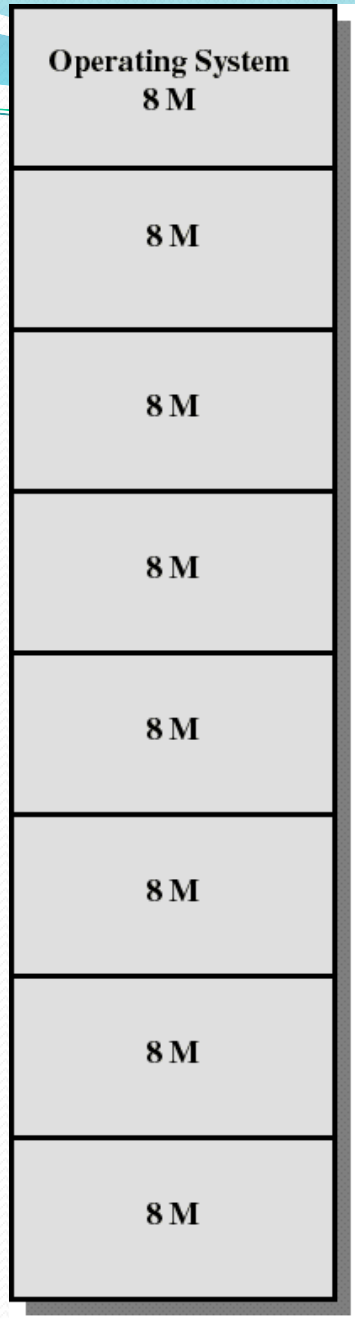
Fixed Partitioned Memory Management

Fixed Partition Multi Programming

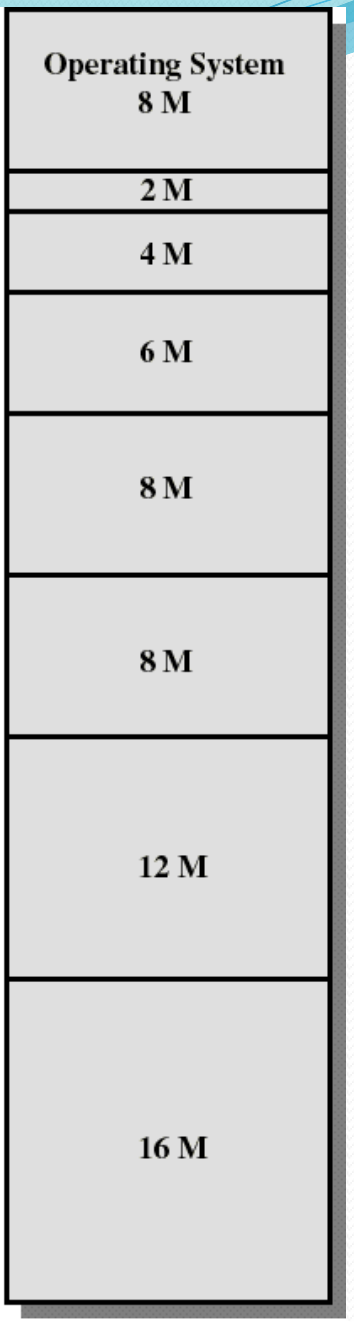
- Divide memory into a number of fixed sized partitions.
- Each partition may contain exactly one process.
- Degree of multiprogramming is bound by number of partitions.
- When a partition is free, a process is selected from the input queue and is loaded into the free partition.
- When a process terminates, the partition becomes available for another process.

Fixed Partitioning

- Divide main memory into a set of non overlapping regions called partitions.
- Partitions can be of equal or unequal sizes.

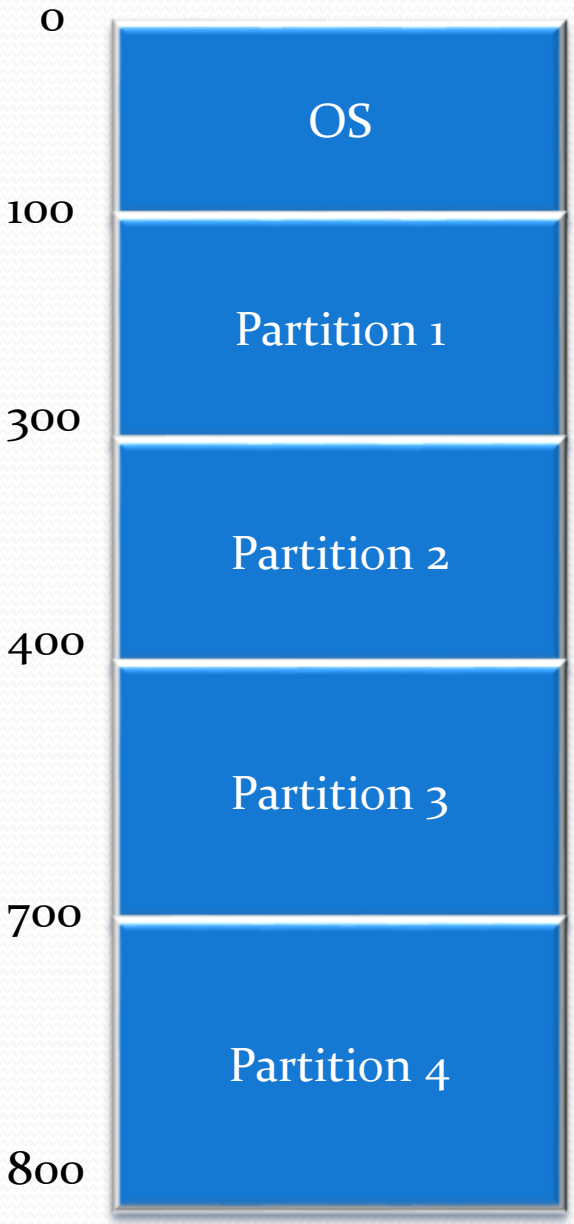


Equal-size partitions



Unequal-size partitions

Partition Description Table (PDT)



Partition ID	Starting Address	Size	Status
0	0	100K	ALLC
1	100K	200K	FREE
2	300K	100K	ALLC
3	400K	300K	ALLC
4	700K	100K	FREE
5	800K	200K	ALLC

Placement Strategies

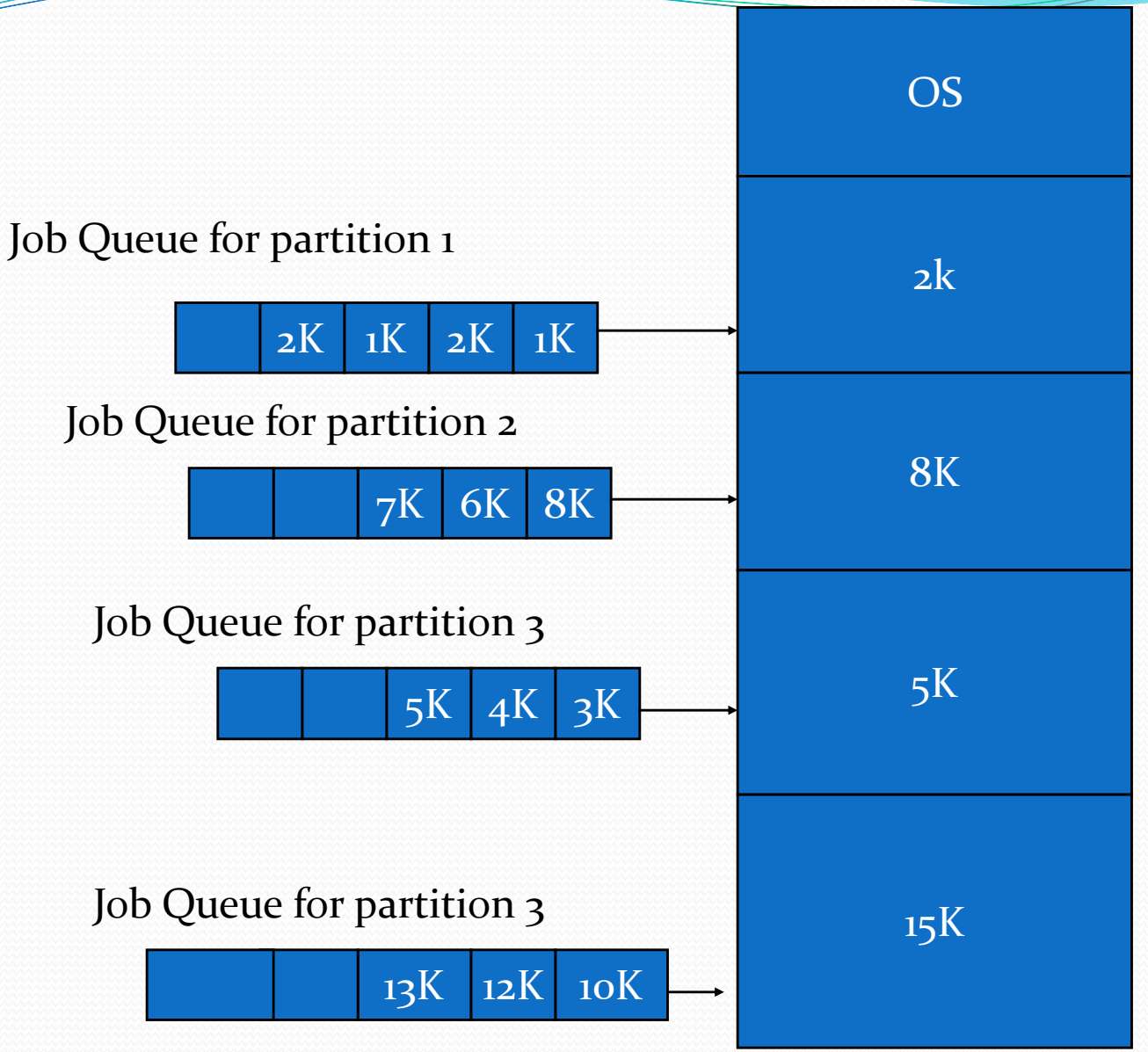
- These are concerned with determining where in main storage to place an incoming program.
  - First Fit: An incoming job is placed in the main storage in the first available hole large enough to hold it.
  - Best Fit : An incoming job is placed in the main memory in which it fits most tightly and leaves the smallest amount of unused space.
  - Worst Fit: It places a program in main memory in the hole in which it fits worst, ie., The largest possible hole.

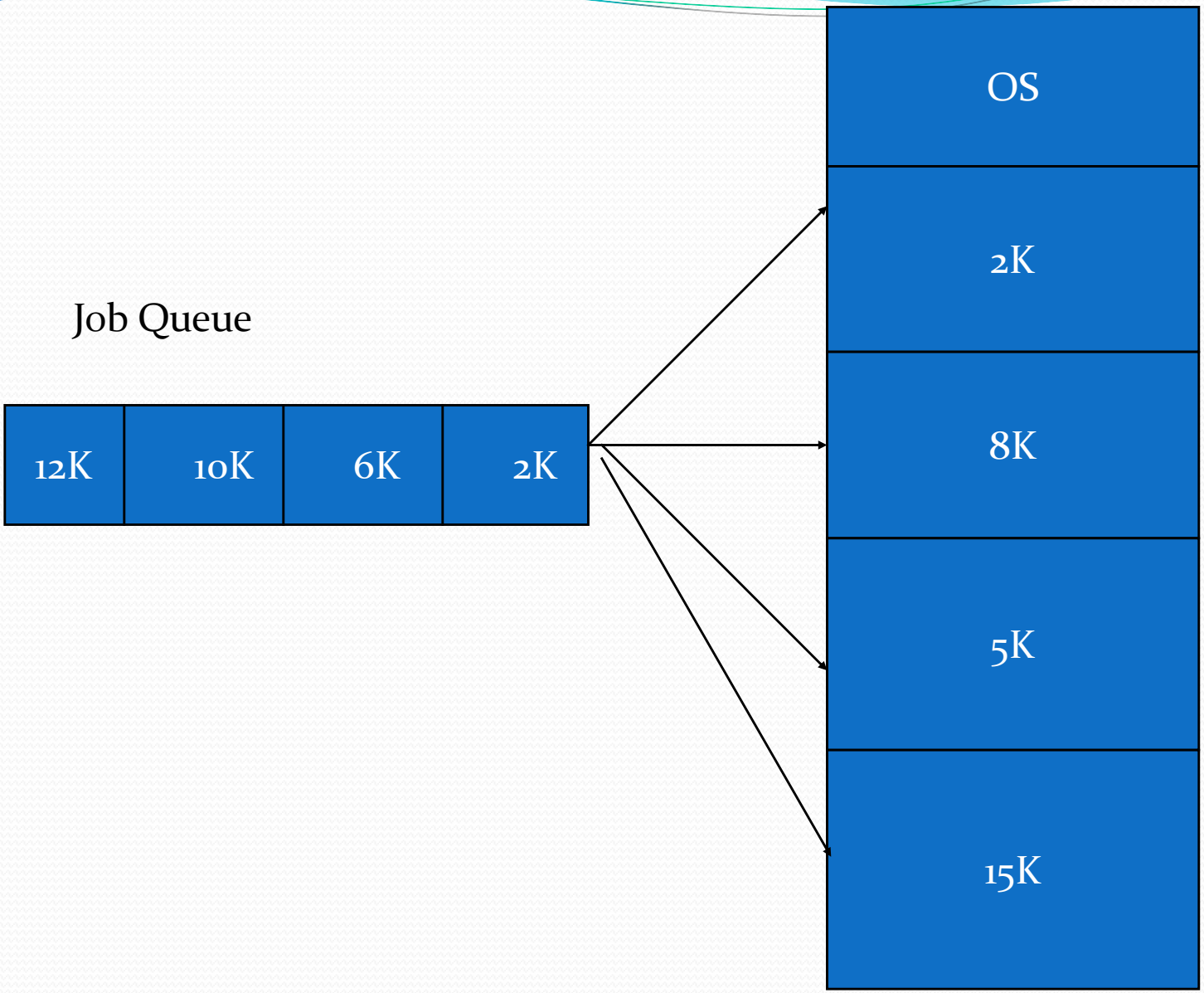
Fixed Partition Multi Programming

- When a partition is to be allocated to a process, the following takes place
  - Long term scheduler decides which process is to be brought into the memory
  - It then find out the size of the program with the help of IM
  - Generates a request to partition allocation routine of MM to allocate a free partition.
  - The routine selects a partition based on the placement strategy.

Fixed Partition Multi Programming

- With the help of IM it then loads the program in selected partition.
- It then update the PCB with the corresponding Partition ID.
- Links the PCB to ready list.
- MM then updates the PDT and make the status of partition as ALLC.
- PM then schedules the process.

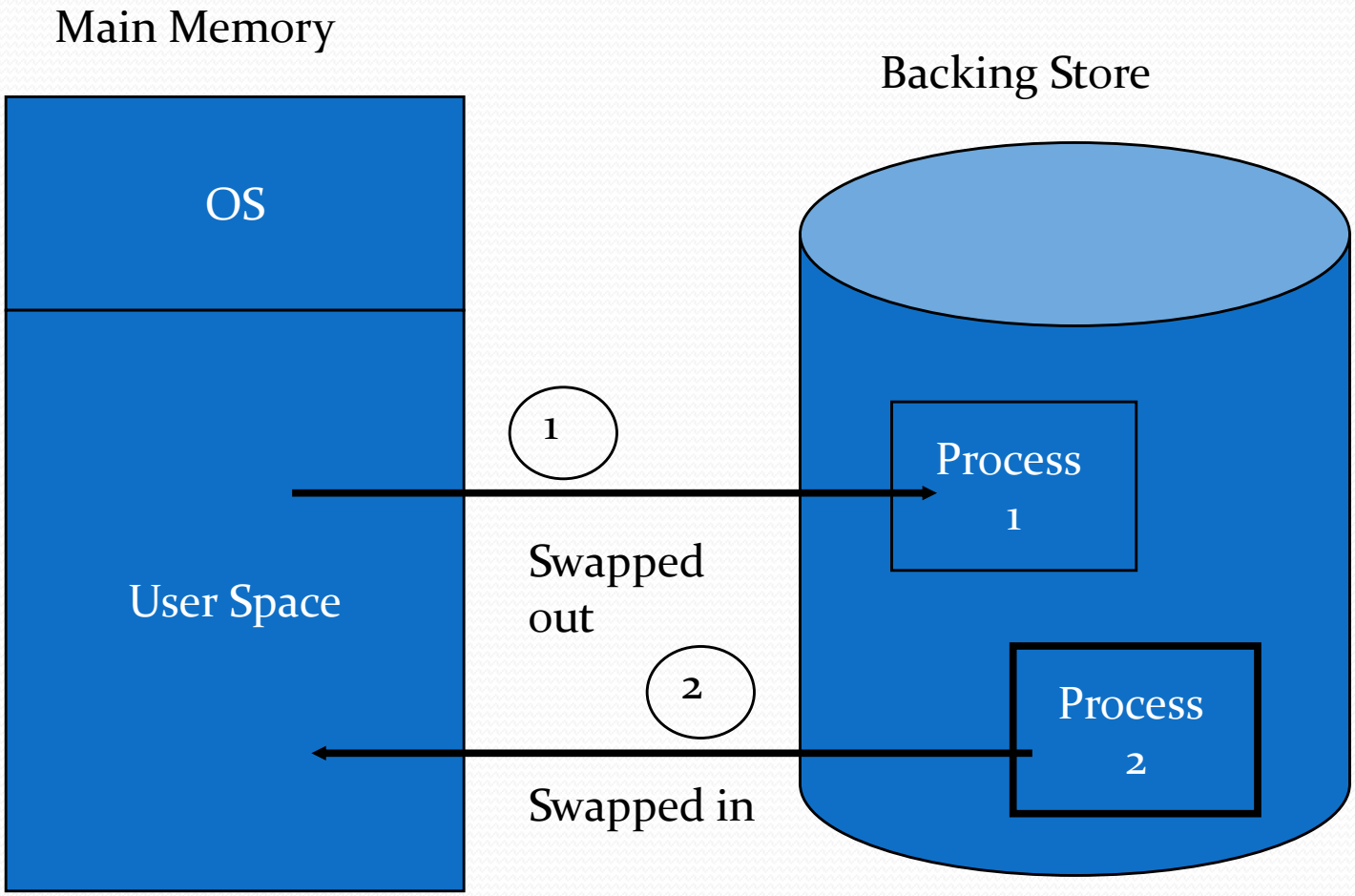

OS
2k
8K
5K
15K
Job Queue for partition 1
2K
1K
2K
1K
Job Queue for partition 2
7K
6K
8K
Job Queue for partition 3
5K
4K
3K
Job Queue for partition 3
13K
12K
10K


Job Queue
12K
10K
6K
2K
OS
2K
8K
5K
15K

Fragmentation

- Storage fragmentation occurs in every computer system regardless of a storage organization.
  - **Internal fragmentation**
    - Memory that is internal to the partition, but is not being used.
  - **External fragmentation**
    - A partition remains unused if it is too small to hold the waiting job.

Swapping


Main Memory
Backing Store
OS
User Space
1
Swapped out
2
Swapped in
Process 1
Process 2

Swapping

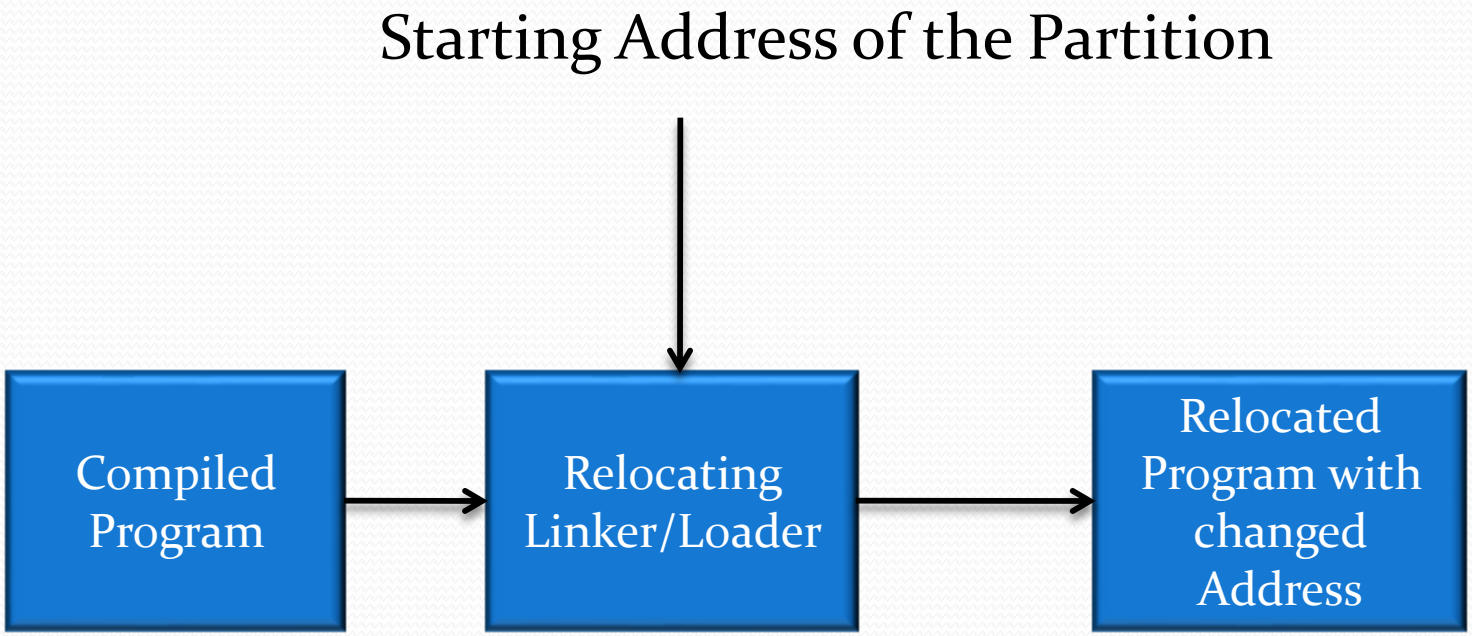
- A process needs to be in memory to be executed. A process can be swapped temporarily out of memory to backing store and then brought back into main memory for continued execution.
- Normally a process that is swapped out will be swapped back into the same memory space that it occupied previously.

Relocation and Address Translation

- Logical Address → Physical Address
- There two ways to achieve relocation
  - Static Relocation and address translation
  - Dynamic Relocation and address translation

Static Relocation and AT

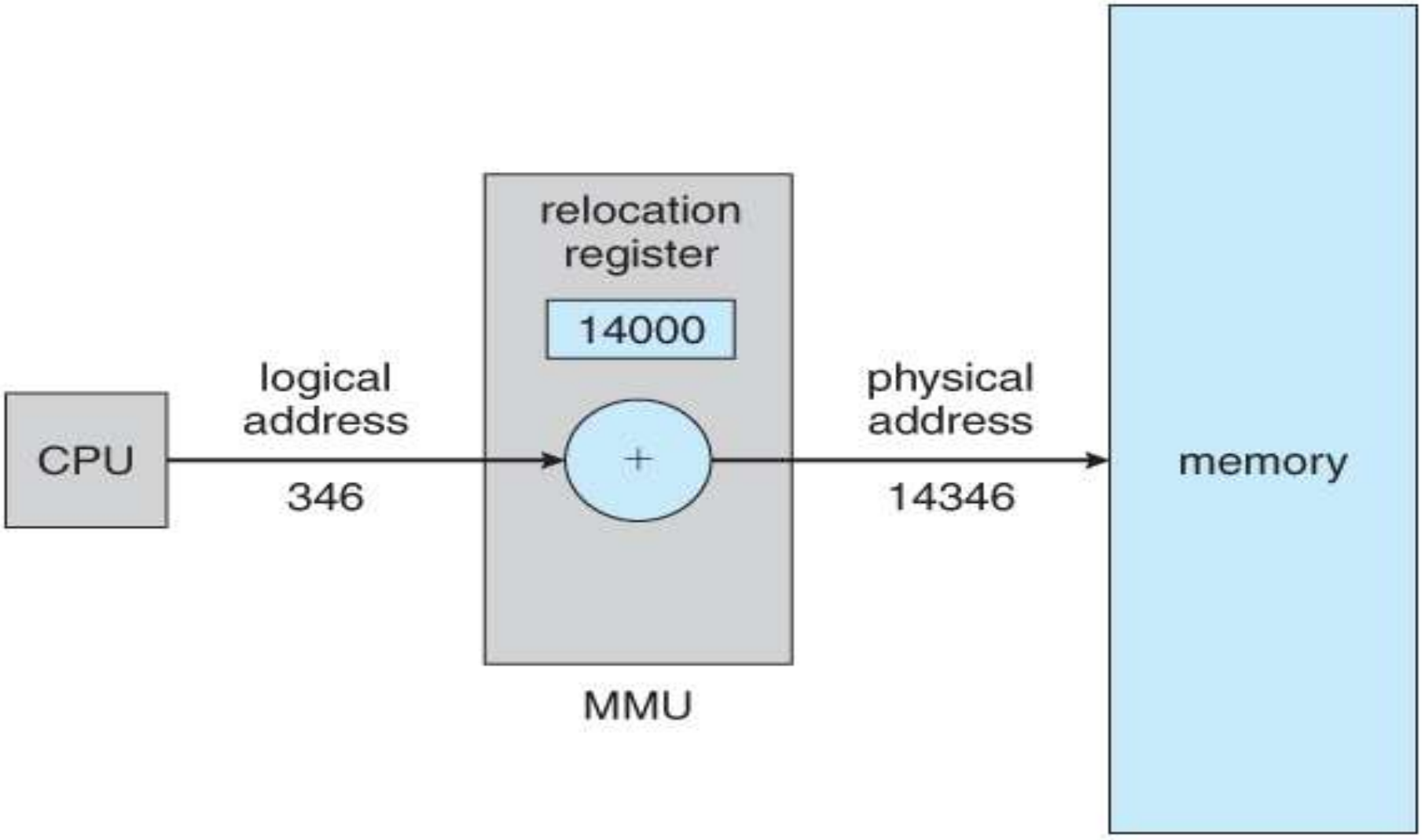
- Relocating Linker or Relocating Loader performs the relocation.



Static Relocation and AT

- The relocating Linker/Loader will have to know which portion of the instruction is an address
- Difficulty in find out the length of the instruction.
- It is a slow process because it is software translation.
- It performs translation only once. No support for swapping.

Dynamic Relocation and AT



Dynamic Relocation and AT

- Commonly used scheme in fixed partition.
- Fast and flexible.
- Supports swapping.

Protection and sharing

- A process should not, by mistake or on purpose, become capable of interfering with other processes.
- Two approaches
  - Protection Bits.
  - Limit Registers.

Protection Bits

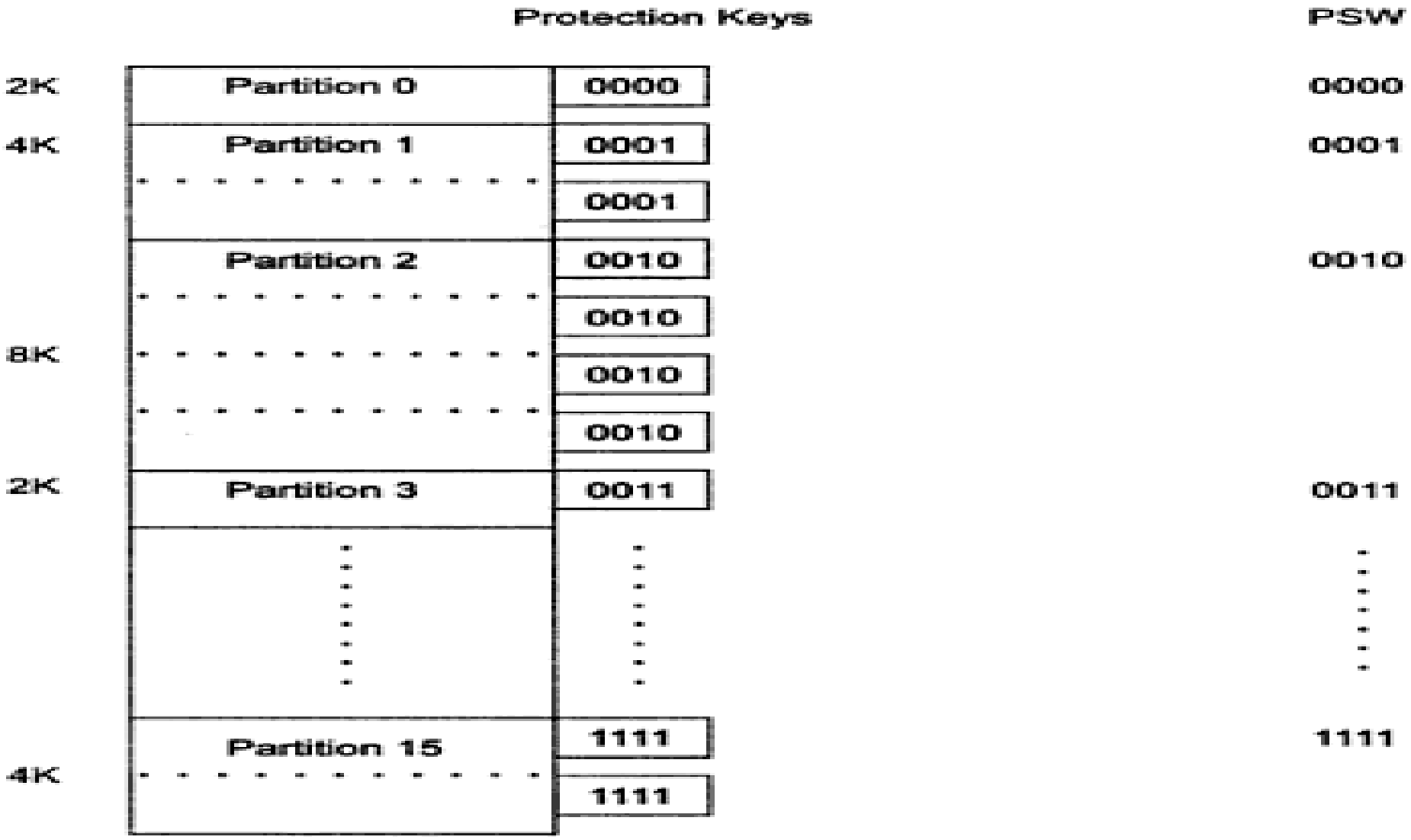
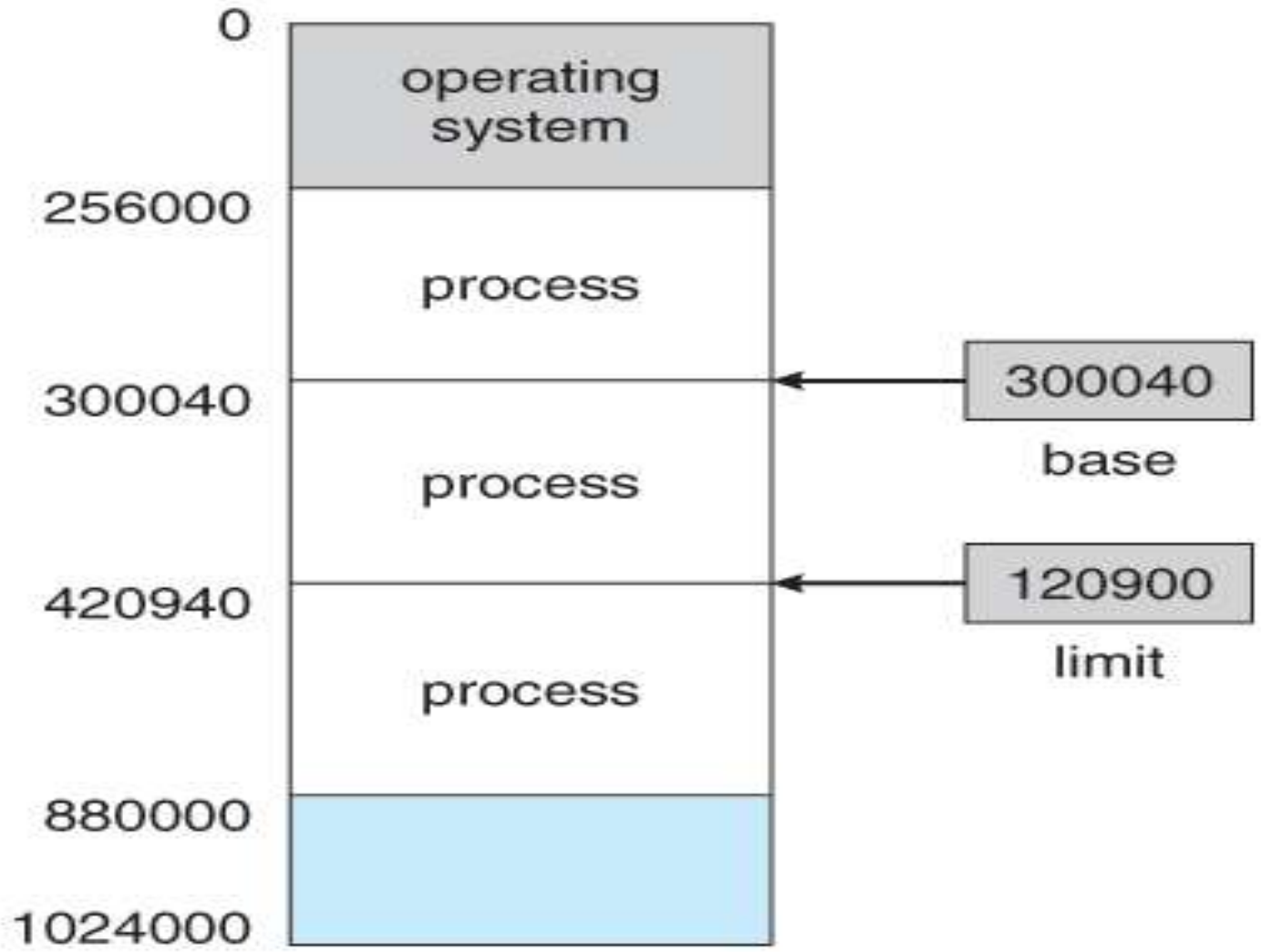
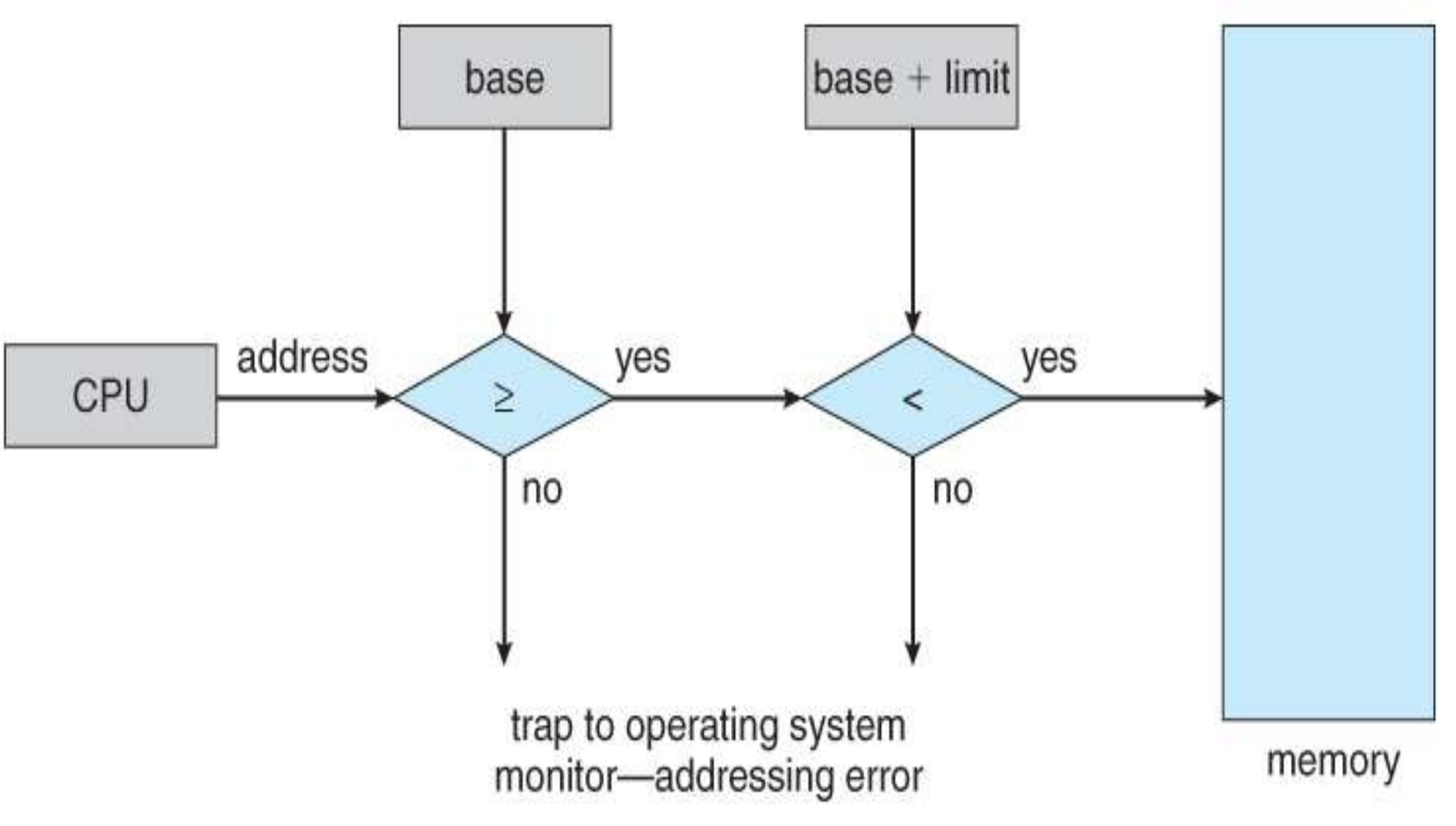


Fig. 8.10 Protection keys

Base and Limit Register



Base and Limit Register



Evaluation

- Wasted Memory
  - Both Internal and external fragmentation can occur.
- Access Time
  - Access times are not very high due to assistance of special hardware.
- Time Complexity
  - Very Low because allocation/de-allocation routines are simple, as the partitions are fixed.