a user indirectly b ut not directly. |
| **4.** | The logical address does not exist physically in the memory and thus termed as a Virtual address. | On the other hand, the physical address is a location in the memory. Thus it can be accessed physically. |
| **5.** | The set of all logical addresses that are generated by any program is referred to as**Logical Address Space.** | The set of all**physical addresses** corresponding to the**Logical addresses** is commonly known as**Physical Address Space.** |
| **6.** | This address is generated by the CPU. | It is computed by the Memory Management Unit(MMU). |

**How to access the physical memory location by CPU?**  
Logical Address is used as a reference to access the physical memory location by CPU. The Memory-Management Unit uses the address-binding methods for mapping the logical address to its corresponding physical address.

|  |  |  |
| --- | --- | --- |
| **PARAMETER** | **LOGICAL ADDRESS** | **PHYSICAL ADDRESS** |
| **Basic** | ADDRESS generated by CPU is called LOGICAL ADDRESS. | PHYSICAL ADDRESS is the actual location in a memory unit. |
| **Generation** | generated by the CPU | Computed by MMU |
| **Access** | Logical address can be used to access the physical address by the user. | The user can’t directly access physical address but it is possible indirectly. |
| **Visibility** | Logical address of a program is visible to the user. | Physical address of program is not visible to the user. |
| **Address Space** | Set of all logical addresses generated by CPU are called logical addresses. | Addresses mapped to the corresponding logical addresses. |

## **Components of a Process Address Space**

* The total amount of shared memory a system can allocate depends on several factors. The overall space may include sections such as stack space, program size required, memory mapped files, shared libraries, as well as memory allocated from the heap.
* Memory allocation policies and address spaces used by the varied operating systems are complicated. They may also differ from one operating system to another. The figure below gives an overall **layout** of a generic process address space for a 32-bit operating system.



## Physical Address Space

Physical address space in a system can be defined as the size of the main memory. It is really important to compare the process size with the physical address space. The process size must be less than the physical address space.

Physical Address Space = Size of the Main Memory  
  
If, physical address space = 64 KB = 2 ^ 6 KB = 2 ^ 6 X 2 ^ 10 Bytes = 2 ^ 16 bytes  
  
Let us consider,  
word size = 8 Bytes = 2 ^ 3 Bytes  
  
Hence,  
Physical address space (in words) = (2 ^ 16) / (2 ^ 3) = 2 ^ 13 Words  
  
Therefore,  
Physical Address = 13 bits  
  
In General,  
If, Physical Address Space = N Words  
  
then, Physical Address = log2 N

## Logical Address Space

Logical address space can be defined as the size of the process. The size of the process should be less enough so that it can reside in the main memory.

**Let's say,**

Logical Address Space = 128 MB = (2 ^ 7 X 2 ^ 20) Bytes = 2 ^ 27 Bytes  
Word size = 4 Bytes = 2 ^ 2 Bytes  
  
Logical Address Space (in words) = (2 ^ 27) / (2 ^ 2) = 2 ^ 25 Words  
Logical Address = 25 Bits  
  
In general,  
If, logical address space = L words  
Then, Logical Address = Log2L bits

**References**

<https://prepinsta.com>

<https://www.javatpoint.com/>