Assignment: 4

Ques 1: Explain the logical and physical memory space.Ans:

Logical Memory Space:

Logical memory space refers to the abstract, virtual address space that a process or program sees and uses. It is a conceptual view of memory where each process has its own isolated address space. In the context of a computer system, the logical memory space is typically divided into fixed-size units called pages or segments, depending on the memory management scheme employed by the operating system.

In a system utilizing virtual memory, the logical memory space allows each process to have its own dedicated address space, starting from address 0 and extending to the maximum logical address. This logical address space is independent of the actual physical memory available in the system. It provides an illusion of a large and contiguous memory to each process, even if the physical memory is limited.

Logical memory space provides several benefits, including process isolation, protection, and flexibility. Each process operates within its own logical address space, which prevents it from accessing or modifying memory locations of other processes. The operating system utilizes memory management techniques to map the logical addresses of a process to physical addresses in the physical memory.

Physical Memory Space:

Physical memory space, also known as real memory or main memory, refers to the actual physical hardware components that store data and instructions. It consists of the physical memory chips, such as RAM (Random Access Memory), that are directly accessible by the CPU.

Physical memory is a finite and limited resource in a computer system. It is divided into fixed-size units called frames or pages, which match the size of the logical memory's pages or segments. The physical memory space is responsible for holding the actual data and instructions that are being executed by the CPU at any given time.

The operating system's memory management unit (MMU) is responsible for mapping the logical addresses of a process to the corresponding physical addresses in the physical memory. This mapping allows the CPU to access and manipulate the data stored in the physical memory.

The size of the physical memory is determined by the hardware limitations of the computer system. It plays a crucial role in determining the overall performance and capacity of the system. Insufficient physical memory can lead to performance degradation and the need to rely heavily on secondary storage devices like hard drives, which are slower compared to the main memory.

Ques 2: Explain non-contiguous memory allocation.

Ans: Non-contiguous memory allocation is a memory management scheme where a process's memory is allocated in a fragmented or scattered manner across the physical memory space. In this scheme, a process's memory segments or pages can be located at different non-contiguous locations in the physical memory.

Non-contiguous memory allocation is commonly used in systems that employ virtual memory, where the logical memory space is larger than the available physical memory. The operating system utilizes various techniques to map the logical addresses of a

process to the physical addresses in the physical memory.

There are several methods for implementing non-contiguous memory allocation:

1. Egju: In paging, the logical memory space is divided into fixed-size units called pages, and the physical memory is divided into corresponding fixed-size units called frames. The logical pages of a process can be scattered across different physical frames in the memory. The operating system maintains a page table that maps the logical page numbers to the physicalframe numbers.
2. Segmentation: In segmentation, the logical memory space is divided into variablesized segments, such as code segment, data segment, stack segment, etc. These segments can be scattered across different non-contiguous regions in the physical memory. The operating system maintains a segment table that maps the segment numbers to the physical memory locations.
3. Dynamic Memory Allocation: In dynamic memory allocation, memory is allocated to a process as and when it is needed. This can involve allocating memory in small chunks or blocks from various locations in the physical memory, resulting in non-contiguous memory allocation. Common dynamic memory allocation techniques include the use of heap memory and functions like and 'free()' in programming languages.

Non-contiguous memory allocation allows efficient utilization of the available physical memory by allocating memory only when needed and allowing memory to be freed when no longer required. However, it introduces additional complexity in memory management, as the operating system needs to maintain data structures (e.g., page tables or segment tables) to track thescattered memory allocations of each process.

Ques 3: Explain paging with example. Differentiate between paging and segmentation.

Ans: Paging is a memory management scheme where the logical memory space is divided into fixed-size units called pages, and the physical memory is divided into corresponding fixed-size units called frames. The primary goal of paging is to provide a virtual memory system that allows efficient utilization of the physical memory and enables processes to have larger address spaces than the available physical memory.

To understand paging, let's consider an example:

Assume a system with a 32-bit address space, where each page and frame size is 4 KB (212bytes). This means each page and frame consists of 4,096 bytes.

Logical Memory:

Page size: 4 KB

* Page number: 2A 20 pages (since 32-bit address space)Physical Memory:
* Frame size: 4 KB
* Frame number: 2A20 frames

Page Table: The operating system maintains a page table for each process to map the logical page numbers to the physical frame numbers. The page table is typically stored in the main memory.

For example, let's assume we have a process with a logical address space of 220 pages. Thepage table for this process would have 220 entries, each entry pointing to the corresponding physical frame number where the page is stored in the physical memory.

Address Translation: When a process generates a logical address, the paging system translates it into a physical address using the page table. The logical address consists of a page number and an offset within the page.

For example, let's assume the logical address generated by the process is:

* Page number: 512
* Offset: 2,048

To translate this address, the page table is consulted. If the page table entry for page 512 points toframe number 1, the resulting physical address would be:

* Frame number: 1  Offset: 2,048
* So the physical address becomes: Frame 1 + Offset 2,048 = 1 \* 4,096 + 2,048 = 6,144.

Differences between Paging and Segmentation:

* Memory Division: In paging, both logical and physical memory are divided into fixed-size units called pages and frames, respectively. In segmentation, logical memory is divided into variable-sized segments, whereas physical memory is divided into fixed-size frames.
* Size and Fraqmentation: Paging avoids external fragmentation as it divides memory into fixed-size units. Segmentation, on the other hand, can suffer from external fragmentation due to the variable-sized segments.
* Address Translation: In paging, address translation involves using a page table that maps logical pages to physical frames. In segmentation, address translation involves using a segment table that maps logical segments to physical memory locations.
* Sharinq and Protection: Paging provides a level of memory protection as each page can have its own access permissions. Sharing memory between processes is morestraightforward in paging, as pages can be easily shared. Segmentation allows for more flexible sharing of segments, but it can be more complex to manage.
* Memory Allocation: Paging allocates memory in fixed-size units (pages), which may leadto internal fragmentation. Segmentation allocates memory in variablesized units (segments), which reduces internal fragmentation but can increase externalfragmentation.

In summary, paging and segmentation are different memory management schemes. Paging divides memory into fixed-size units, uses a page table for address translation, and aims to eliminate external fragmentation. Segmentation divides memory into variable-sized units, uses a segment table for address translation, and allows for more flexible sharing and protection but cansuffer from external fragmentation.

Ques 4: Explain the difference between external and internal fragmentation. How paging solvethis problem. Ans: External Fraqmentation:

External fragmentation occurs when free memory blocks are scattered throughout the memory space, but the total amount of free memory is sufficient to satisfy a memory request. It arises dueto the allocation and deallocation of variable-sized memory blocks over time. As a result, the memory becomes fragmented into small, non-contiguous blocks, making it challenging to find a large enough contiguous block to allocate to a process. This fragmentation reduces the overall memory utilization.

Internal Fraqmentation:

Internal fragmentation occurs when allocated memory blocks are larger than what is actually required by a process. It happens when memory is allocated in fixed-size blocks, such as pages or frames, and the allocated block is larger than the actual size needed by the process. As a result,there is wasted memory within each allocated block, reducing the overall memory efficiency.

Paging and Solving Fragmentation:

Paging is a memory management technique that helps address both external and internal fragmentation problems.

I . External Fragmentation:

Paging mitigates external fragmentation by dividing both the logical memory space and physical memory into fixed-size units called pages and frames, respectively. Since pages and frames are of the same size, the entire memory space is uniformly divided into these fixed-size blocks. This eliminates the possibility of large non-contiguous free memory blocks, reducing external fragmentation. When a process requests memory, the memory manager can allocate any available free page, regardless of its physical location, as long as it can find a free page frame. This flexibility of paging allows for efficient utilization of memory and reduces external fragmentation.

2. Internal Fragmentation:

Paging also helps address internal fragmentation by allocating memory in fixed-size pages. Each process's memory is divided into multiple pages, and the memory manager allocates whole pagesto the process. If the memory requested by a process is smaller than a full page, there will still besome internal fragmentation within that page. However, compared to other memory allocation schemes like contiguous allocation, the internal fragmentation is limited to the size of a page rather than the size of the entire memory block. This reduces the overall impact of internal fragmentation and improves memory utilization.

By employing a fixed-size paging mechanism, memory management can efficiently allocate memory to processes without suffering from significant external fragmentation or excessive internal fragmentation. It provides a balance between flexibility and efficient memory utilization, making paging a popular solution to fragmentation problems in modern memory management systems.

Ques 5: Explain the concept of virtual memory.

Ans. Virtual memory is a memory management technique that allows a computer system to use more memory than is physically available by utilizing secondary storage, such as a hard disk. It provides an abstraction layer between the physical memory (RAM) and the logical memory seen by processes, enabling efficient and flexible memory allocation.

The key idea behind virtual memory is to create an illusion of a large and contiguous address space for each process, even if the physical memory is limited. It allows processes to operatewith an address space larger than the available physical memory, providing several benefits:

1. Increased Address Space: Virtual memory allows each process to have its own isolated address space, typically much larger than the physical memory. This enables the execution of large programs and supports memory-intensive applications.
2. Memorv Isolation: Each process operates within its own virtual address space, isolated from other processes. This prevents processes from accessing or modifying memory locations of other processes, ensuring data privacy and protection.
3. Efficient Memory Utilization: Virtual memory enables efficient utilization of physical memory. Only the actively used portions of a process's address space are loaded into the physical memory, while the less frequently used parts can be swapped out to secondary storage. This swapping mechanism is known as paging.
4. Demand Paqinq: Demand paging is a technique used in virtual memory systems where pages of a process are loaded into physical memory only when they are accessed. This allows for fasterprocess startup times and reduces the overall memory requirements, as only the necessary pages are loaded.
5. Memory Manaqement: Virtual memory simplifies memory management for the operating system. It provides a uniform and consistent memory interface to processes, regardless of the physical memory constraints. The operating system is responsible for managing the mapping between logical addresses and physical addresses, paging in/out of memory, and handling page faults.
6. Memory Protection: Virtual memory provides memory protection mechanisms. Each page of aprocess ls address space can be assigned access permissions, such as readonly or no access, ensuring the integrity and security of the system.

Overall, virtual memory allows for efficient utilization of physical memory, provides memory isolation between processes, and enables the execution of large programs. It plays a vital role in modern operating systems, contributing to improved performance, multitasking capabilities, and supporting memory-intensive applications.