STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

CS 492: Operating Systems

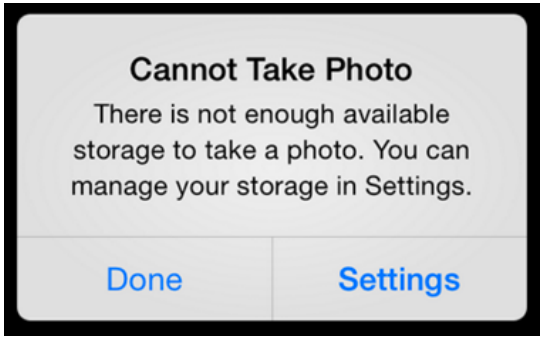
Memory Management

Instructor: Iraklis Tsekourakis

Email: itsekour@stevens.edu

Memory

- Paraphrase of Parkinson's Law, "*Programs expand to fill the memory available to hold them.*"

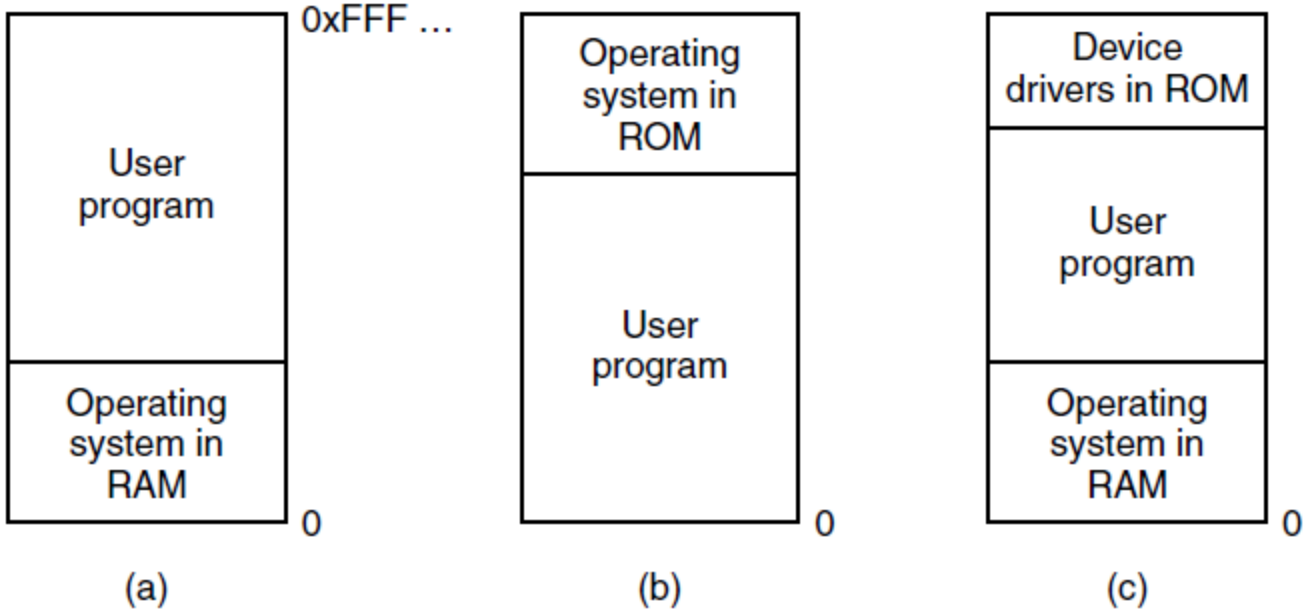


- Average home computer nowadays has 10,000 times more memory than the IBM 7094, the largest computer in the world in the early 1960s

Memory Management

- Goals of memory management
  - Convenient abstraction for programming
  - Provide isolation between different processes
  - Allocate scarce physical memory resources across processes
- Mechanisms
  - Virtual address translation
  - Paging and TLBs
  - Page table management
- Policies
  - Page replacement policies

No Memory Abstraction



Three simple ways of organizing memory with an operating system and one user process. Other possibilities also exist

Running Multiple Programs Without a Memory Abstraction

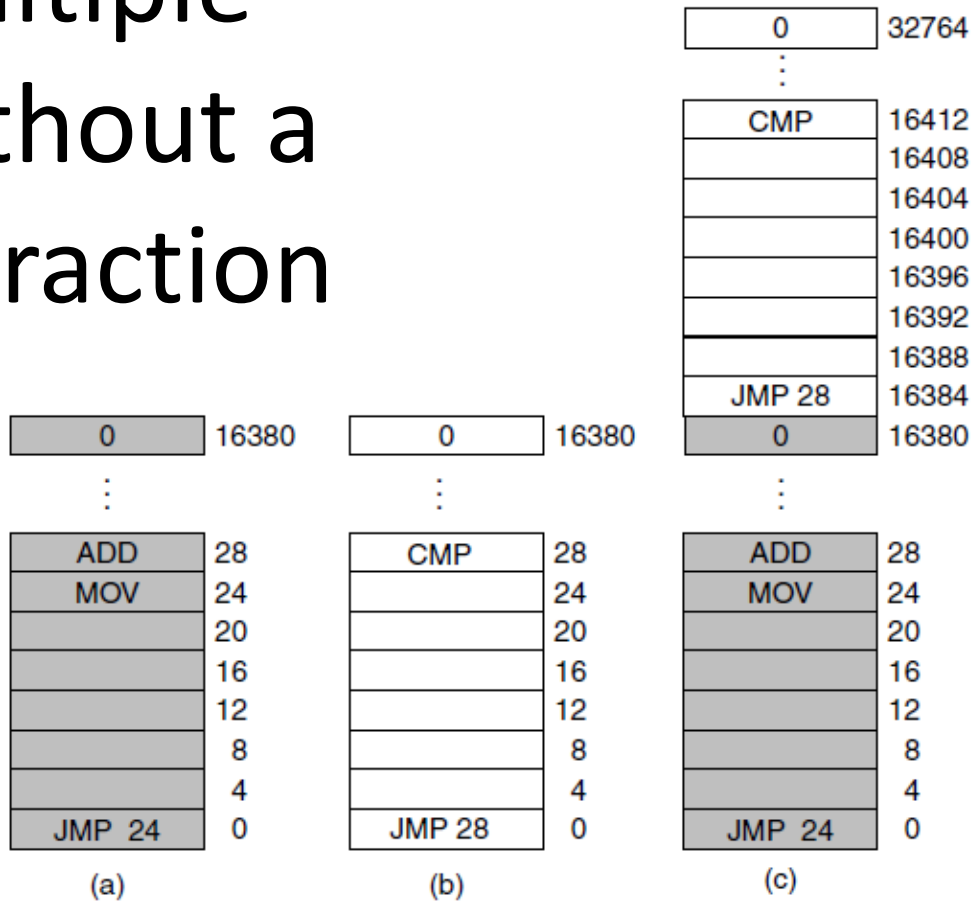
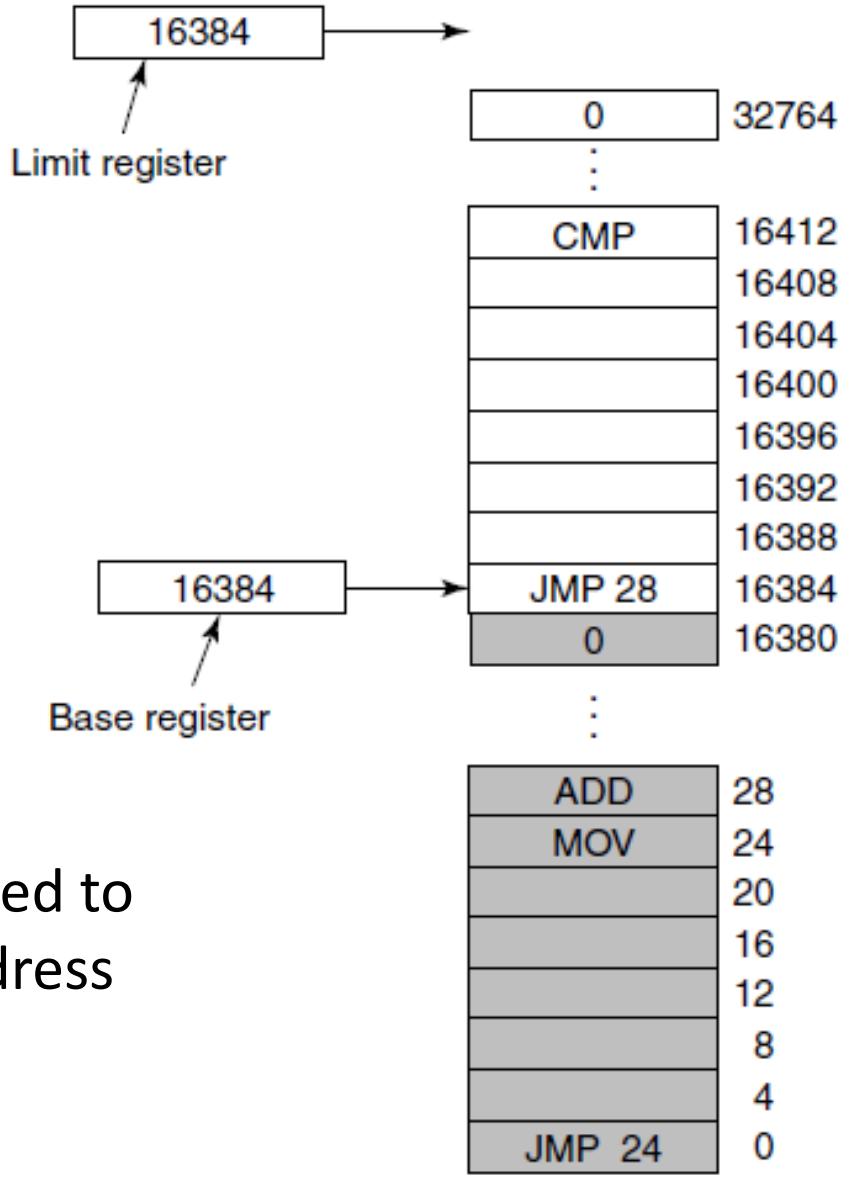


Illustration of the relocation problem. (a) A 16-KB program. (b) Another 16-KB program. (c) The two programs loaded consecutively into memory.

Address Space

- Physical memory exposed to processes
  - user processes trash OS
  - difficult multi-programming
- Two problems: Protection and Relocation
- Better solution: Address Space
  - the set of addresses that a process can use to address memory

Base and Limit Registers

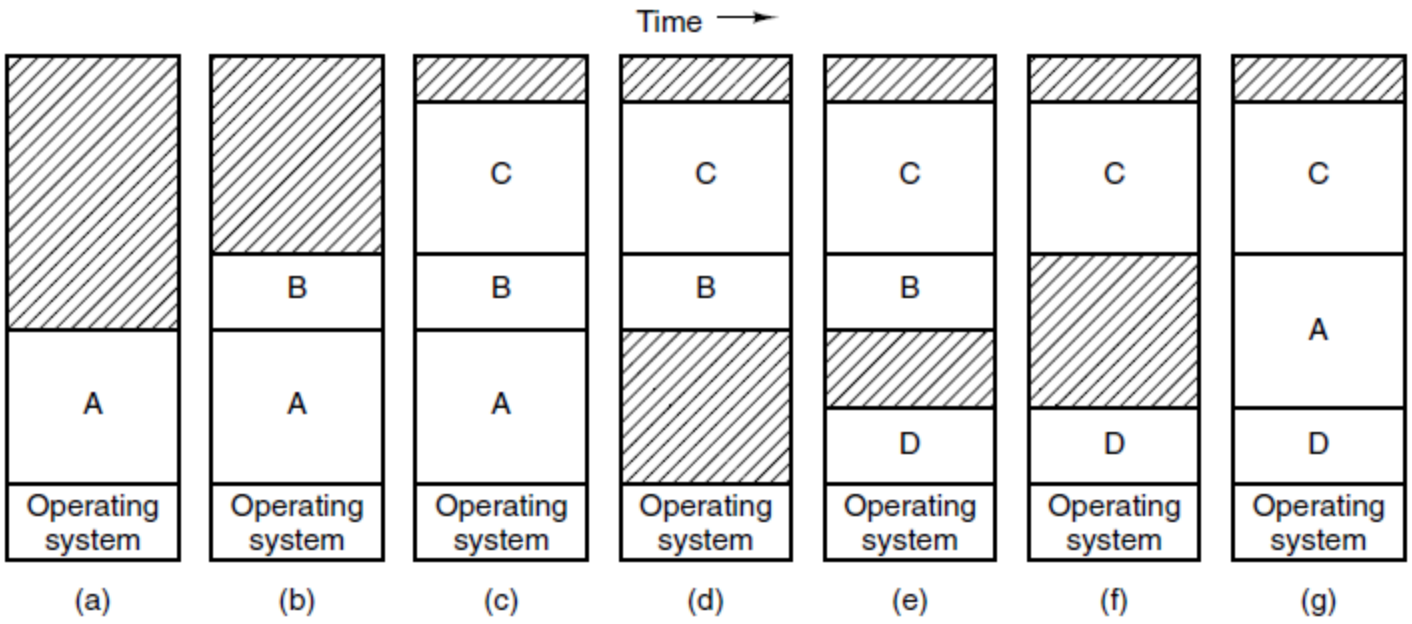


Base and limit registers can be used to give each process a separate address space.

Question?

- In the previous Figure the base and limit registers contain the same value, 16,384. Is this just an accident, or are they always the same?

Swapping (1)



Memory allocation changes as processes come into memory and leave it. The shaded regions are unused memory

Swapping (2)

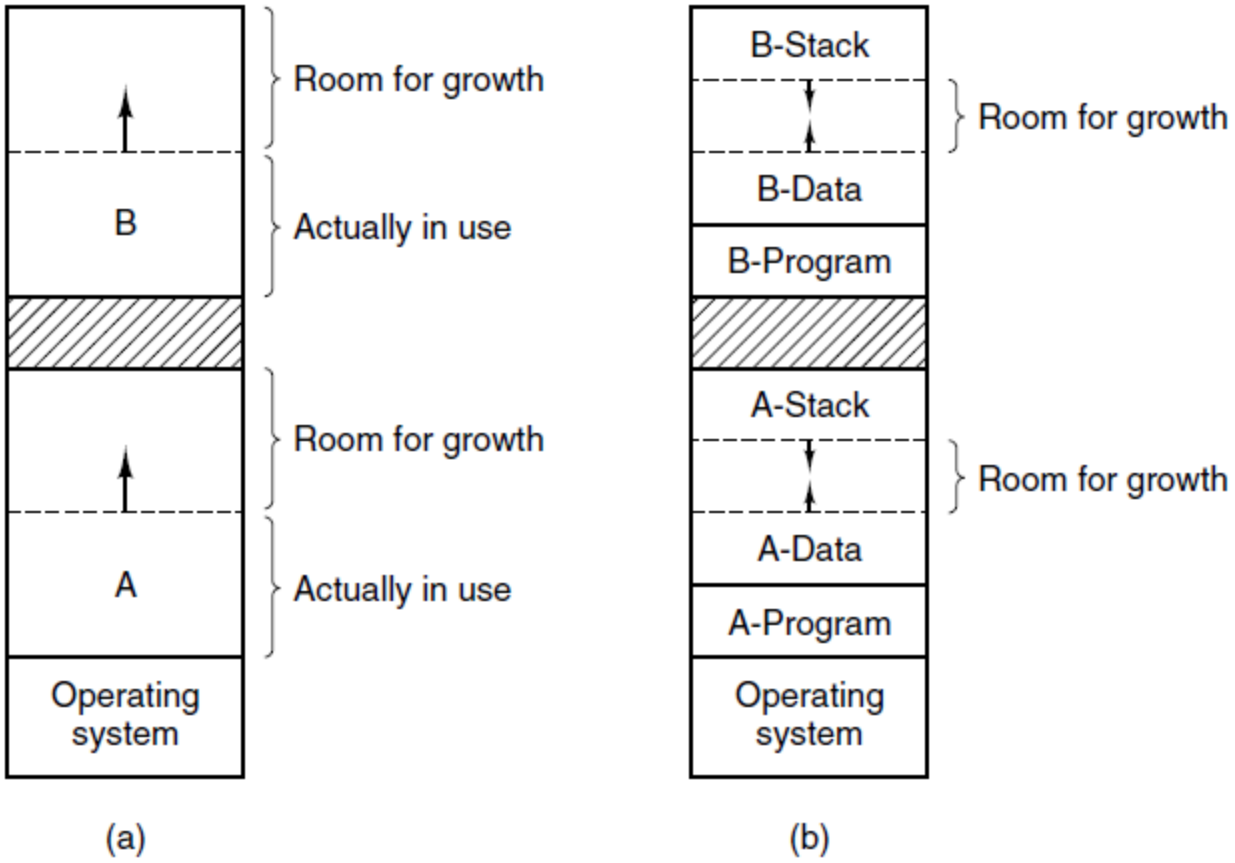
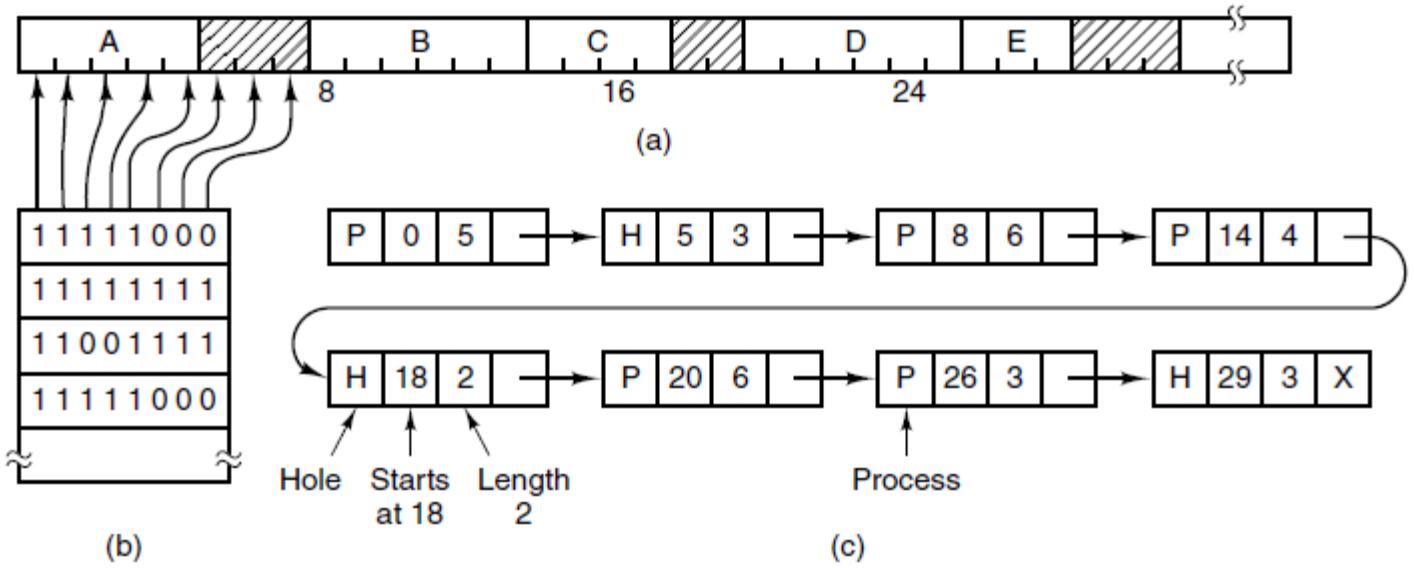


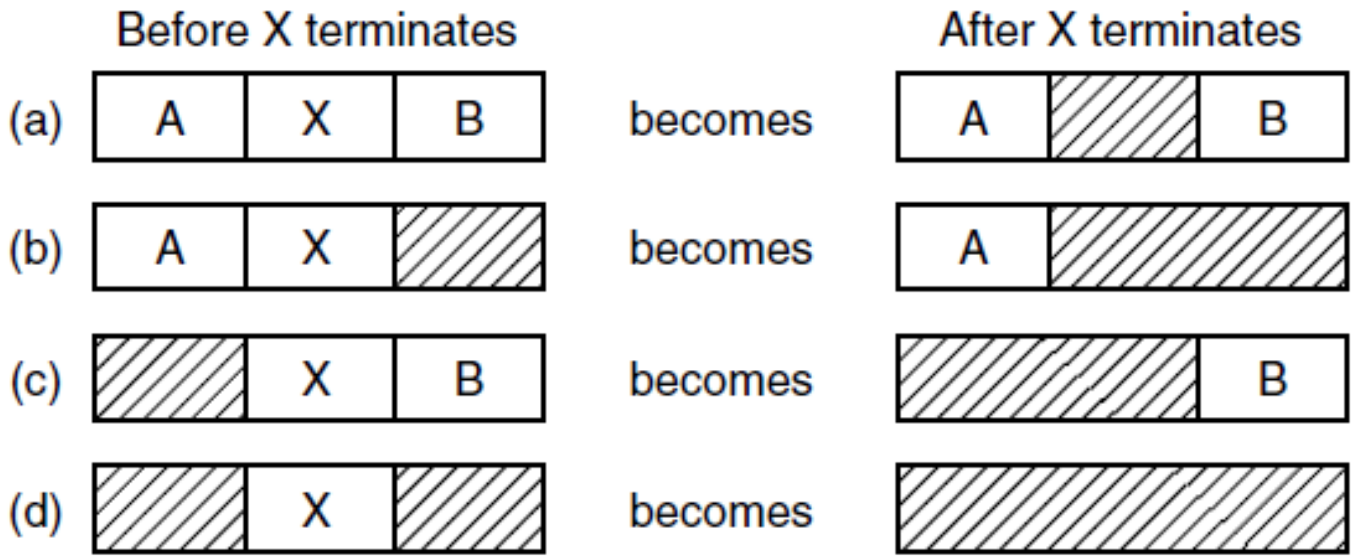
Figure 3-5. (a) Allocating space for a growing data segment.
(b) Allocating space for a growing stack and a growing data segment.

Memory Management with Bitmaps



(a) A part of memory with five processes and three holes. The tickmarks show the memory allocation units. The shaded regions (0 in the bitmap) are free. (b) The corresponding bitmap. (c) The same information as a list.

Memory Management with Linked Lists



Four neighbor combinations for the terminating process, *X*.

Memory Management Algorithms

- First fit
- Next fit
- Best fit
- Worst fit
- Quick fit

Virtual Memory

- Basic idea
  - Each program has its own address space, which is broken into *pages*
  - Not all pages have to be in physical memory to run the program
- Paging switching is hidden from processes
  - Processes still see an address space [0, max]
  - Movement of information to and from disk is handled by the OS alone

Virtual Memory (Cont.)

- Virtual memory (VM) isolates processes
  - Each process has its own isolated address space
  - One process cannot access memory addresses in others

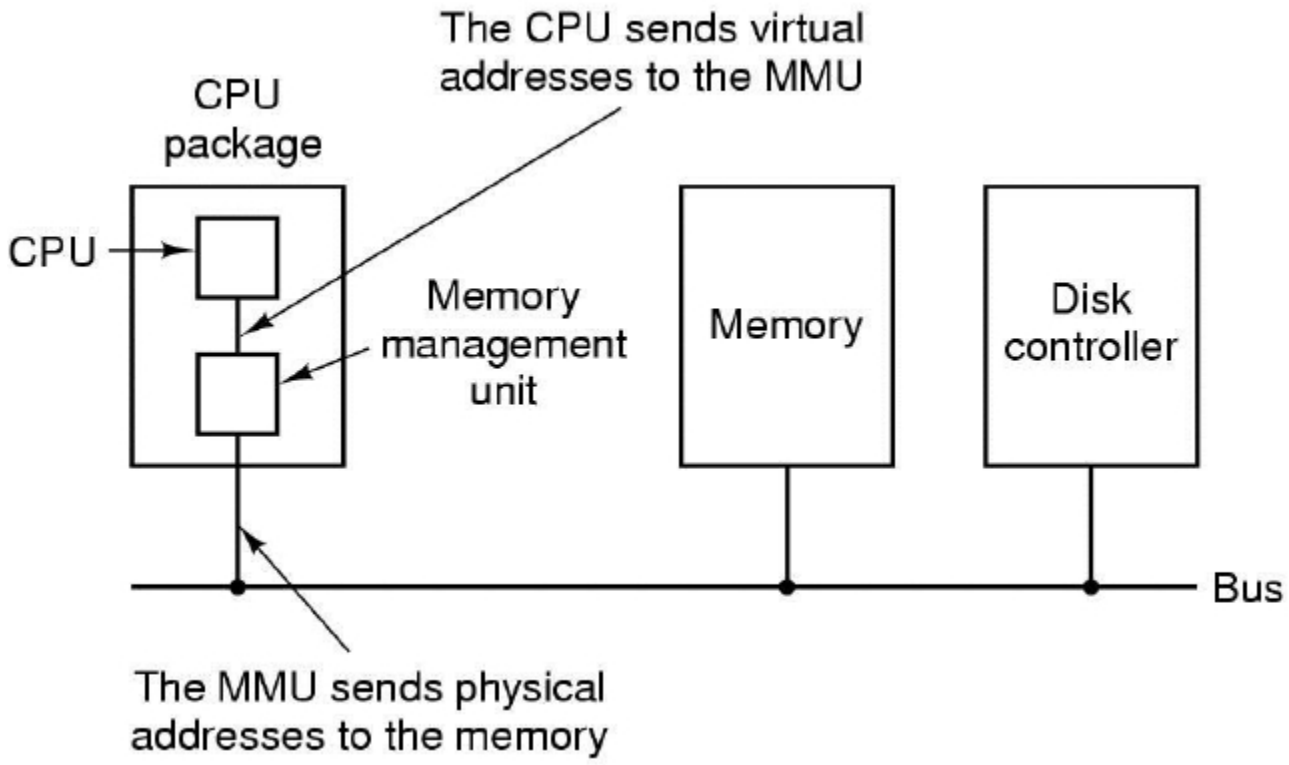
- VM requires both hardware and OS support
  - Hardware support: *memory management unit* (MMU) and *translation lookaside buffer* (TLB)
  - OS support: virtual memory system to control the MMU and TLB

Virtual Address

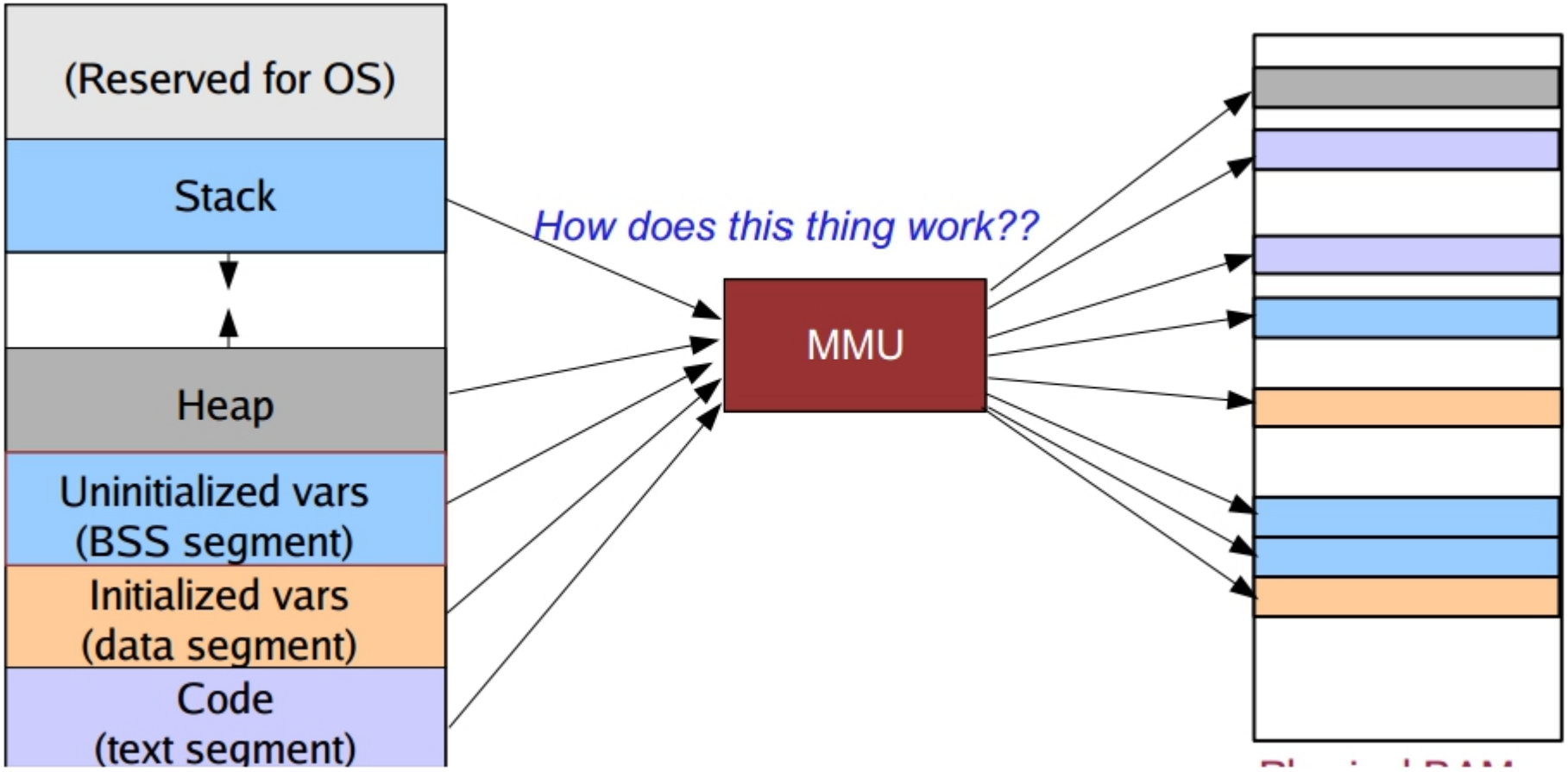
- A *virtual address* is a memory address that a process uses to access its own memory
  - The virtual address ≠ the physical RAM address
  - When a process accesses a virtual address, the memory management unit (MMU) translates the virtual address to a physical address

Position and Function of MMU

- MMU is usually on the same chip as the CPU
- Translation is transparent to user/program

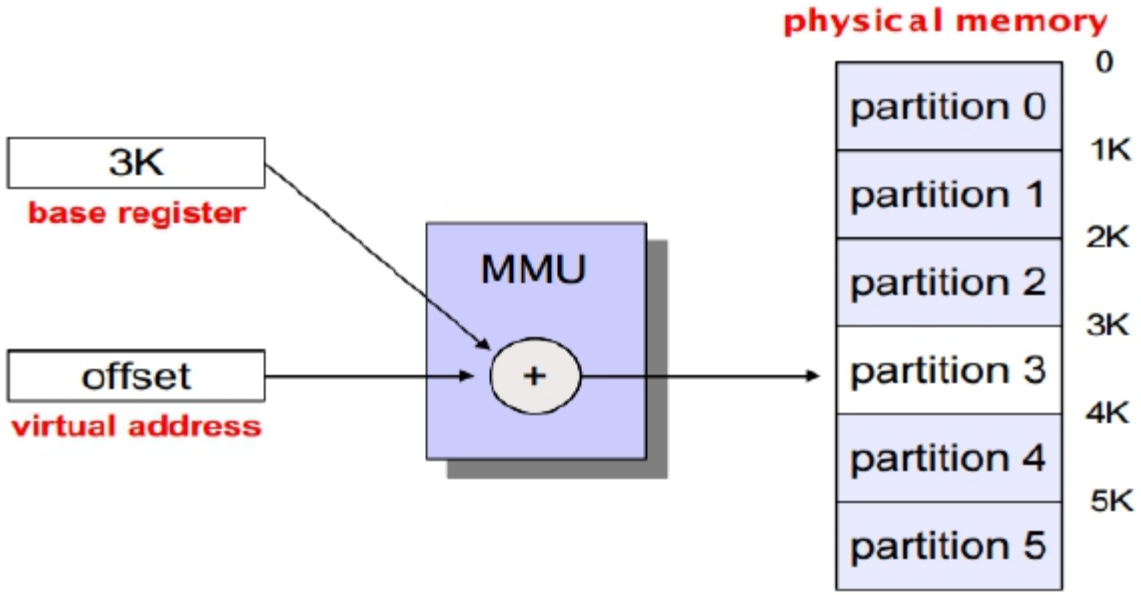


Address Translation Paradigm



Simple approach: Fixed Partitions

- Original memory management technique: Break memory into fixed-size partitions
  - Hardware requirement: *base register*
  - Translation from virtual to physical address: simply add base register to *vaddr (offset)*



Pros & Cons of Fixed Partitions

- Advantages:
  - Fast context switch – only need to update base register
- Disadvantages:
  - Internal fragmentation
    - Must consume entire partition, rest of partition is "wasted"
  - Static partition sizes
    - No single size is appropriate for all programs!

Variable Partitions

- Obvious next step: Allow variable-sized partitions
  - Now requires both a base register and a limit register for performing memory access
  - Solves the internal fragmentation problem: size partition based on process needs

Any problems here?