Q. 1 Physical Memory and Virtual Memory

Ans:

It is a common word while we are dealing with computer industry, without memory we cannot deal with computer system, sometime we are comparing computer with human being because computer also having memory power to hold data for future reference. But remember any memory in computer has its own physical parts like Ram, Hard Disk etc... But we classified some memory with their functionality like temporary memory, permanent memory. We also classified main memory in parts like main memory, virtual memory, cache memory etc…

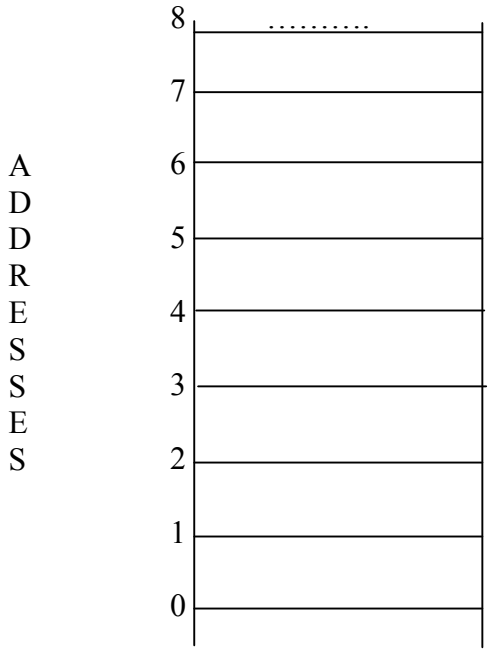
Here we see the classification of two memory like logical memory and physical memory.

- **Physical Memory**

Physical memory is the workhorse of the memory system. It is also known as the main memory. It is internal to the computer system, and comes in the form of chips. The word main is used to distinguish it from external mass storage devices such as disk drives.

Physical memory constructs the middle layer in the memory hierarchy of a computer system. It is slower and cheaper compared to cache memory, while faster and costly compared to disks.

Physical memory generally consists of Random Access memory (RAM). It is volatile, means it needs to have electrical power in order to maintain its information. When power is lost, the information is lost too. By word 'random' means that any piece of data from this memory can be accessed quickly in constant time.



Main/Physical Memory

Generally all the information is stored in disks in form of various files. But access from disk is much slower. So, any information to be used must be brought into main memory from disk before using it. Such information can be directly accessed by the CPU.

Physical memory is expensive compared to disks. So it has limited storage capacity. The capacity available for a given price is increasing all the time. For example many home personal computer now have a capacity of 512 MB or 1 GB.

The entire physical memory can be considered as sequential list of bytes. In most modern computers, each byte has an address that is used to locate it. This address is called physical address.

The above figure depicts the small part of a main memory. Each box in this figure represents a single byte. Each byte has an address. In this figure the addresses are the integers to the left of the boxes: 0,1,2,3,4……. and so on. The addresses for most computer memory start at 0 and go up in sequence until each byte has an address.

●Virtual Memory

Definition: "A virtual memory is a technique that allows a process to execute even though it is partially loaded in main memory"

- Virtual memory is a separation of user logical memory from physical memory
- In this method, we keep only a part of the user's point of view process in the memory and other part on the disk (secondary storage).
- We must need to maintain entire image of process in disk storage.
- Virtual memory removes the requirement that an entire process should be in main memory for its execution. So the main **advantage** of this is: a process can be larger than the main memory even if it execute properly.
- Here **logical addresses** are referred as **virtual addresses**. Also, logical address space of a process is referred as virtual address space.

Advantages:

1. Only Part of the program needs to be in memory for execution
2. Logical address space is much larger than physical address space.
3. Need to allow pages to be swapped in and out.

Implementation:

The virtual memory can be implemented in one of the following three ways:

1. Demand paging
2. Demand Segmentation

3. Segmentation with Paging

In demand paging, pages are moved to memory as per requirement. Similarly, in demand segmentation, segments are moved only as per requirement. Here, on all pages or all segments are moved to memory together at the same time. In segmentation with paging scheme, segments are divided into pages.

Demand paging

- In page is brought into the memory for its execution only when it in demanded.
- Demand paging is required when complete process should be in disk in the form of page.
- Swap the page only when it needed is called **lazy swapper**
- Demand paging is combination of paging and swapping.

Advantages

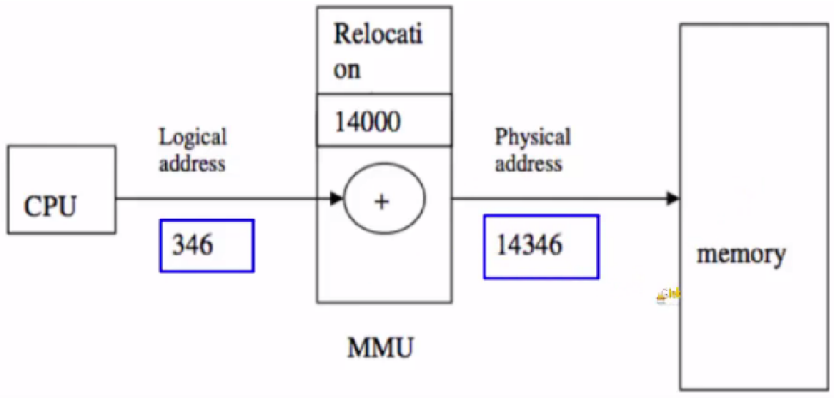
1. reduces memory requirement
2. swap time is also reduce
3. it increases degree of multiprogramming

Disadvantages

1. Occurrence of Page fault. (Page fault is whenever any process execute in memory and one particular page required in memory which not available in the main memory. it is used in performance measurement)

Q-2 What is Logical and Physical Addresses Space in Operating System?

- An address generated by the CPU is commonly referred to as a logical a logical address.
- Whereas, an address seen by the memory unit- that is, the one loaded into the memory-address register of the memory- is commonly referred to as physical address.



- Logical address space is generated from CPU: it is bound to a separate physical address space that is central to proper memory management.
- Physical address space is seen by the memory unit.
- Logical address space is virtual address space.
- Both these address space will be same at compile time but differ at execution time.

Q. 3 Difference Logical Address V/s Physical Address

● **Logical Address**

1. The process address space can be considered as a sequential list of bytes. Each byte has an address that is used to locate it. These addresses are called logical addresses.
2. Logical addresses are generated by the CPU. This means that CPU determines the location of each instruction and data in process address space.
3. Logical address space (LAS): It is a set of all logical addresses that can be referenced by a process.
4. Here, addresses in LAS starts from zero and goes up to some maximum value based on the size of the process.
5. Process can read and write addresses of its own logical address space.
6. The logical addresses are limited by the address size of the processor. For example, 32-bit processor can generate up to 232 (4 GB) addresses.

● **Physical Address**

1. The entire physical memory can be considered as a sequential list of bytes. Each byte has an address that is used to locate it. These addresses are called physical addresses.

2. Physical addresses are seen by the main memory.

3. Physical Addresses Space (PAS): it is a set of all physical addresses occupied by a process in main memory during its execution.

4. Here, address in PAS may or may not be contiguous based upon the memory allocation method.

5. Process can read and write only those physical addresses that belong to its own physical address space.

6. The physical addresses are limited to the amount of installed memory.

For any process to get executed, it should be in main memory. For these reason, whenever process (code, data, and stack) is fetched in main memory, the required storage space is allocated for that process. Now, here the physical addresses allocated to the process may not exactly match the logical addresses of the process. For this reason, there should be some mapping between physical addresses and logical addresses.

Q. 3 Memory Allocation (Contiguous Memory Allocation and Non Contiguous Memory Allocation?

User programs are stored in main memory before execution. Also, any data to be used is stored in main memory. In a similar way, operating system also needs to be stored in main memory.

For all these purposes, there is a need of memory allocation. Memory should be allocated to various processes and files (data) as per requirements. Memory allocation refers to the operations of allocating memory.

Also when there is no need, allocated memory should be freed.

Memory has a limited capacity compared to disk. And, it is possible that more than one process may be executing simultaneously. So, memory should be allocated carefully among all these processes.

While allocating memory, the two goals should be fulfilled:

1. High Utilization

- Maximum possible memory should be utilized
- No any single piece of memory should be wasted.

2. High Concurrency:

- Maximum possible processes should be in main memory
- As there are no more and more processes in memory, CPU will remain busy most of the time in executing one of the process; resulting in better thoughput.

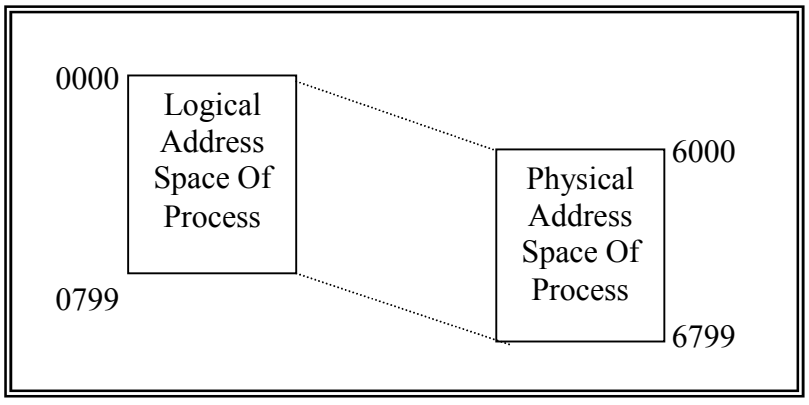
Basically, there are two different ways in which memory is allocated. These are described below.

1. Contiguous Memory Allocation
2. Non-Contiguous Memory Allocation

1. Contiguous Memory Allocation

This is simple and old method of memory allocation. Generally, it is not used in modern operating system. Here, each process occupies a block of contiguous memory locations in main memory. Entire process is kept together in a contiguous section of memory. When a process is brought in main memory, a memory is searched to find out a chunk of free memory having enough size to hold a process. Once such chunk is found, required memory is allocated. If a contiguous memory space of the required size is not available in the main memory, the process is made to wait until contiguous space of the required size is available. Here, logical address space is not divided into any partitions. Also, physical address space will be contiguous, without any gaps.

The following figure clears out this allocation method.



Contiguous Memory Allocation

As described in figure, in contiguous memory allocation, each process occupies a block of contiguous memory locations in main memory. Here, entire process is kept together in a contiguous section of memory.

There are three different way in which contiguous memory allocation method can be used.

1. Single Process Monitor
2. Multiprogramming with Fixed Partitions
3. Multiprogramming with Dynamic Partitions

1. Single Process Monitor

Here, only single process is allowed to run. No more than one process can run simultaneously. This is the simplest possible memory management scheme. Memory is shared between the process and the operating system.

2. Multiprogramming with Fixed Partitions

This method allows multiple processes to execute simultaneously. Here, memory is shared among the operating system and various simultaneously running processes. Multiprogramming increases the CPU Utilization. CPU can be kept busy almost all the time by keeping more than one process simultaneously in the memory.

3. Multiprogramming with Dynamic Partitions

This method is also allows multiple processes to execute simultaneously. Here, memory is shared among operating system and various simultaneously running processes. Here, memory is not divided into any fixed size partitions. Also the number of partitions is not fixed. Process is allocated exactly as much memory as it requires. Initially, the entire available memory is treated as a single free partition.

Advantages

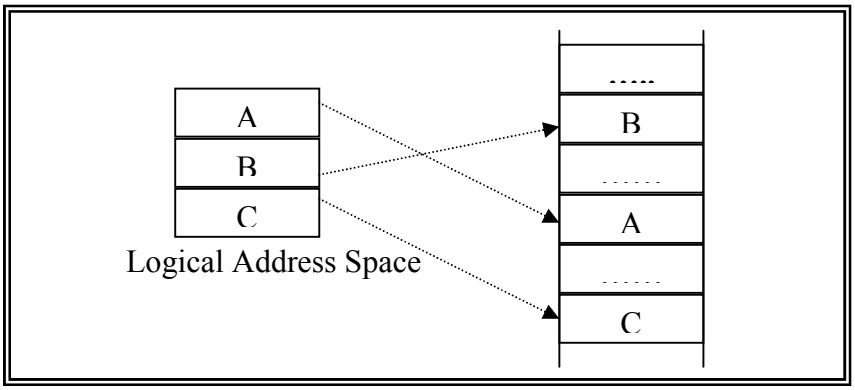
1. It is easy to implement.
2. It is easy to understand.

Disadvantages

1. Having poor memory utilization.

Non-Contiguous Memory Allocation

This method is used by most modern operating system. Here, logical address space of a process is divided into partitions. And, for each partition contiguous chunk of free memory is allocated. Physical address space will not be contiguous now. The following figure clears out this allocation method. Here, logical address space of a process is divided into three partitions (A,B and C) and each partition is allocated separate chunk of memory in physical memory.



Non-Contiguous Memory Allocation

Physical address space of a process will not be contiguous now. Physical addresses will be scattered over entire memory. There are two kind of allocation techniques used here: One is paging and other is segmentation.

Paging:

Here, the logical address space of a process is divided into blocks of fixed size, called pages. Also, the physical memory is divided into blocks of fixed size, called Frame or Page Frame.

Segmentation

This technique implements user's view of logical address space. From user's point of view, logical address space of any process is a collection of code, data and stack. Code can be comprise of main function, other user defined functions and

library function. Data can be local variables, global variables, arrays, symbol table and other data structures.

Here, the logical address space of a process is divided into blocks of varying size, called segment. Each segment contains a logical unit of a process. Logical unit can be main function, other functions or procedures, stack, array, symbol table etc…

Advantages of Non-Contiguous Memory Allocation

Having better memory utilization

Disadvantages of Non-Contiguous Memory Allocation

It is complex to implement.

Q. 4 Explain Non-Contiguous Memory Allocation: Paging

Most of modern operating systems use this method for allocation memory. In this method, logical address space of a process is divided into partitions. For each partition contiguous large piece of free memory is allocated in physical memory.

Paging

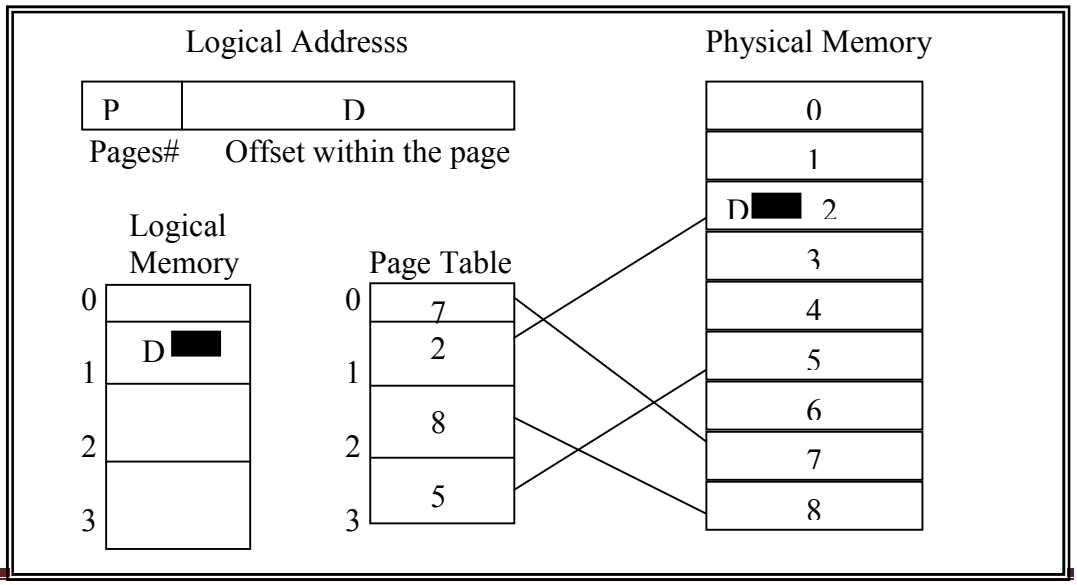
- Paging is non-contiguous memory allocation**.**
- Allows the Physical addresses of a process to be non-contiguous.
- Logical address space is divided into blocks of fixed size, called Pages.
- Physical address space is divided into blocks of fixed size, called Frames or Page Frames.
- The pages and frames will be of the same size. [page size = frame size] This size will be typically a power of 2. It varies between 512 Bytes to a few MB.

Whenever a process is to be executed, its pages are moved from secondary storage, i.e. a disk, to the available frames in physical memory. So, memory allocation refers to the task of finding free frames in memory, allocating them to pages, and keeping track of which frame belongs to which page.

Operating system maintains a table, called page table, for each process. The page table is indexed by a page number, which gives the number of pages are stored, is kept in this page table.

Here, logical address is divided into two parts i) page number, which gives the number of a page; and ii) an offset, which gives the actual location within a page.

The following figure gives idea about this method. Here, the logical address space of a process, also called logical memory, has been divided into four pages. Physical memory contains total of 11 frames. Also, a page table is described which shows the mapping between pages and frames.



Implementation

The above figure show how the paging scheme is implemented. It describes how logical addresses are converted to physical addresses.

A table, called page tale, is used to implement paging. When a process is to be executed, its pages are moved to free frames in physical memory. The information about frame number, where a page is stored, is kept in this page table.

During the process execution, a CPU generates a logical address (L) to access instruction or data from a particular location. This logical address is divided into two parts: a page number (p) and an offset (d) within a page.

Logical Address (L)

Page Number (p)	Offset (d)

The page number is used to search the page table. Corresponding frame number (f) is obtained from the page table. This frame number indicates the actual frame on physical memory in which the page is stored.

The physical address (P) of a particular location is obtained by combining the frame number (f) with the offset (d).

Physical Address (P)

Frame Number (f)	Offset (d)

Address Translation:

Let the page size (or frame size) be 2^n bytes.

So, the logical address (L) can be divided into two parts, page number (P) and offset (d), such that –

Page Number (p) = L / 2^n, and

Page Offset (d) = L % 2^n.

This page number (p) is used to find out corresponding frame number (f) from page table. Once frame number (f) is found, physical address can be found as –

Physical Address (P) = f * 2^n + d.

Advantages of Paging Method

1. **Allocation and De-Allocation** of memory is very fast. For this, only the list of free frames needs to be maintained. No need of any algorithms such as first fit, best fit, etc.
2. **Swap-in and Swap-out** operations are fast. As page size matches disk block size, data can be moved to and from disk very easily.
3. There is no external fragmentation. Each and every frame from memory can be allocated to processes.

Disadvantages of Paging Method

1. **Additional memory reference** is required to read information from page table. Every instruction or data reference requires two memory accesses: One for page table, and one for instruction or data.
2. **Size of page table** may be too large to keep it in main memory. Pages table contains entry for all the pages in logical address space. For large process, page table will be large.
3. **Internal fragmentation :** A process size may not be an exact multiple of the page size. So, some space would remain unoccupied in the last page of a process, page results in internal fragmentation.

Q. 5 Explain Non-Contiguous Memory Allocation : Segmentation

This is one of the techniques of non-contiguous memory allocation: segmentation. This technique implements user's view of logical address space. From user's point of view, logical address space of any process is a collection of code, data and stack. Code can be comprise of main function, other user defined functions and library function. Data can be local variables, global variables, arras, symbol table and other data structures.

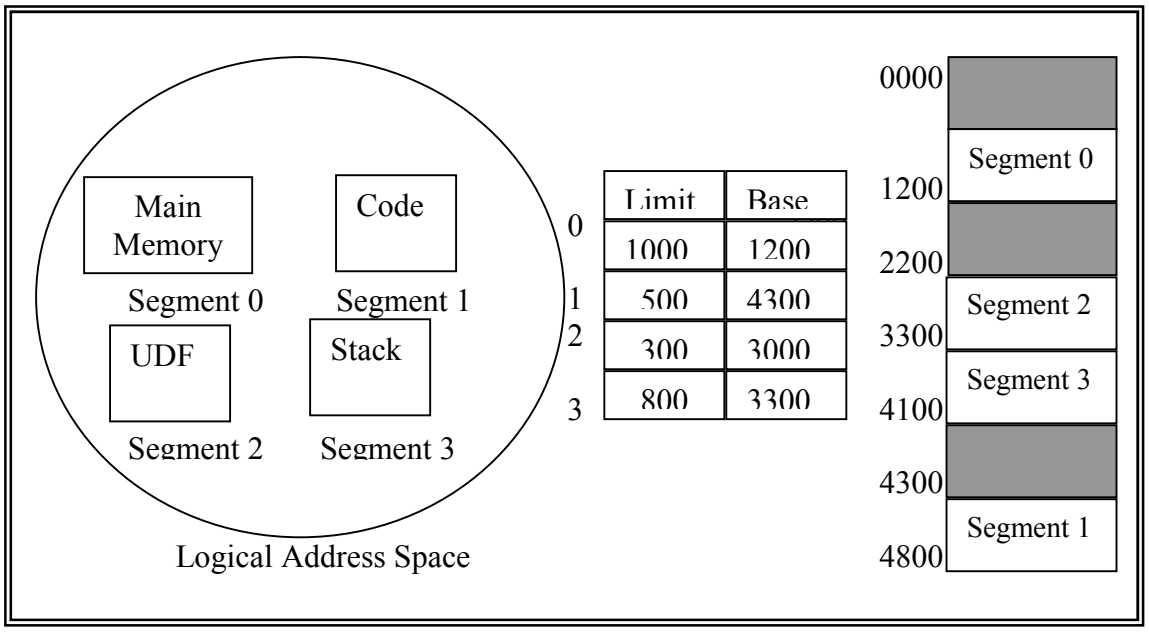
Here, the logical address space of a process is divided into blocks of varying size, called Segments. Each segment contains a logical unit of a process. Logical unit can be main function, other functions or procedures, stack, array, symbol table etc.

Each segment can be considered as a completely independent address space. It consist of linear sequence of addresses starting from '0' to some maximum limit. All segments are of varying lengths and length depends upon the size of a logical unit. All segments are given unique numbers to identify them.

Whenever a process is to be executed, its segments are moved from secondary storage, i.e. a disk, to the main memory. Each segment is allocated a chunk of free memory of the size equal to that segment.

Operating system maintains a table, called segment table, for each process. The segment table contains information about each segment of a process. This information includes size of segment and the location in memory where the segment has been loaded.

Here logical address is divided into two parts: i) segment number, which identifies a segment and ii) and offset, which gives the actual location within a segment



The above figure gives ideas about this method. Here, the logical address space of a process has been divided into four segments. A physical memory is shown with these four segments. A segment table is described which gives information such as where each segment is loaded in physical memory and what is the size (length) of each segment.

Implementation

A table, called segment table, is used to implement segmentation. Each entry of the segment table has a segment base and segment limit. The segment base contains the starting physical address where the segment resides in memory. The segment limit specifies the length of the segment.

During the process execution, a CPU generates a logical address (L) to access instruction or data from a particular location. This logical address is divided into two parts: First, a segment number (s), and Second, and offset (d) within that segment.

Logical Address (L) :

Segment Number (s)	Offset (d)

The segment number is used as an index into the segment tale. The offset (d) of the logical address must be between 0 and the segment limit. If it is not, illegal address error will be generated. Else, if it is within limit, it is added to the segment base to produce the physical address.

Advantages of Non-Contiguous Memory: Segmentation

1. All segments are independent from each other. So, segments can grow or shrink without affecting other segment.
2. If the procedure (or function) in segment 'n' is modified and recompiled, no other segments need to be changed or recompiled.
3. Sharing of procedures and data among various processes simple.
4. Different segments of a single process can be given different kind of protection. One can be read-only, while other can be writable and so on.
5. There is no Internal Fragmentation. Segments are allocated exactly as much memory as required.

Disadvantages Of Non-Contiguous Memory: Segmentation

1. It is still expensive to allocate contiguous free memory to segments. Some algorithm, such as first fit, is required for allocation of a memory.
2. External Fragmentation is possible, which requires memory de-fragmentation or compaction.