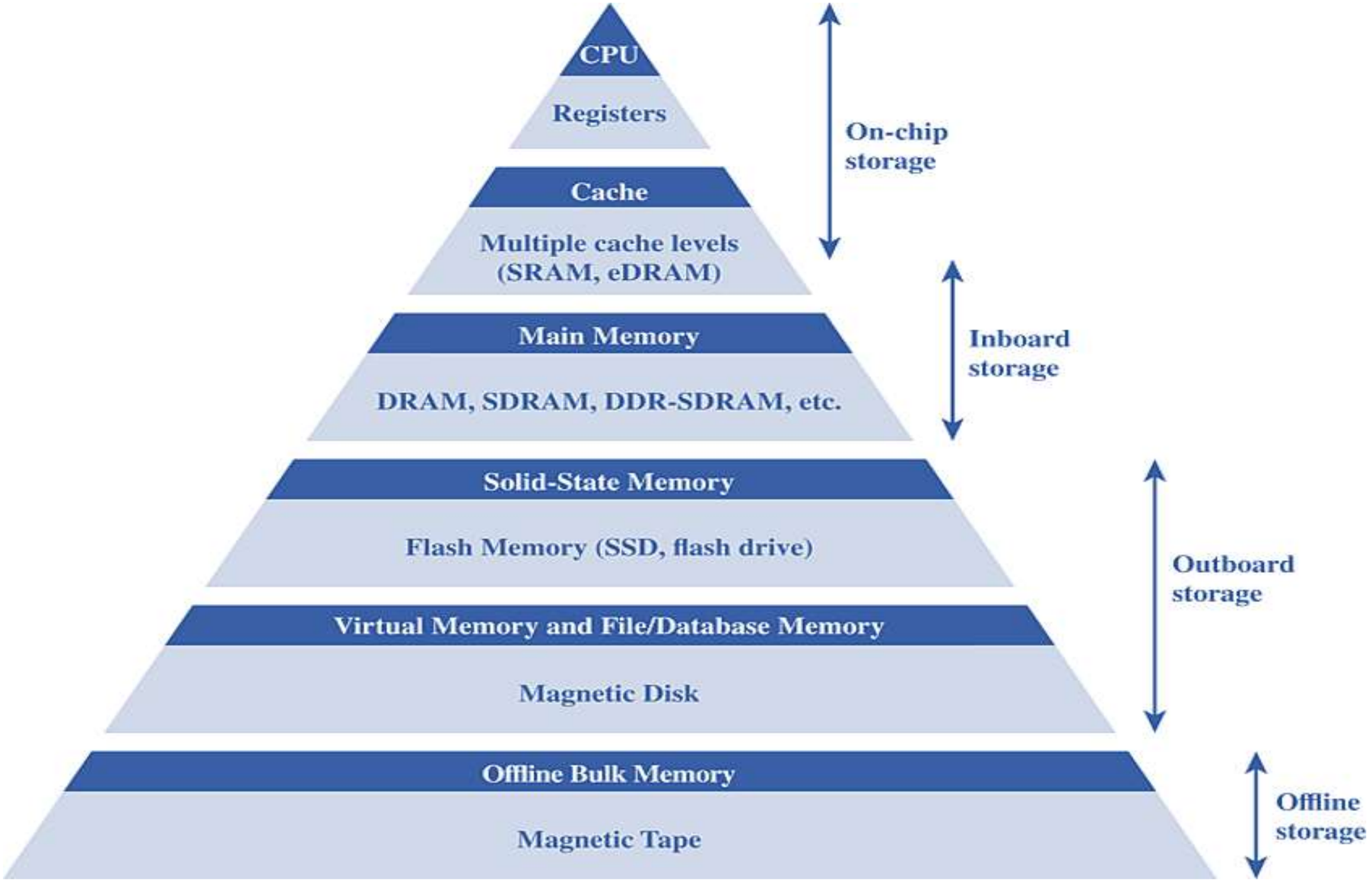
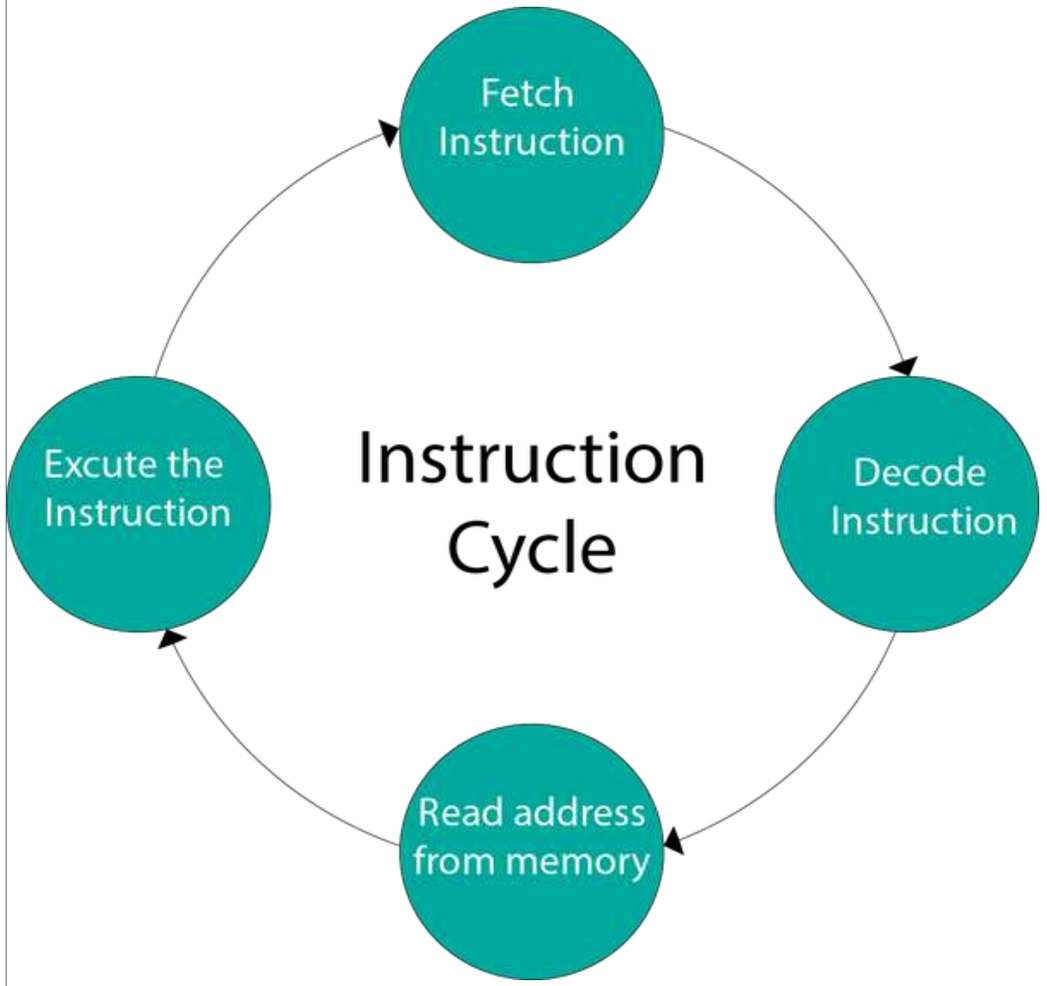
Memory Management

Memory Hierarchy



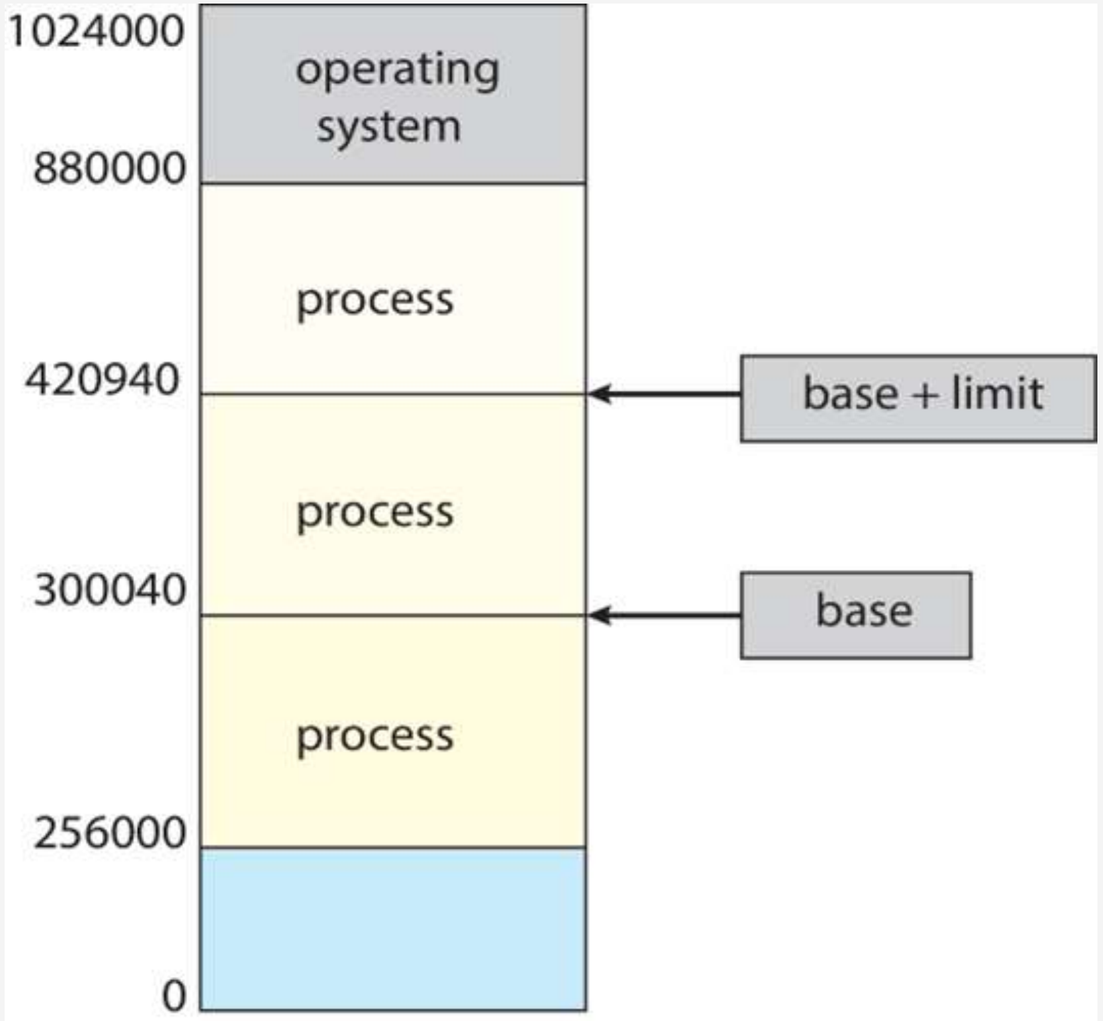
Background

- ✓ Large array of words or bytes, each with its own address.
- ✓ The memory unit sees only a stream of memory addresses
- ✓ Memory unit only sees a stream of:
  - ✓ addresses + read requests, or
  - ✓ address + data and write request



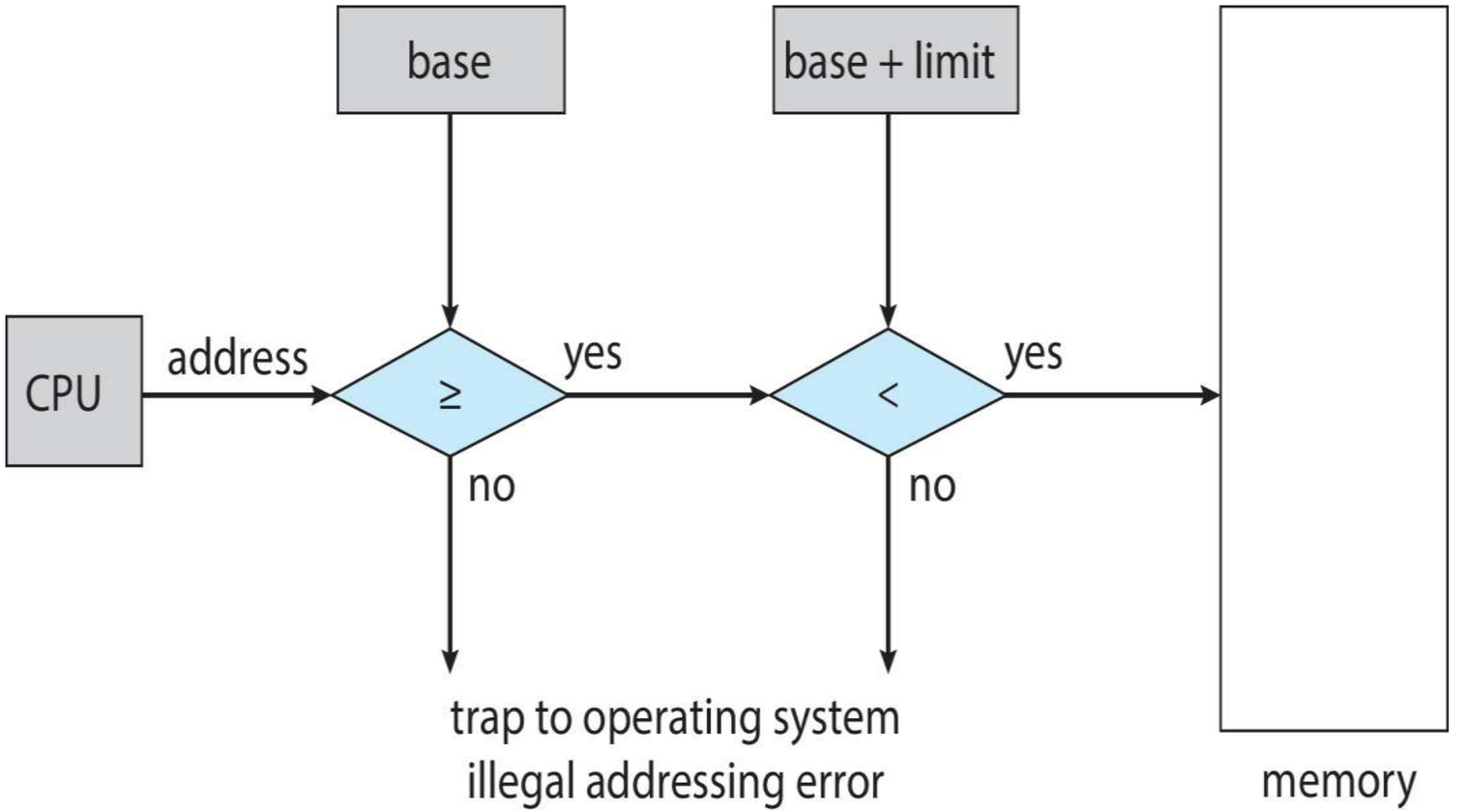
Protection

- Need to ensure that a process can access only those addresses in its address space.
- We can provide this protection by using a pair of **base** and **limit registers** define the logical address space of a process



Hardware Address Protection

- CPU must check every memory access generated in user mode to be sure it is between base and limit for that user
- We can provide this protection by using a pair of **base** and **limit registers** define the logical address space of a process



The instructions to loading the base and limit registers are privileged

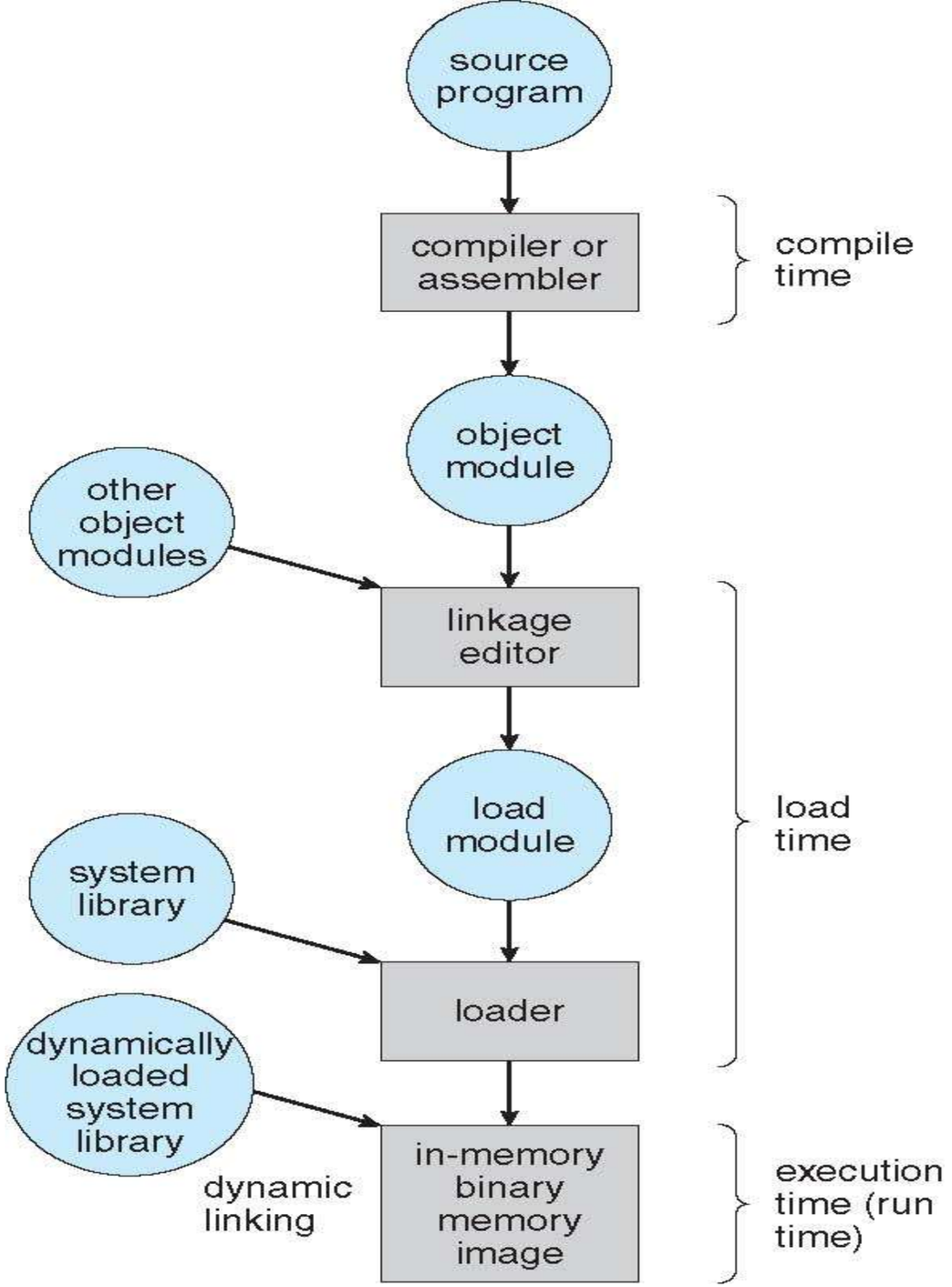
Address Binding

- A program resides on a disk as a binary executable file.
- The program must be brought into memory and placed within a process for it to be executed.
- Depending on the memory management in use, the process may be moved between disk and memory during its execution.
- As the process is executed, it accesses instructions and data from memory. Eventually, the process terminates, and its memory space is declared available.

Binding of Instructions and Data to Memory

- ✓ Address binding is the process of mapping from one address space to another address space.
- ✓ Three Stages
  - ➢ **Compile time**: If memory location known a priori, **absolute code** can be generated; must recompile code if starting location changes
  - ➢ **Load time**: Must generate **relocatable code** if memory location is not known at compile time
  - ➢ Loader will converts this relocatable address to physical address
  - ➢ **Execution time**: Binding delayed until run time if the process can be moved during its execution from one memory segment to another.

Multistep Processing of a User Program



Logical Vs Physical Address Space

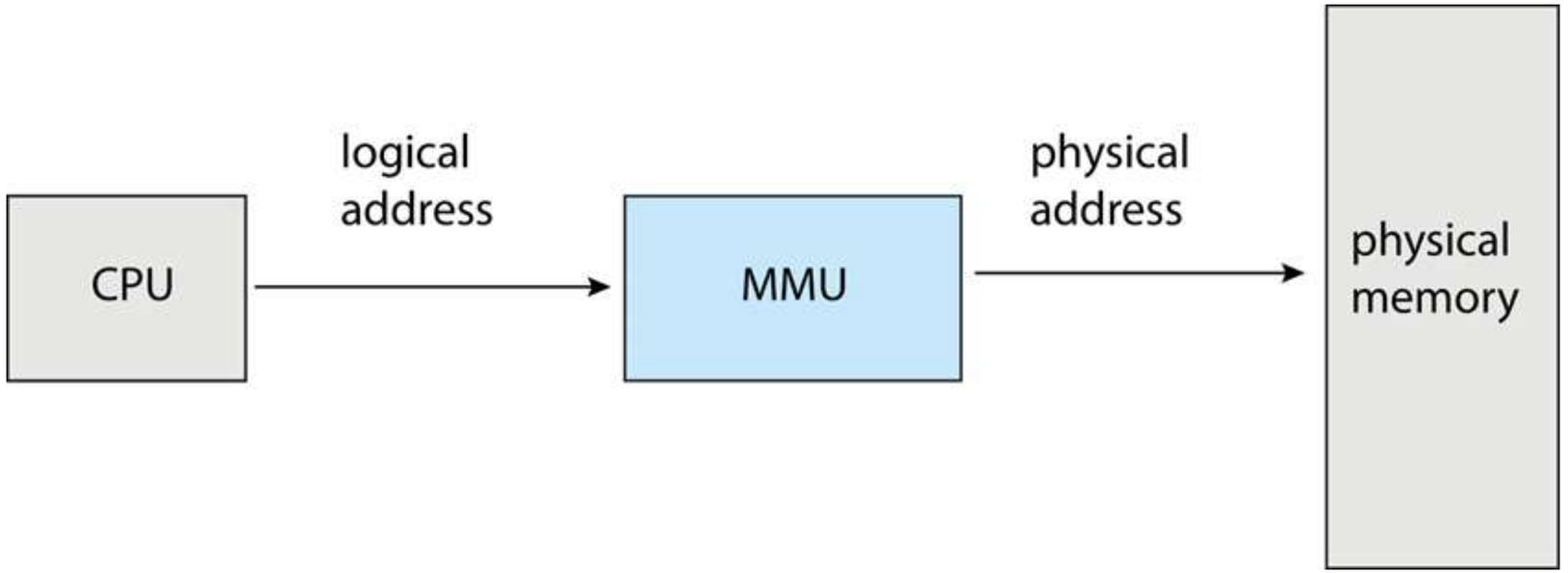
- ✓ **Logical address** – generated by the CPU; also referred to as **virtual address**
- ✓ **Physical address** – address seen by the memory unit
- ✓ Logical and physical addresses are the same in compile-time and load-time address-binding schemes.
- ✓ logical (virtual) and physical addresses differ in execution-time address-binding scheme.

Logical Vs Physical Address Space

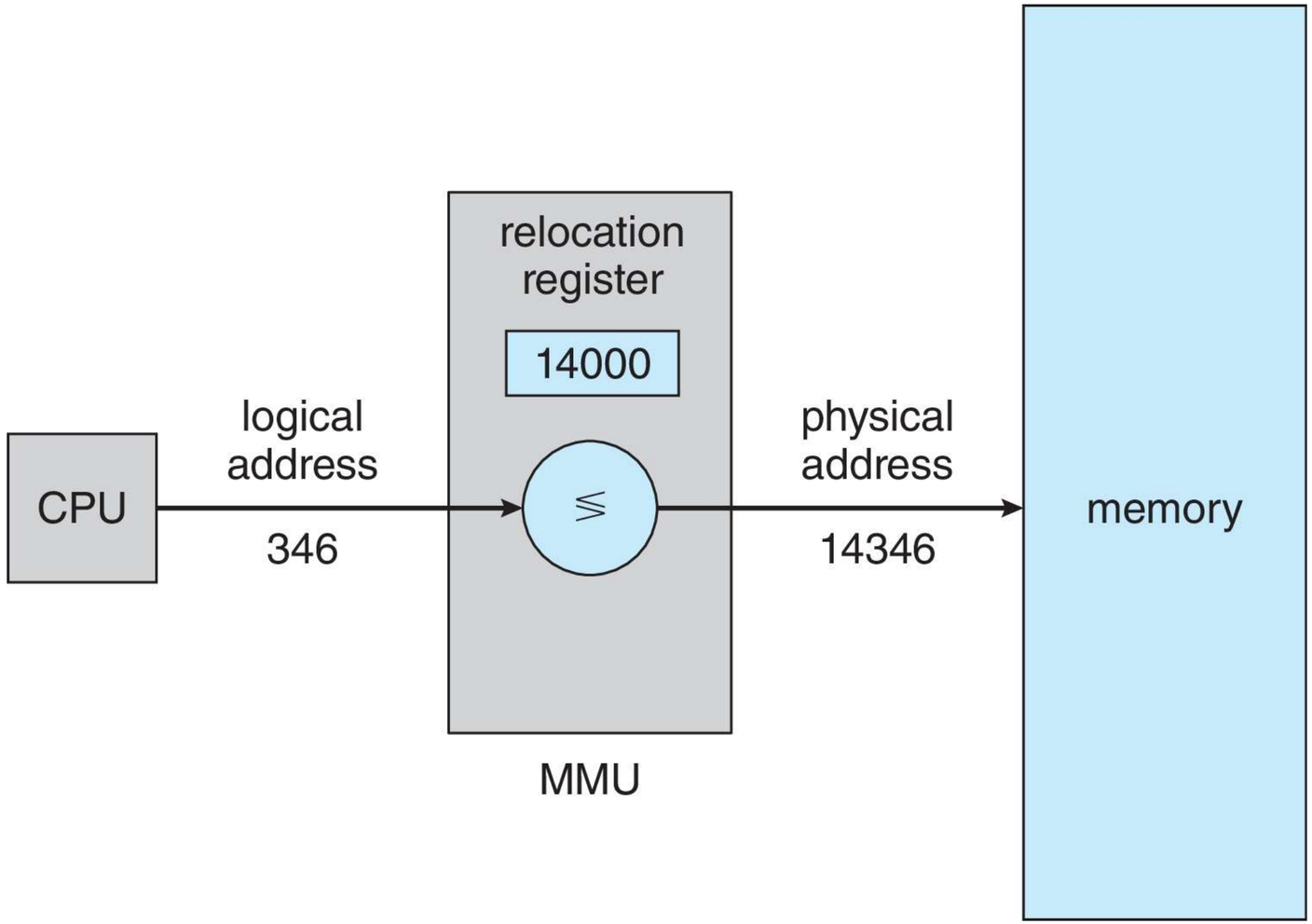
- ✓ **Logical address space** is the set of all logical addresses generated by a program
- ✓ **Physical address space** is the set of all physical addresses generated by a program

Memory Management Unit

- Hardware device that at run time maps virtual to physical address

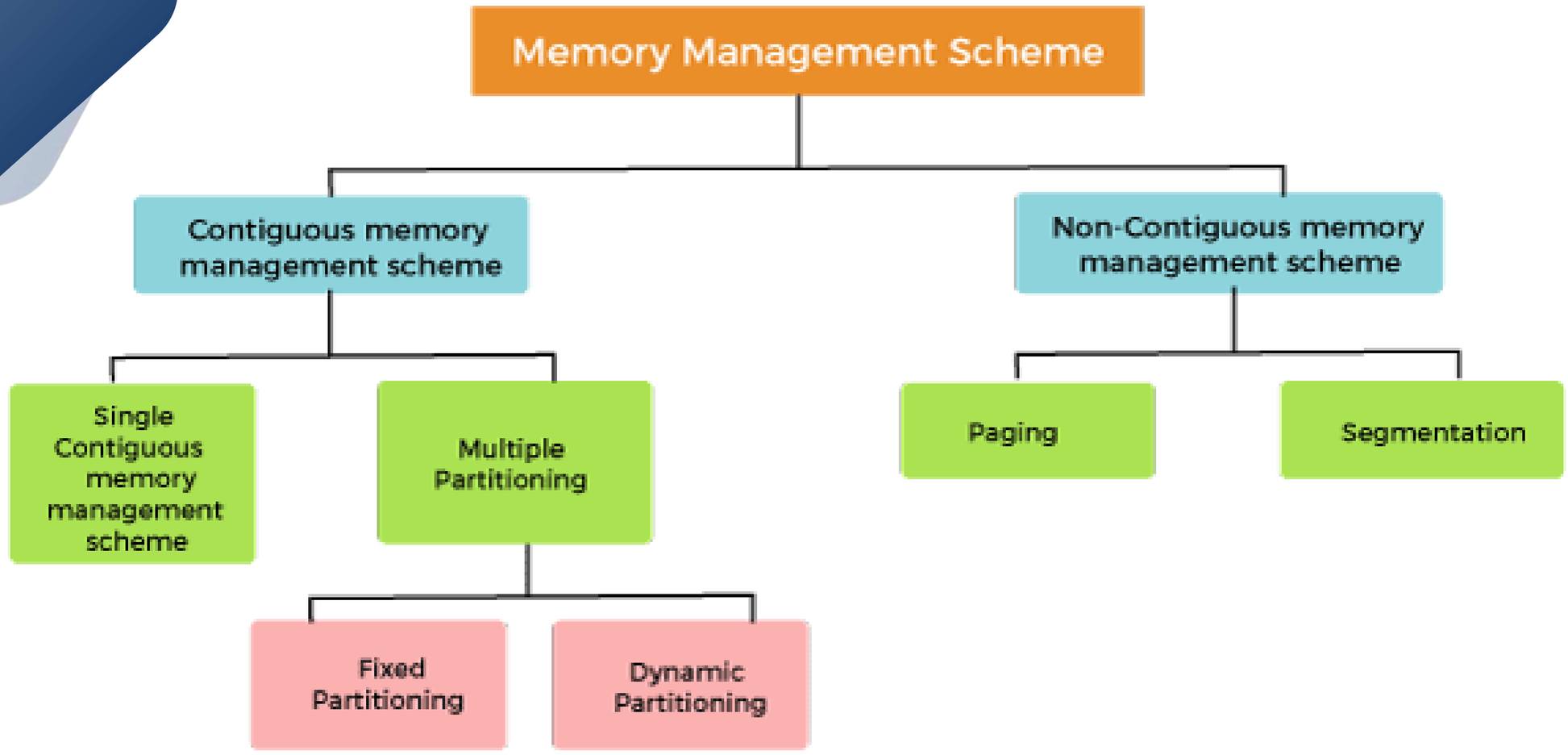


Memory Management Unit



- The base register now called ***relocation register***
- The value in the relocation register is added to every address generated by a user process at the time it is sent to memory

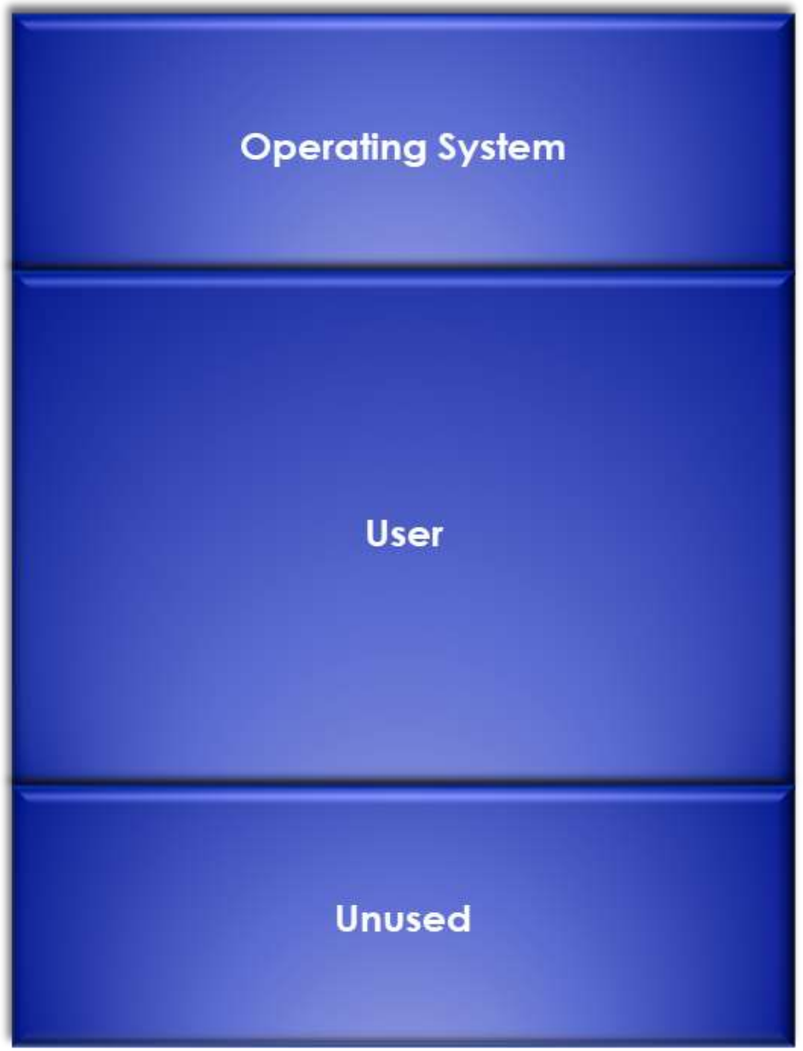
Memory Management



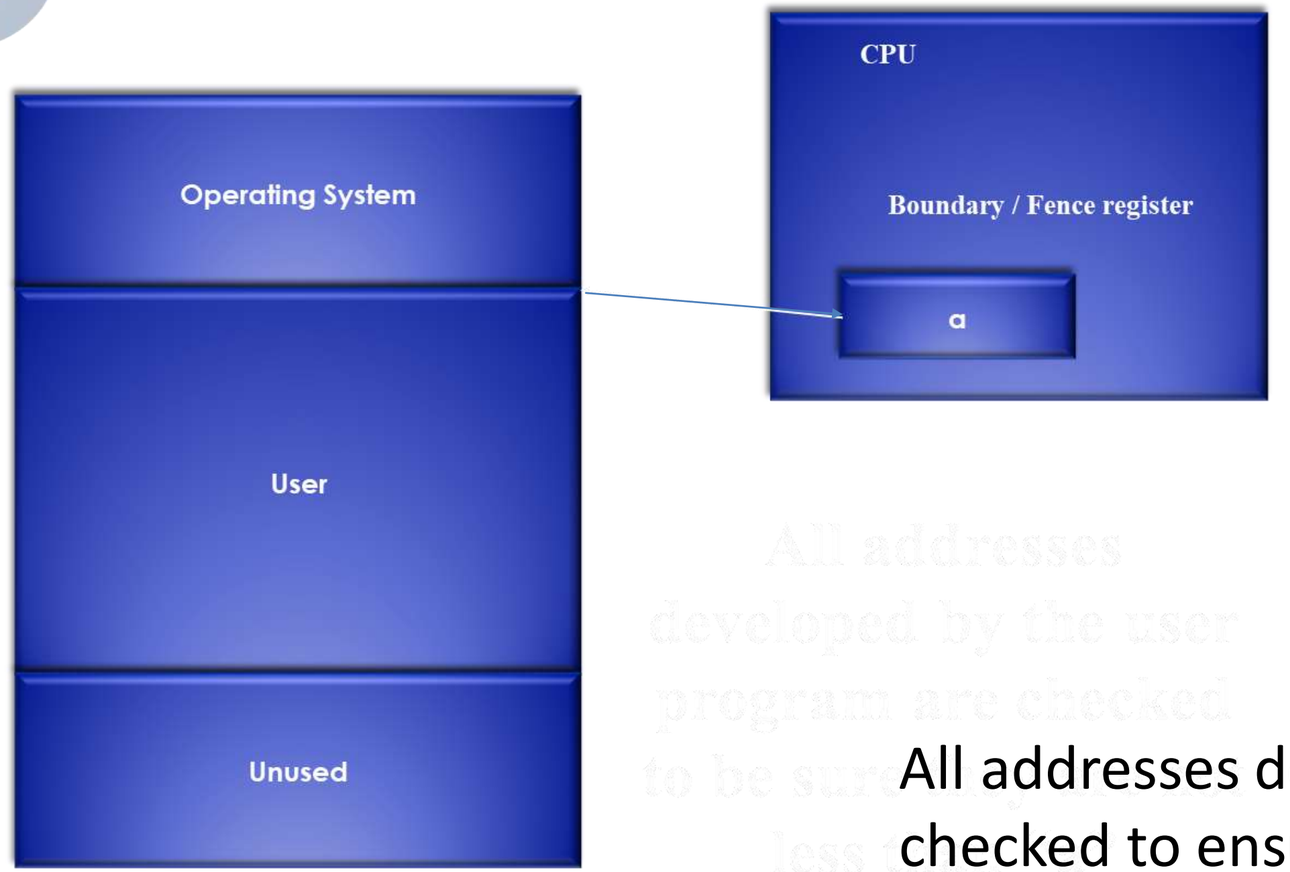
Classification of memory management schemes

Contiguous Allocation

- ✓ **Single User Continuous Allocation**



Storage Protection with single user contiguous system



All addresses developed by the user program are checked to ensure they are not less than “a”