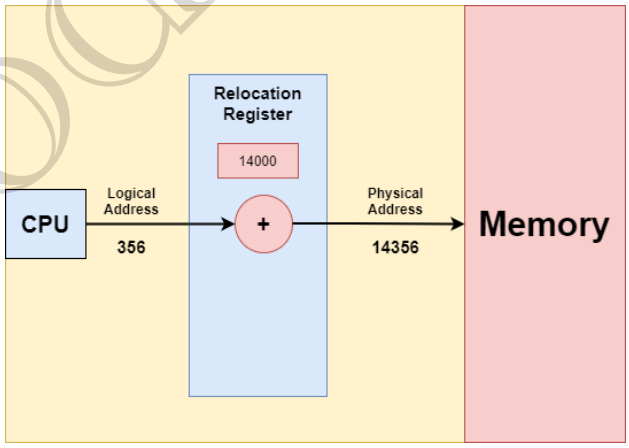
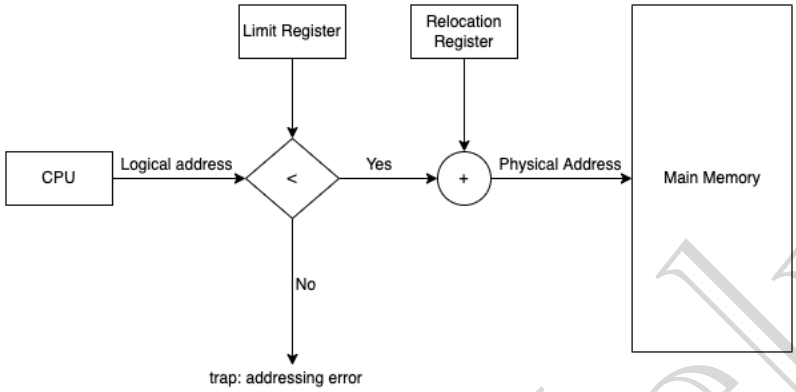
LEC-24: Memory Management Techniques | Contiguous Memory Allocation

1. In Multi-programming environment, we have multiple processes in the main memory (Ready Queue) to keep the CPU utilization high and to make computer responsive to the users.
2. To realize this increase in performance, however, we must keep several processes in the memory; that is, we must **share** the main memory. As a result, we must **manage** main memory for all the different processes.
3. **Logical versus Physical Address Space**
    - **a.** Logical Address
        - **i.** An address generated by the **CPU**.
        - **ii.** The logical address is basically the address of an instruction or data used by a process.
        - **iii.** User can access logical address of the process.
        - **iv.** User has indirect access to the physical address through logical address.
        - **v.** Logical address does not exist physically. Hence, aka, **Virtual address.**
        - vi. The set of all logical addresses that are generated by any program is referred to as Logical Address Space.
        - **vii. Range: 0 to *max*.**
    - b. Physical Address
        - i. An address loaded into the memory-address register of the physical memory.
        - ii. User can never access the physical address of the Program.
        - iii. The physical address is in the memory unit. It's a location in the main memory physically.
        - iv. A physical address can be accessed by a user indirectly but not directly.
        - v. The set of all physical addresses corresponding to the Logical addresses is commonly known as Physical Address Space.
        - vi. It is computed by the **Memory Management Unit (MMU).**
        - **vii. Range: (R + 0) to (R + max), for a base value R.**
    - **c. The runtime mapping from virtual to physical address is done by a hardware device called the memory-management unit (MMU).**
    - d. 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.
    - e.

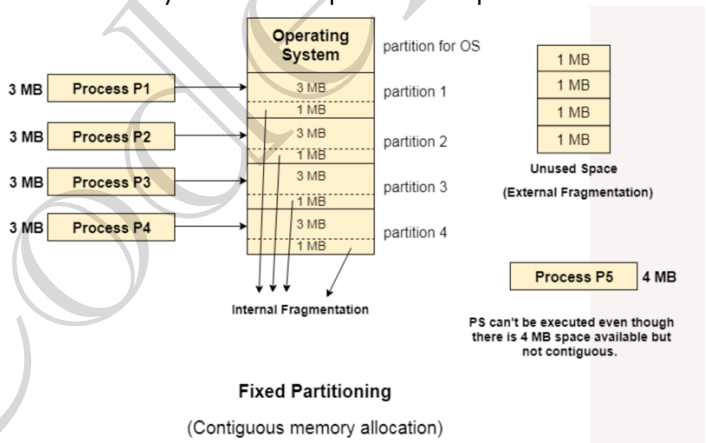


4. How OS manages the isolation and protect? **(Memory Mapping and Protection)**
    - a. OS provides this Virtual Address Space (VAS) concept.
    - b. To separate memory space, we need the ability to determine the range of legal addresses that the process may access and to ensure that the process can access only these legal addresses.
    - c. The relocation register contains value of smallest physical address (Base address [R]); the limit register contains the range of logical addresses (e.g., relocation = 100040 & limit = 74600).
    - d. Each logical address must be less than the limit register.

e. MMU maps the logical address dynamically by adding the value in the relocation register.
f. When CPU scheduler selects a process for execution, the dispatcher loads the relocation and limit registers with the correct values as part of the context switch. Since every address generated by the CPU (Logical address) is checked against these registers, we can protect both OS and other users' programs and data from being modified by running process.
g. Any attempt by a program executing in user mode to access the OS memory or other uses' memory results in a trap in the OS, which treat the attempt as a fatal error.
h. **Address Translation**



5. **Allocation Method on Physical Memory**
    a. Contiguous Allocation
    b. Non-contiguous Allocation
6. **Contiguous Memory Allocation**
    a. In this scheme, each process is contained in a single contiguous block of memory.
    b. Fixed Partitioning
        i. The main memory is divided into partitions of equal or different sizes.

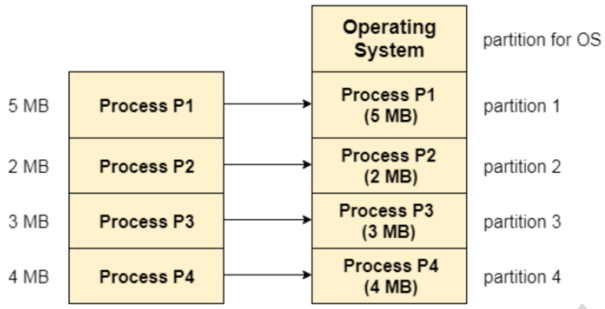


        ii.
        iii. Limitations:
            1. **Internal Fragmentation:** if the size of the process is lesser then the total size of the partition then some size of the partition gets wasted and remain unused. This is wastage of the memory and called internal fragmentation.
            2. **External Fragmentation:** The total unused space of various partitions cannot be used to load the processes even though there is space available but not in the contiguous form.
            3. Limitation on process size: If the process size is larger than the size of maximum sized partition then that process cannot be loaded into the memory. Therefore, a limitation can be imposed on the process size that is it cannot be larger than the size of the largest partition.

4. Low degree of multi-programming: In fixed partitioning, the degree of multiprogramming is fixed and very less because the size of the partition cannot be varied according to the size of processes.

c. Dynamic Partitioning

i. In this technique, the partition size is not declared initially. It is declared at the time of process loading.



Dynamic Partitioning

(Process Size = Partition Size)

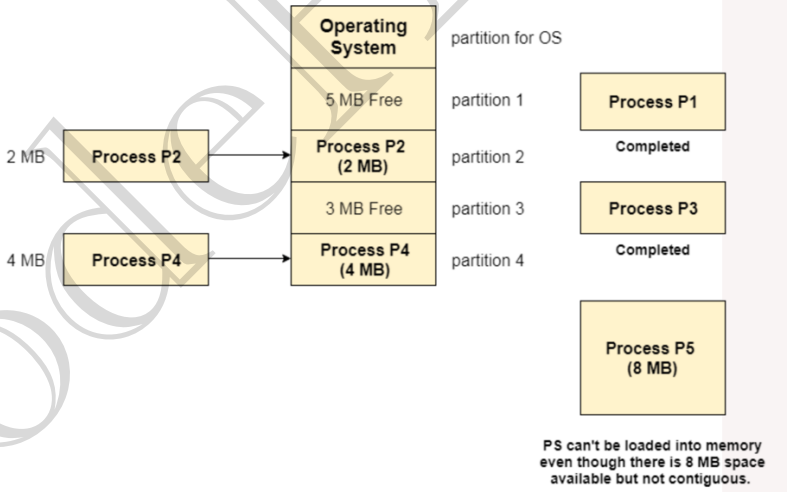
ii.

iii. Advantages over fixed partitioning

1. No internal fragmentation
2. No limit on size of process
3. Better degree of multi-programming

iv. Limitation

1. External fragmentation



External Fragmentation in Dynamic Partitioning

X