Amirkabir University of Technology
(Tehran Polytechnic)

Operating Systems

Main Memory-Part1

Seyyed Ahmad Javadi

sajavadi@aut.ac.ir

Fall 2021

Chapter 9: Memory Management

- Background
- Contiguous Memory Allocation
- Paging
- Structure of the Page Table
- Swapping

Objectives

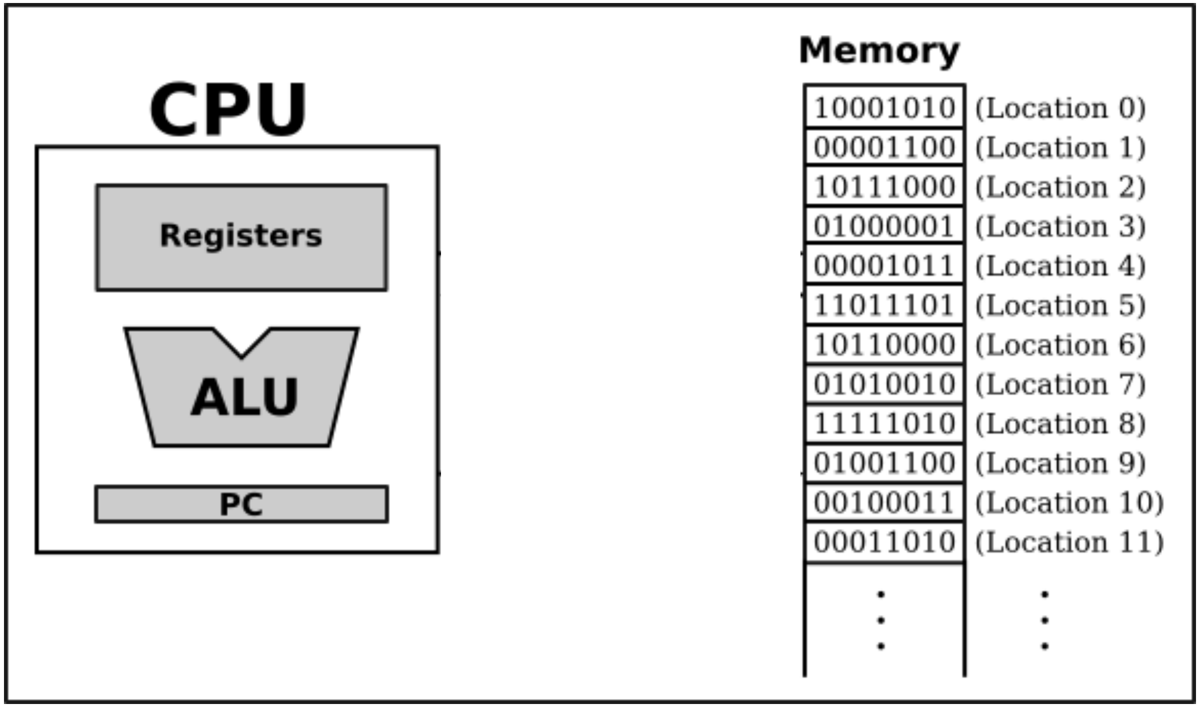
- To provide a detailed description of various ways of organizing memory hardware.

- To discuss various memory-management techniques.

- To provide a detailed description of the Intel Pentium, which supports both pure segmentation and segmentation with paging.

Background

- Program must be brought (from disk) into memory and placed within a process for it to be run.
- Main memory and registers are only storage CPU can access directly.

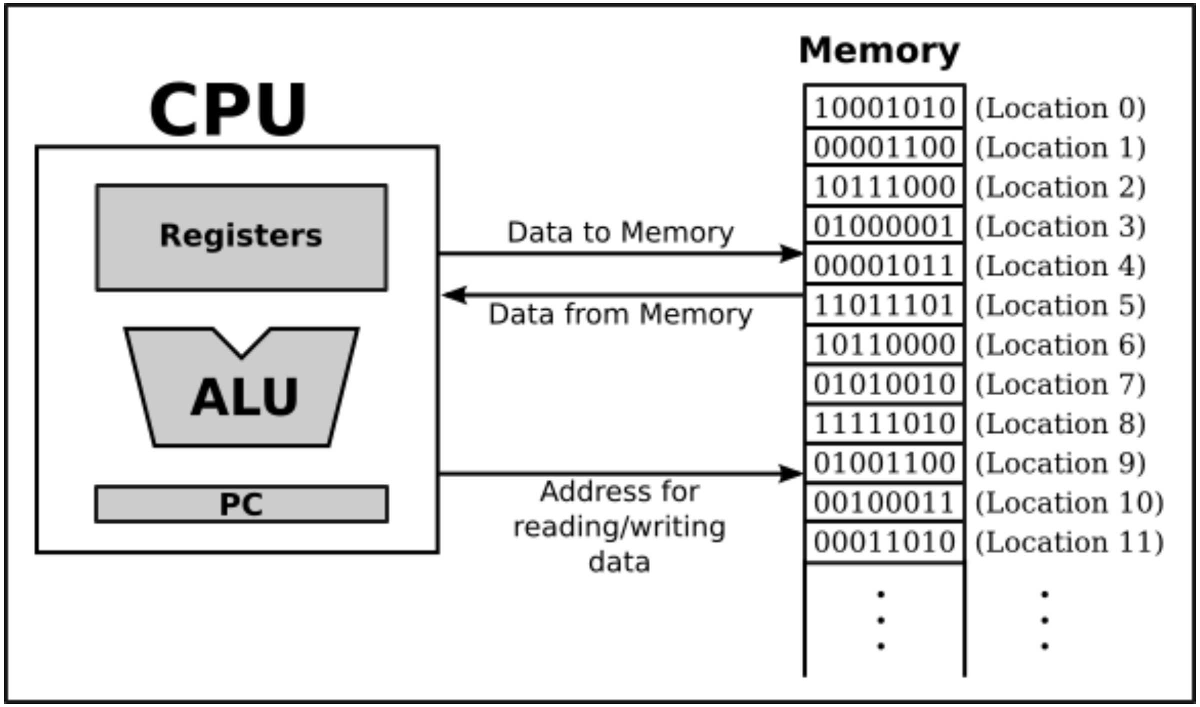


10001010	(Location 0)
00001100	(Location 1)
10111000	(Location 2)
01000001	(Location 3)
00001011	(Location 4)
11011101	(Location 5)
10110000	(Location 6)
01010010	(Location 7)
11111010	(Location 8)
01001100	(Location 9)
00100011	(Location 10)
00011010	(Location 11)
⋮	⋮

https://math.hws.edu/javanotes/c1/s1.html

Background (cont.)

- Memory unit only sees a stream of:
  - addresses + read requests, or
  - address + data and write requests

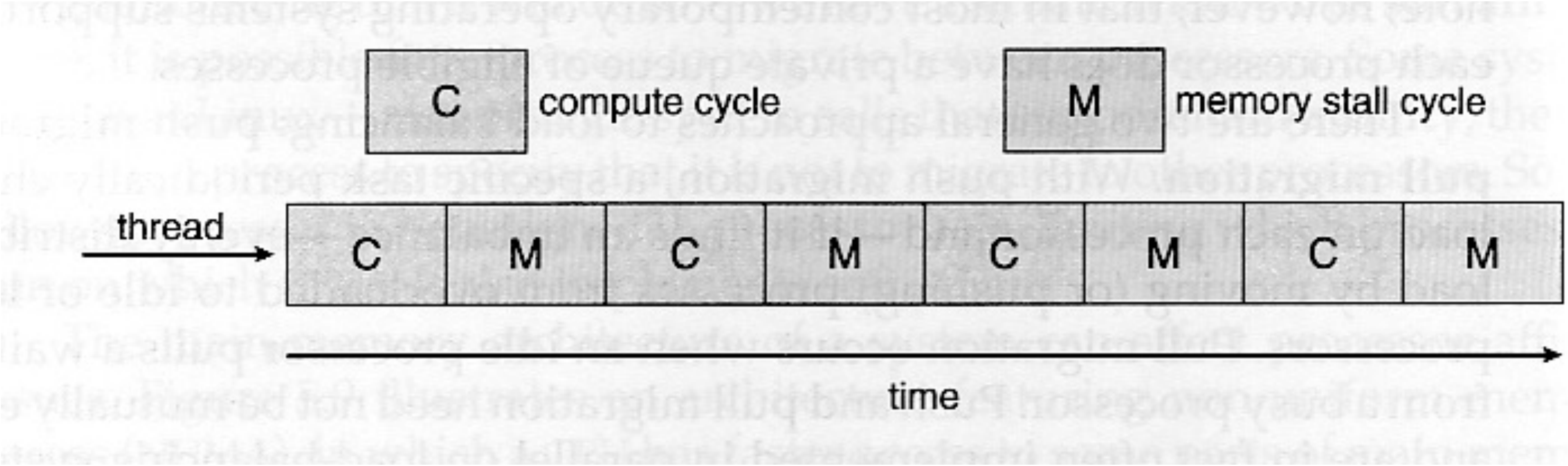


10001010	(Location 0)
00001100	(Location 1)
10111000	(Location 2)
01000001	(Location 3)
00001011	(Location 4)
11011101	(Location 5)
10110000	(Location 6)
01010010	(Location 7)
11111010	(Location 8)
01001100	(Location 9)
00100011	(Location 10)
00011010	(Location 11)
⋮	⋮

https://math.hws.edu/javanotes/c1/s1.html

Background (cont.)

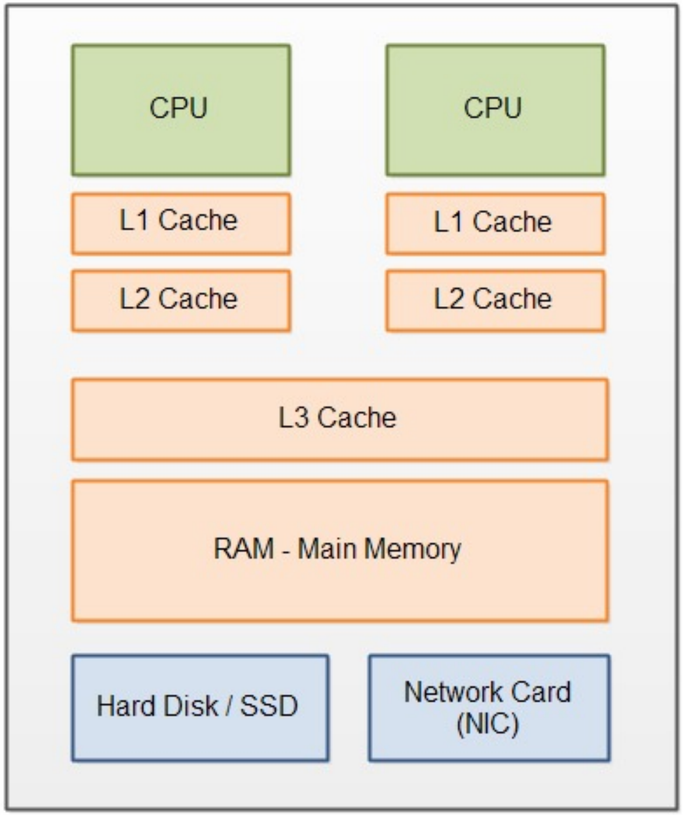
- Register access is done in one CPU clock (or less)
- Main memory can take many cycles, causing a **memory stall**



https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/5_CPU_Scheduling.html

Background (cont.)

- **Cache** sits between main memory and CPU registers
- Protection of memory required to ensure correct operation

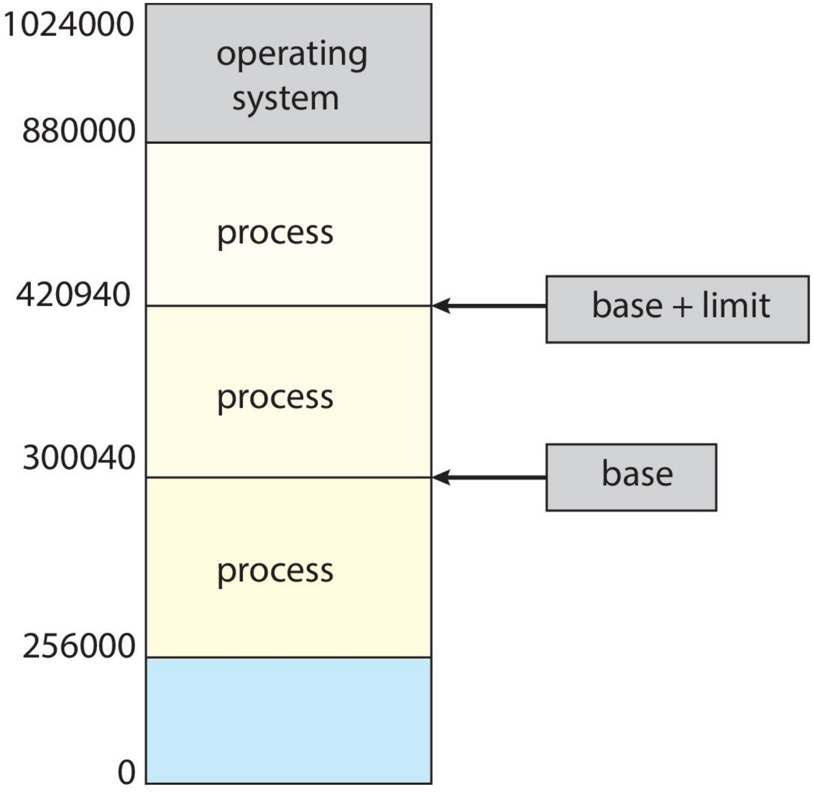


https://software.rajivprab.com/2018/04/29/myths-programmers-believe-about-cpu-caches/

Protection

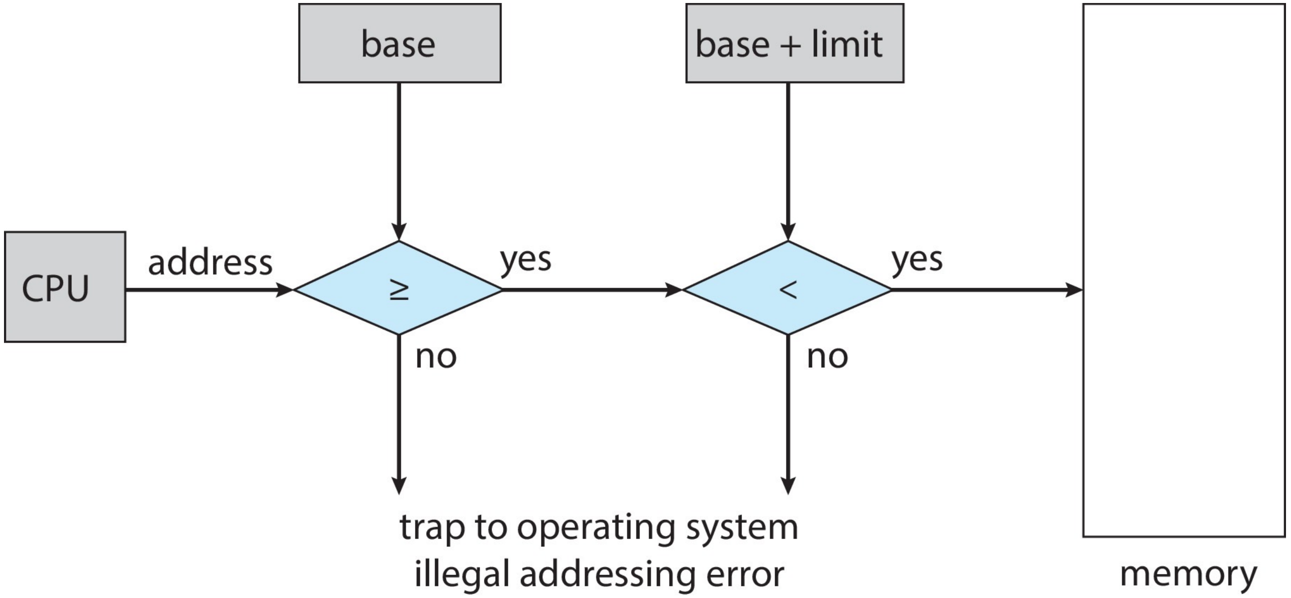
- Need to censure that a process can access only access those addresses in it address space.

- 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



- The instructions to loading the base and limit registers are **privileged**.

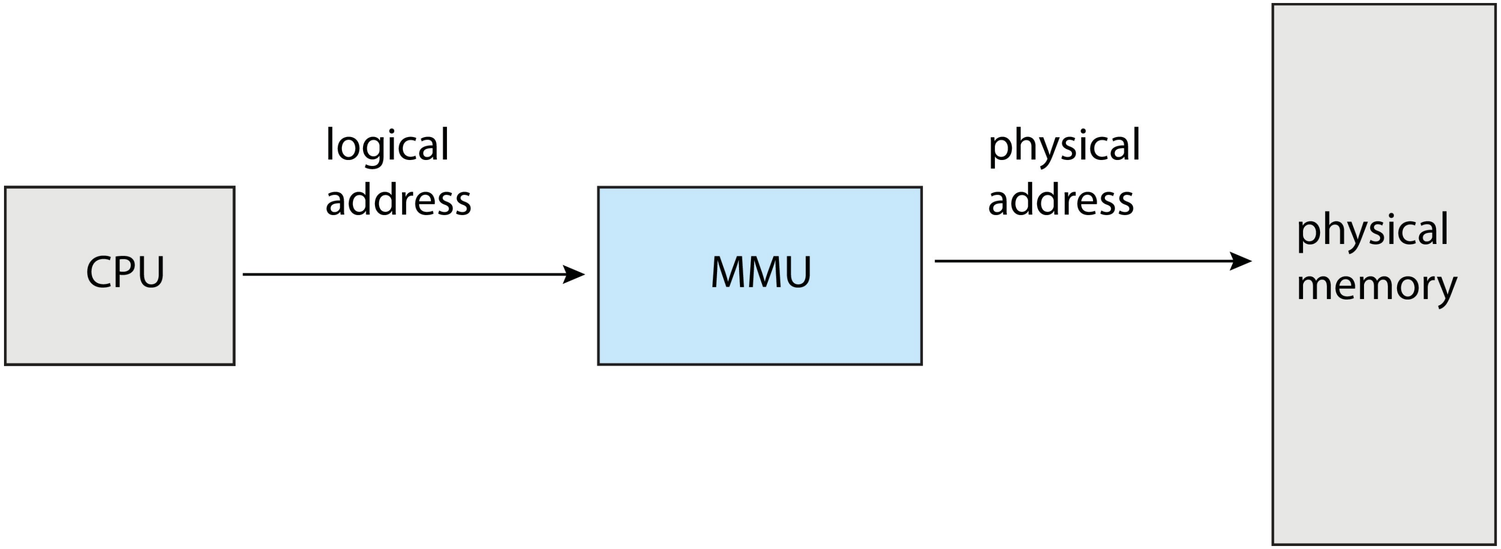
Logical vs. Physical Address Space

- The concept of a logical address space that is bound to a separate **physical address space** is central to proper memory management.

- **Logical address**
  - Generated by the CPU
  - Also referred to as **virtual address**
- **Physical address**
  - Address seen by the memory unit

Memory-Management Unit (MMU)

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



- Many methods possible, covered in the rest of this chapter

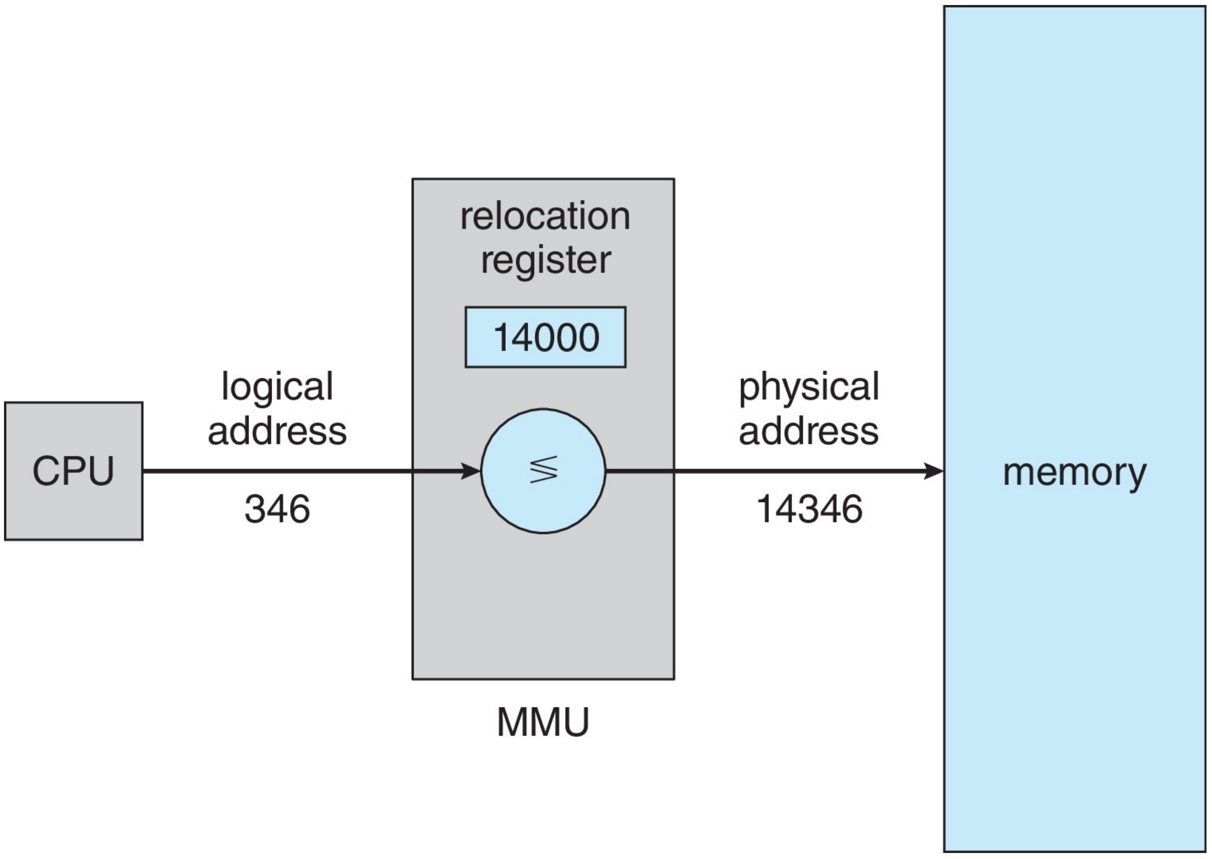
Memory-Management Unit (Cont.)

- Consider simple scheme. which is a generalization of the base-register scheme.

- The base register now called **relocation register.**

Memory-Management Unit (Cont.)

- 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 Unit (Cont.)

- The user program deals with logical addresses
  - It never sees the real physical addresses

- Execution-time binding occurs when reference is made to location in memory.
  - Logical address bound to physical addresses

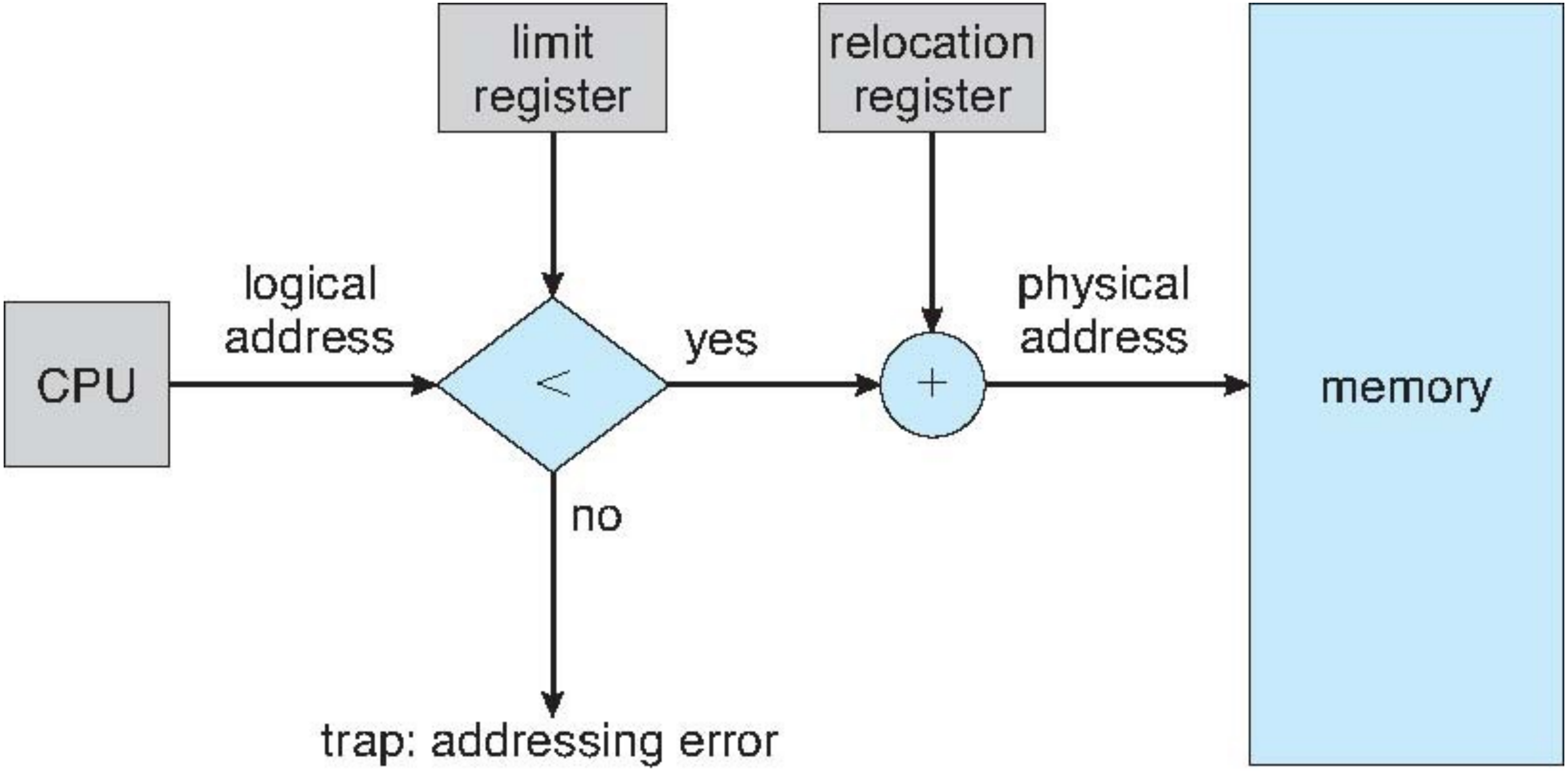
Contiguous Allocation

- Main memory must support both OS and user processes
- Limited resource, must allocate efficiently
- Contiguous allocation is **one early method**
- Main memory usually into two **partitions**:
  - Resident operating system, usually held in low memory with interrupt vector
  - User processes then held in high memory
  - Each process contained in single contiguous section of memory

Contiguous Allocation (cont.)

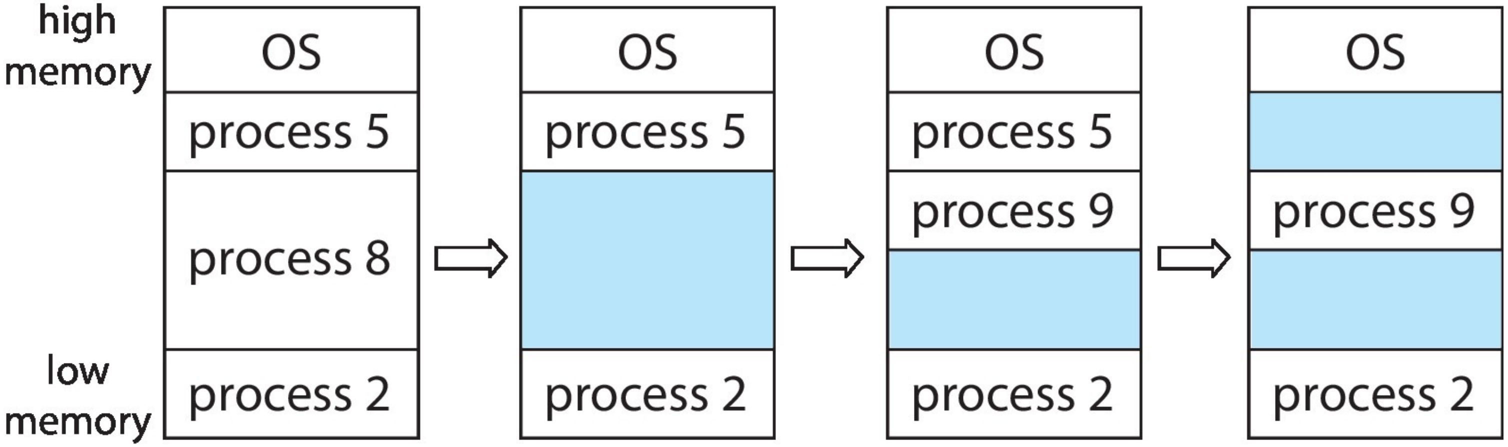
- **Relocation registers** used to protect user processes from each other, and from changing operating-system code and data
  - **Base register** contains value of smallest physical address
  - **Limit register** contains range of logical addresses – each logical address must be less than the limit register
  - **MMU** maps logical address ***dynamically***

Hardware Support for Relocation and Limit Registers



Multiple-partition Allocation

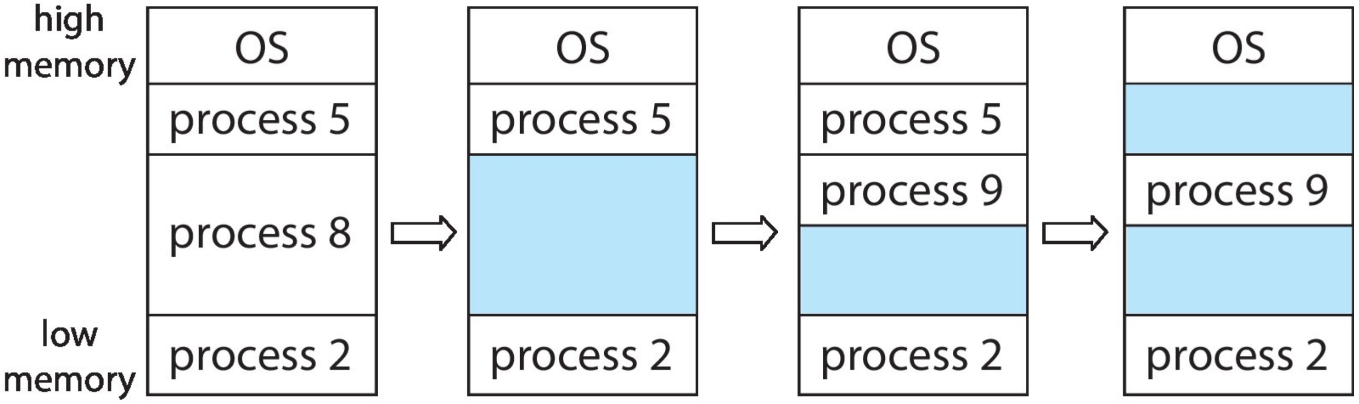
- Degree of multiprogramming limited by number of partitions
- **Variable-partition** sizes for efficiency (sized to a given process' needs)
- **Hole:** block of available memory
  - Holes of various size are scattered throughout memory



Multiple-partition Allocation (cont.)

- When a process arrives, it is allocated memory from a hole large enough to accommodate it.
- Process exiting frees its partition, adjacent free partitions combined
- Operating system maintains information about:

  a) allocated partitions b) free partitions (hole)



Dynamic Storage-Allocation Problem

How to satisfy a request of size ***n*** from a list of free holes?

- **First-fit**: Allocate the first hole that is big enough
- **Best-fit**: Allocate the smallest hole that is big enough; must search entire list, unless ordered by size
  - Produces the smallest leftover hole
- **Worst-fit**: Allocate the largest hole; must also search entire list
  - Produces the largest leftover hole

Dynamic Storage-Allocation Problem

First-fit and best-fit better than worst-fit in terms of speed and storage utilization