



# **SNS COLLEGE OF TECHNOLOGY**

**Coimbatore-35.**

**An Autonomous Institution**

**Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A+’ Grade  
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai**



**COURSE NAME : 19CST201 – OPERATING SYSTEMS**

**II YEAR/ IV SEMESTER**

**UNIT – III STORAGE MANAGEMENT**

**Topic: Memory Management**

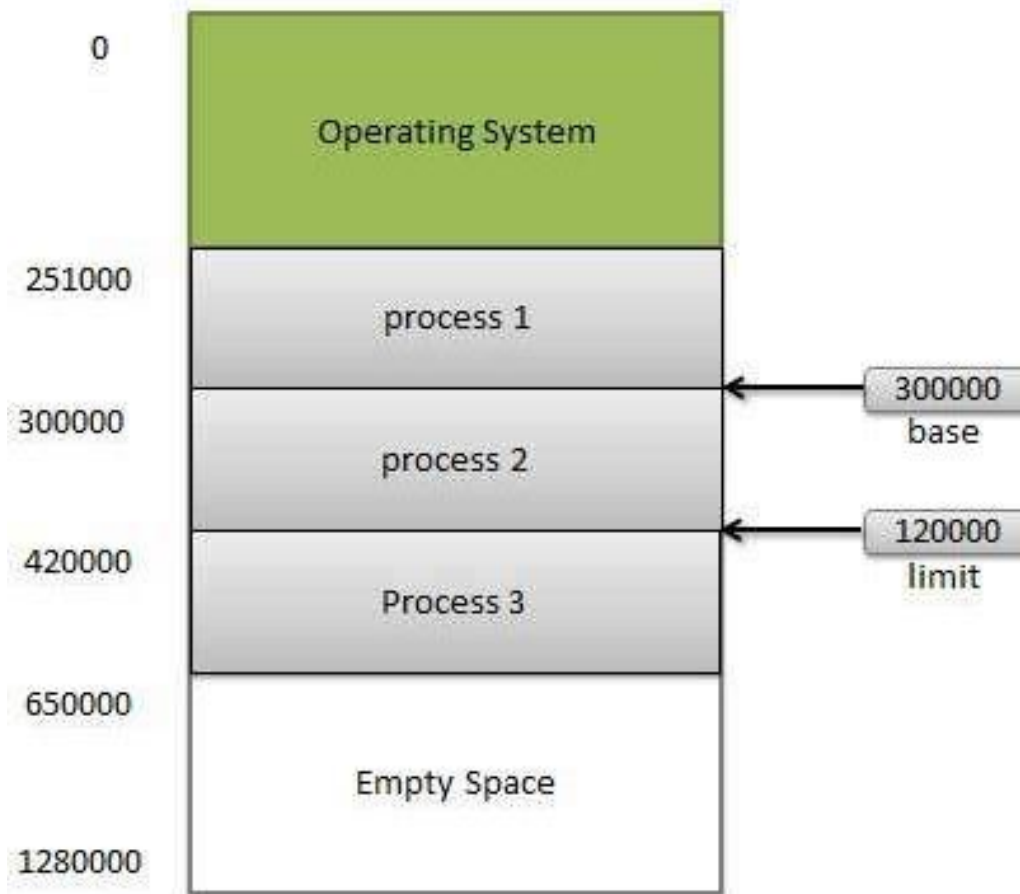
Dr.A.Sumithra

Associate Professor

Department of Computer Science and Engineering

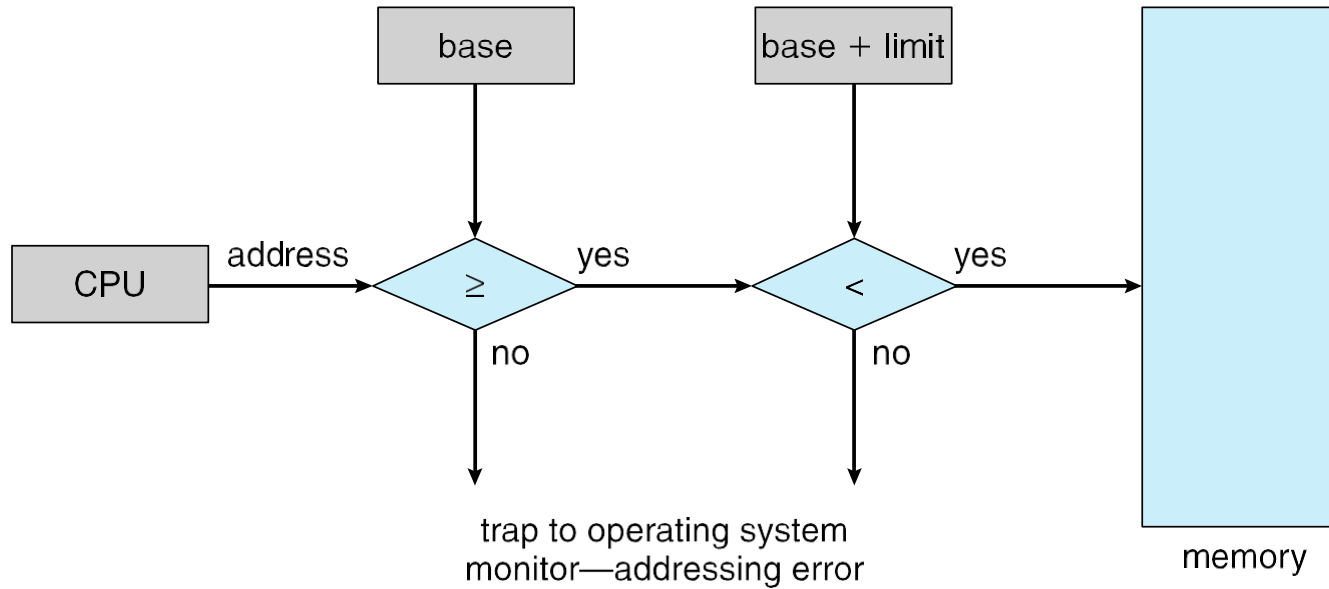


# Background -Base and Limit Registers





# Hardware Address Protection



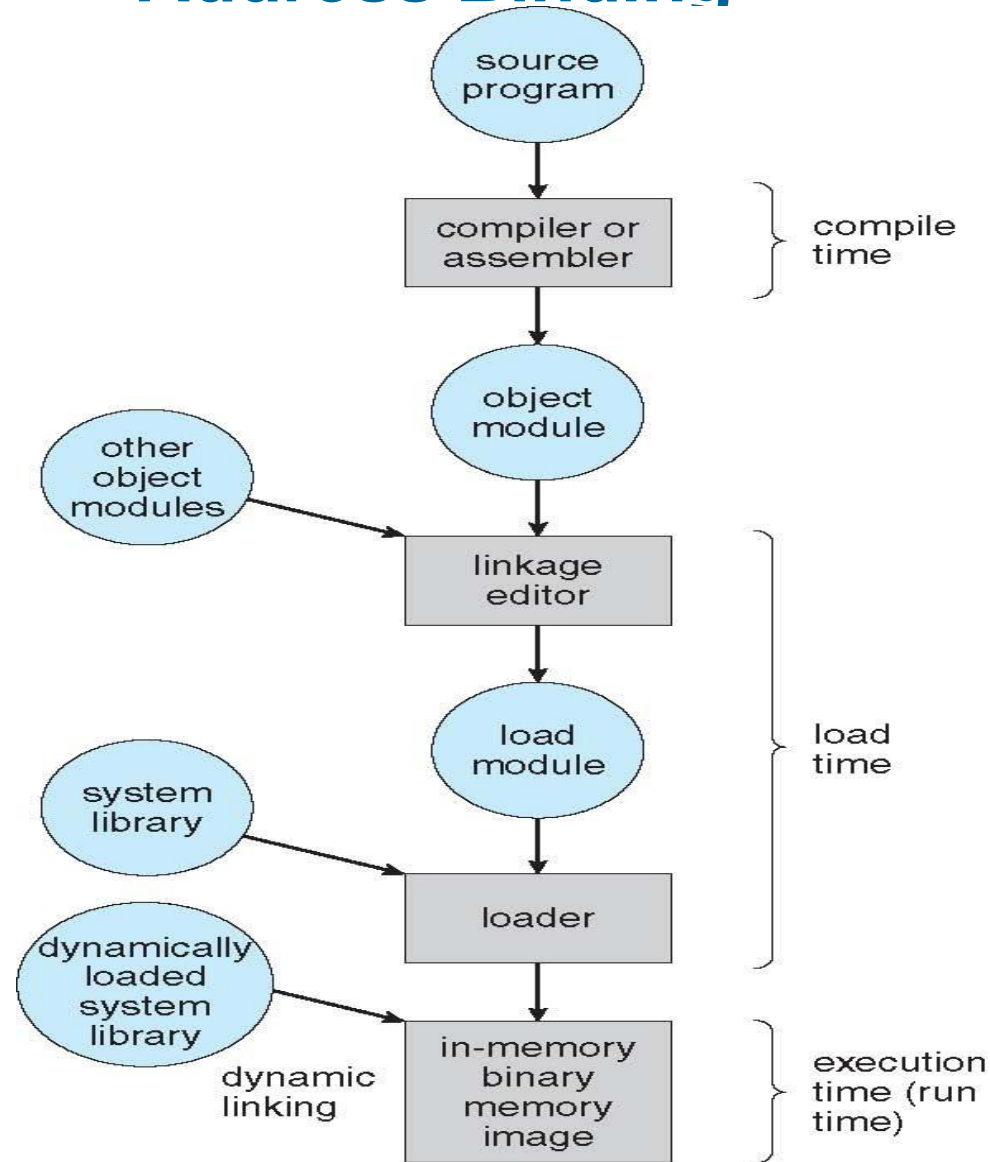


# Address Binding

- **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
- **Execution time:** Binding delayed until run time if the process can be moved during its execution from one memory segment to another
  - ▶ Need hardware support for address maps (e.g., base and limit registers)



# Address Binding





# 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 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



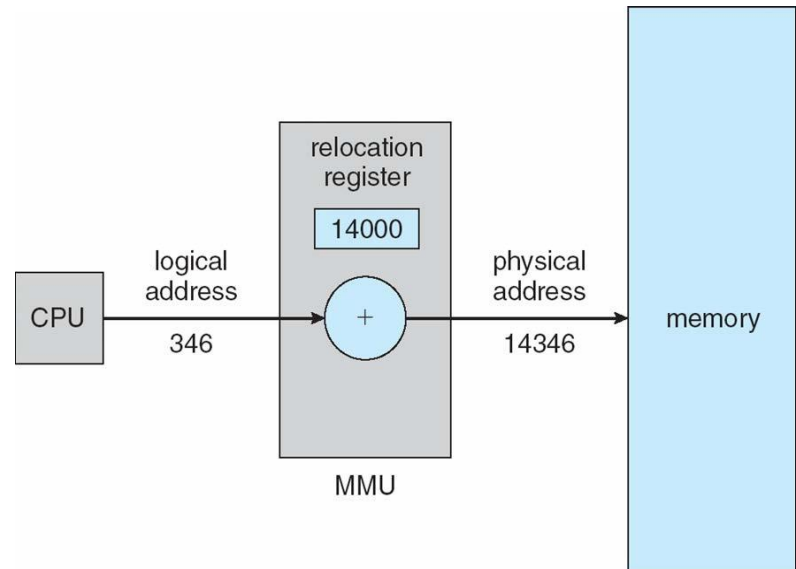
# Memory-Management Unit (MMU)

- To start, consider simple scheme where the value in the relocation register is added to every address generated by a user process at the time it is sent to memory
  - Base register now called **relocation register**



# Dynamic relocation using a relocation register

- Routine is not loaded until it is called
- Better memory-space utilization; unused routine is never loaded
- All routines kept on disk in relocatable load format
- Useful when large amounts of code are needed to handle infrequently occurring cases
- No special support from the operating system is required
- Implemented through program design
- OS can help by providing libraries to implement dynamic loading







# Dynamic Linking & Loading

- Dynamic linking –linking postponed until execution time
- Dynamic linking is particularly useful for libraries