Segmentation, Segmentation with Paging, Virtual Memory Concept

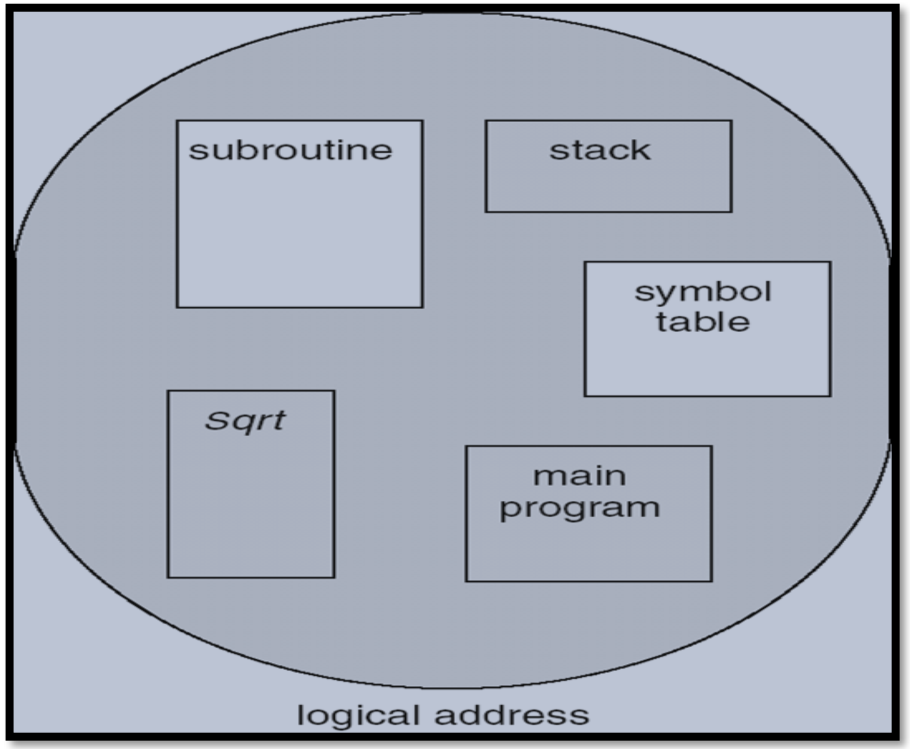
The Content is prepared with the help of existing text books mentioned below:

1. Silberschatz, Abraham, Peter B. Galvin, and Greg Gagne. *Operating system concepts with Java*. Wiley Publishing, 2009.
2. Stallings, William. *Operating Systems 5th Edition*. Pearson Education India, 2006.
3. Tannenbaum, Andrew S. "Modern Operating Systems, 2009."

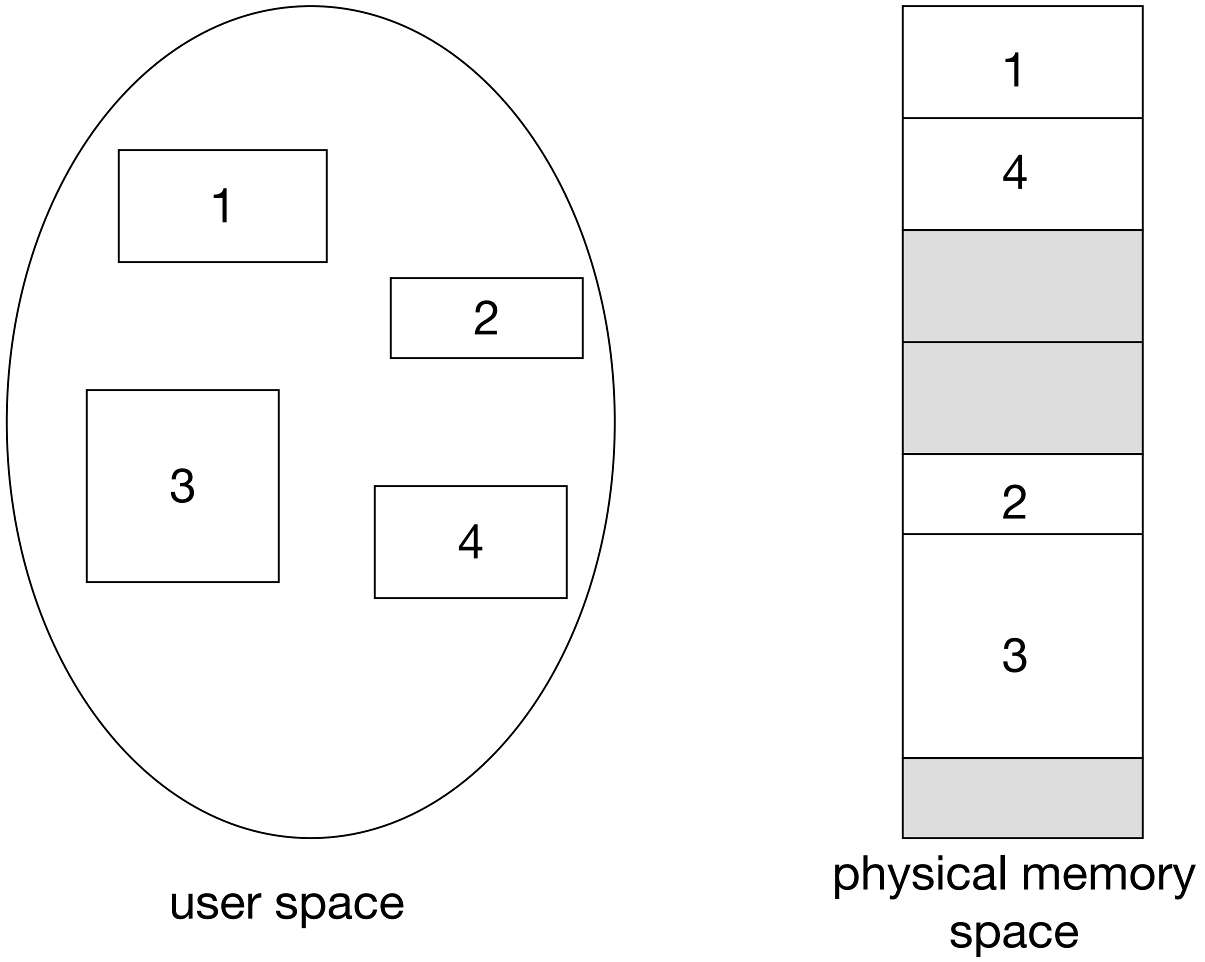
Segmentation

- Memory-management scheme that supports user view of memory
- A program is a collection of segments
  - A segment is a logical unit such as:
    - main program
    - procedure
    - function
    - method
    - object
    - local variables, global variables
    - common block
    - stack
    - symbol table
    - arrays

User’s View of a Program



Logical View of Segmentation



Segmentation Architecture

- Logical address consists of a two tuple:

  <segment-number, offset>,

- **Segment table** – maps two-dimensional physical addresses; each table entry has:
  - **base** – contains the starting physical address where the segments reside in memory
  - **limit** – specifies the length of the segment

- **Segment-table base register (STBR)** points to the segment table's location in memory

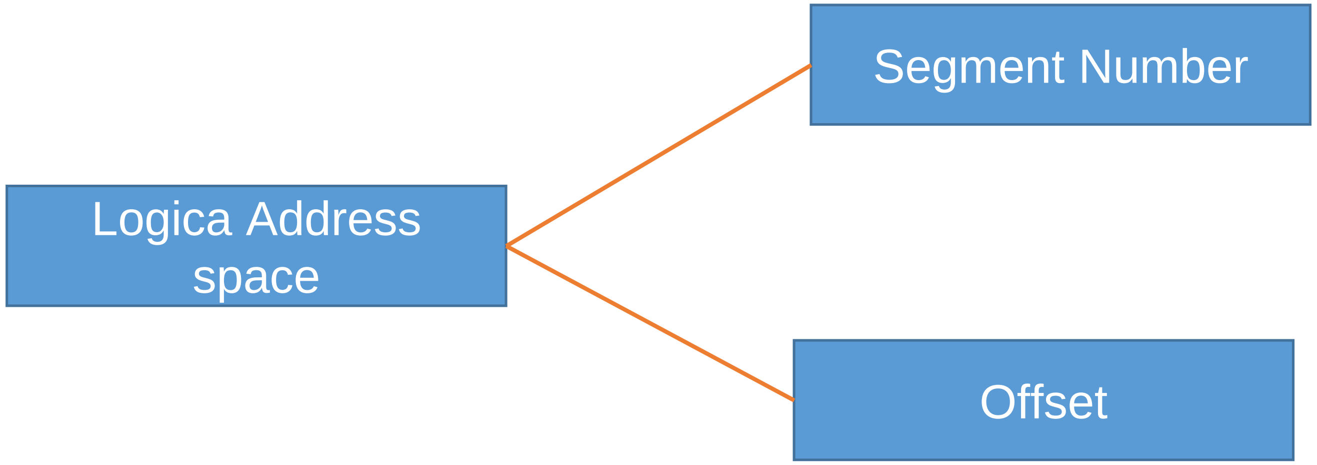
- **Segment-table length register (STLR)** indicates number of segments used by a program;

  segment number ***s*** is legal if ***s*** < **STLR**

Segmentation Architecture

- Protection
  - With each entry in segment table associate:
    - validation bit = 0 ⇒ illegal segment
    - read/write/execute privileges
- Protection bits associated with segments; code sharing occurs at segment level
- Since segments vary in length, memory allocation is a dynamic storage-allocation problem
- A segmentation example is shown in the following diagram

Logical addressing in Segmentation

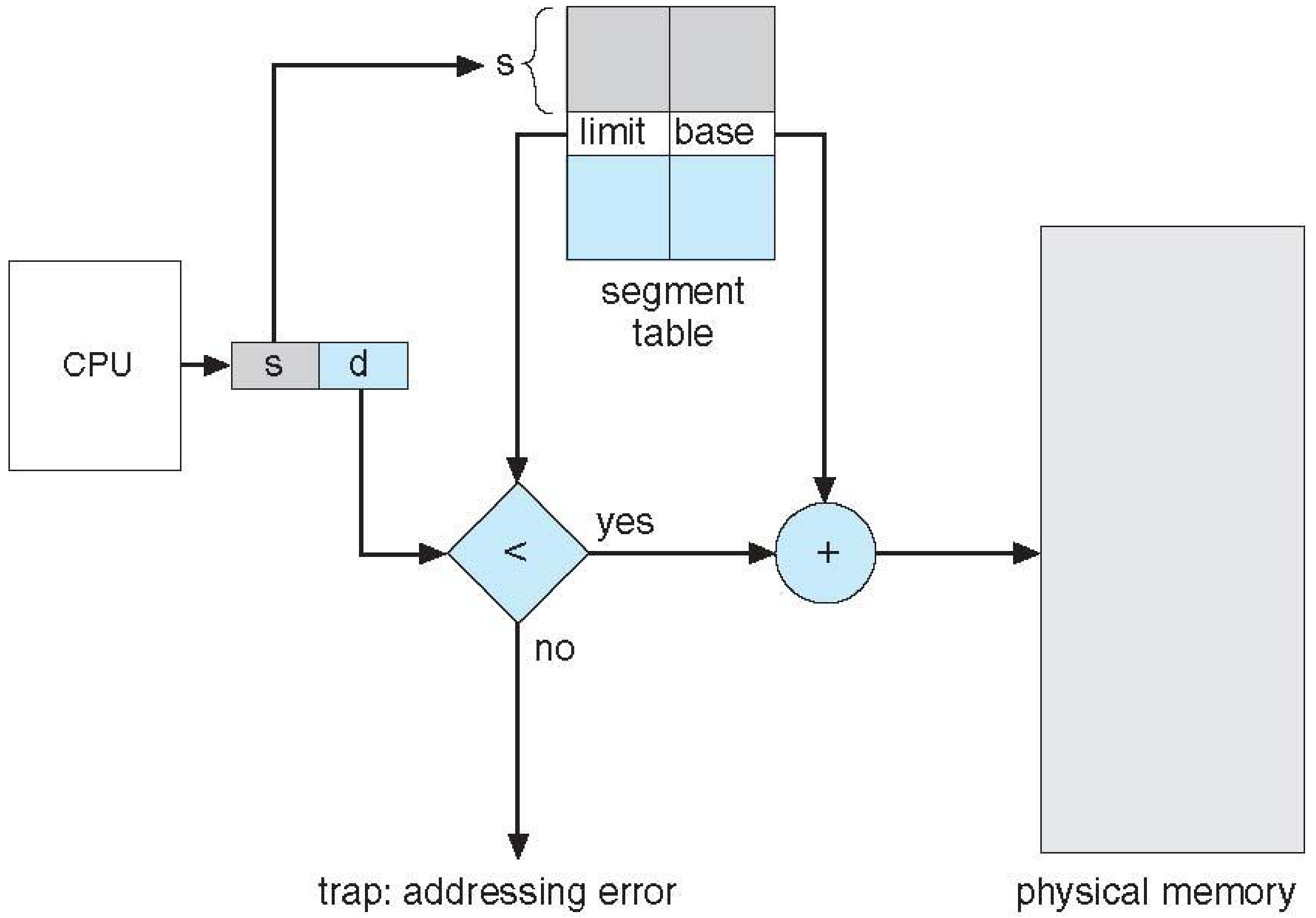


The mapping of the logical address to the physical address is done with the help of the segment table.

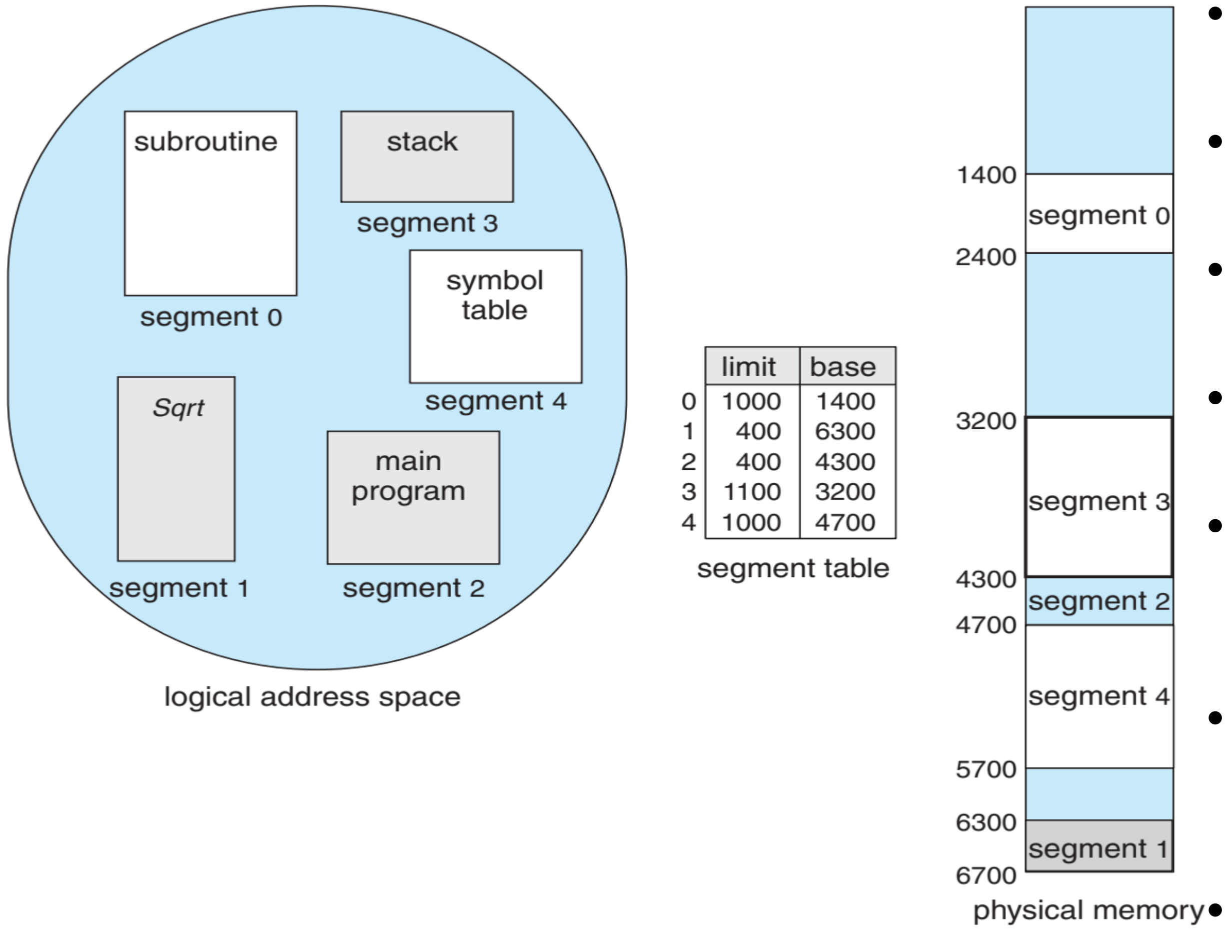
Segment Limit	Segment Base	Other Bits

Cont.

Segmentation Hardware



Example of Segmentation



	limit	base
0	1000	1400
1	400	6300
2	400	4300
3	1100	3200
4	1000	4700

- Five Segments numbered from 0 to 4.
- Segment are stored in physical memory.
- Segment table has a separate entry for each segment.
- Segment 2 is 400 bytes long and begins at location 4300.
- A reference to byte 53 of segment 2 is 400 bytes long and begins at location 4300.
- A reference to byte 53 of segment 2 is mapped onto location 4300+53=4353
- A reference to segment3 byte 852 is mapped to 3200+852=4052

Advantages of Segmentation

- No internal fragmentation
- Segment tables consume less memory then page tables.
- Because of the small segment table, memory reference is easy
- Lends itself to sharing data among processes.
- As the individual lines of a page do not form one logical unit, it is not possible to set a particular access right to a page.
- Easier to grow and shrink individual segments
- More efficient use of physical space

Disadvantages of Segmentation

- External Fragmentation
- Costly memory management algorithm
- Unequal size of segments is not good in the case of swapping

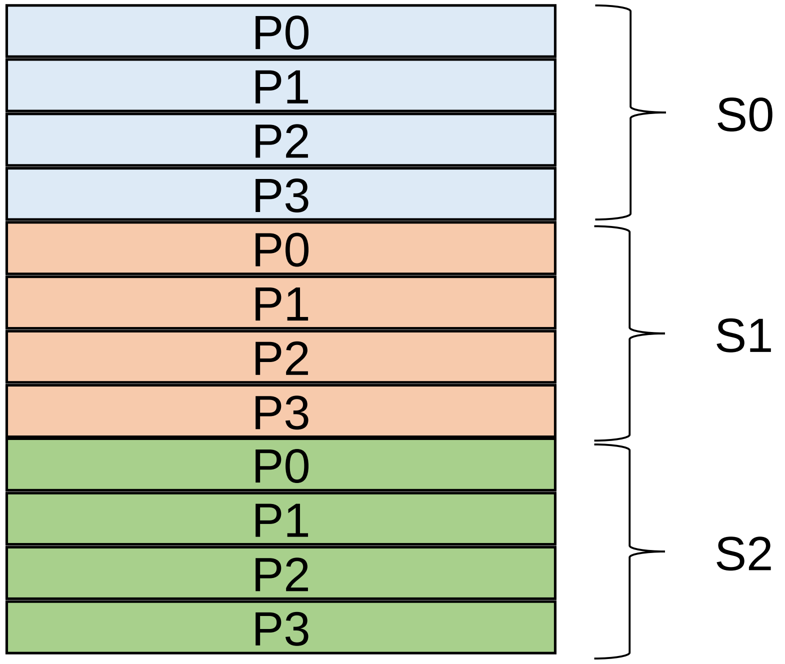
Segmentation with Paging

- Most architectures support segmentation and paging
- Basic idea,
  - segments exist in virtual address space
  - base address in segment descriptor table is a virtual address
  - use paging mechanism to translate this virtual address into a physical address
- Now an entire segment does not have to be in memory at one time
  - only the part of the segment that we need will be in memory

Segmentation with Paging

- Paged segmentation is a memory management scheme which combines features of paging system and segmentation system .
- User’s logical address space is divided into segments and each segment is divided into pages .
- User specified his logical address with component
  - A Segment number
  - A Relative offset.
- Paging hardware splits the offset into a page number and an offset within that page.

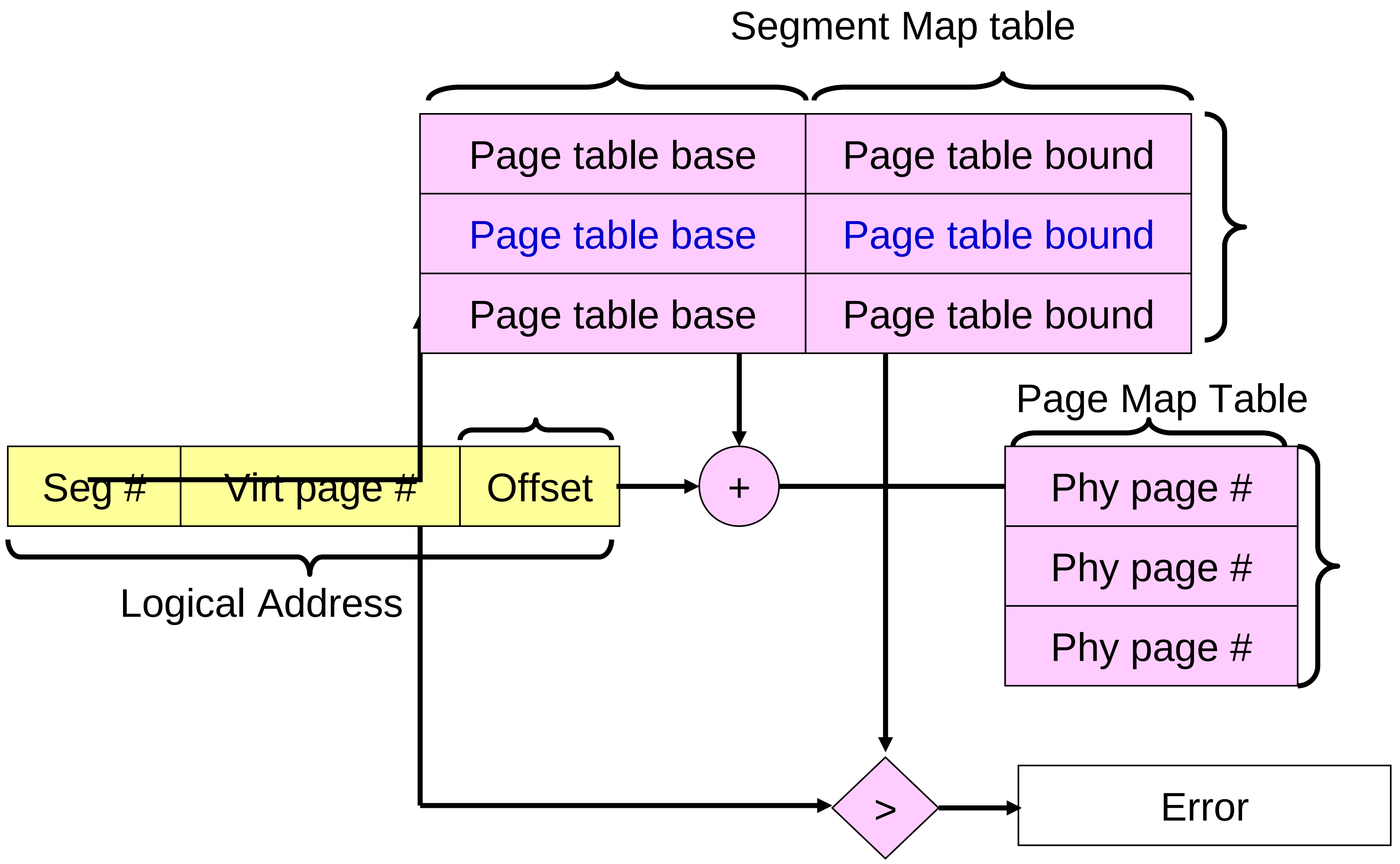
Segmentation with Paging



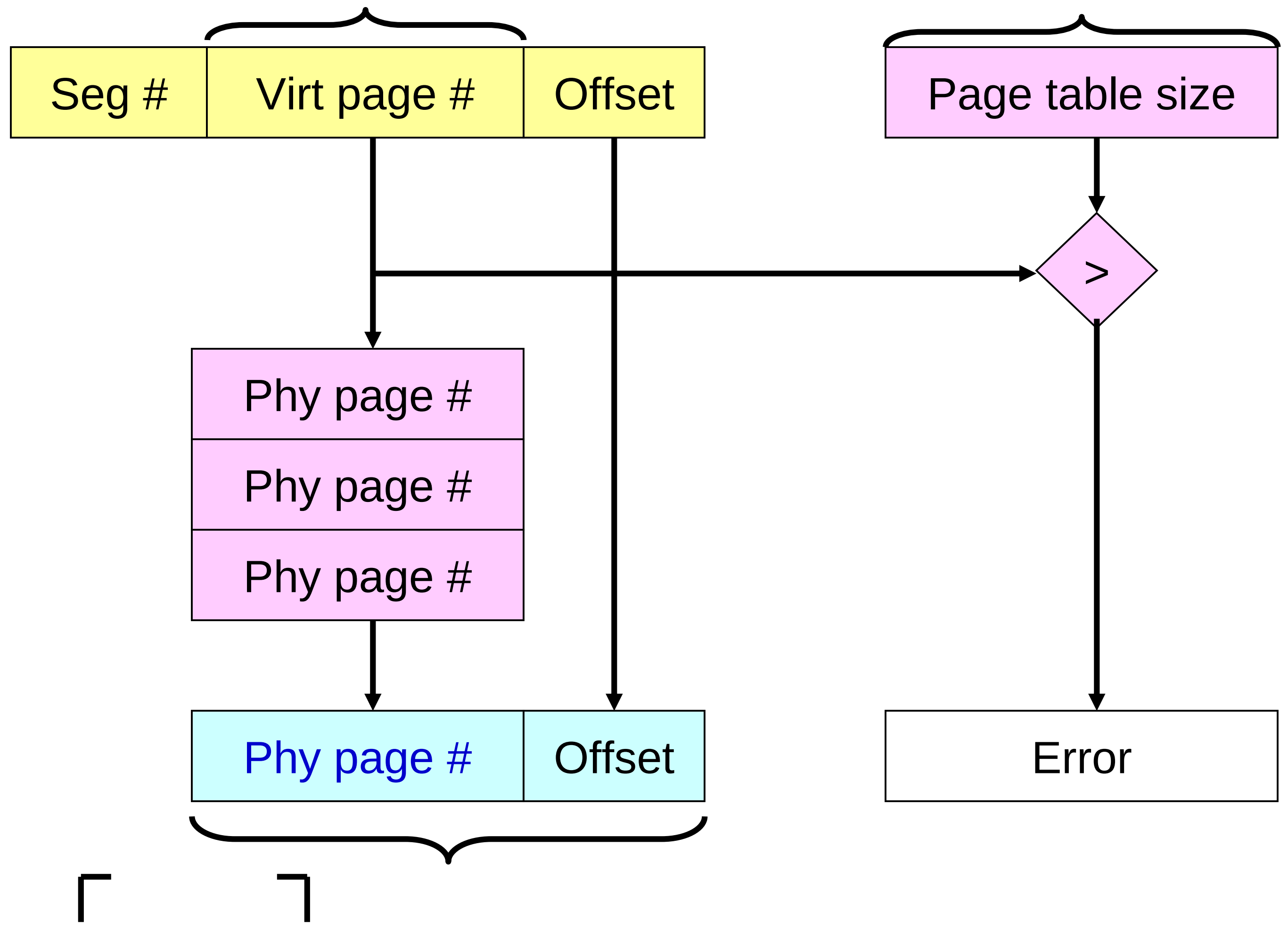
S0 : Segment 0
S1 : Segment 1
S2 : Segment 2
P0 : Page 0
P1 : Page 1
P2 : Page 2
P3 : Page 3

Logical address space of one user process. Logical address space is divided into segments and segments and are divided into pages

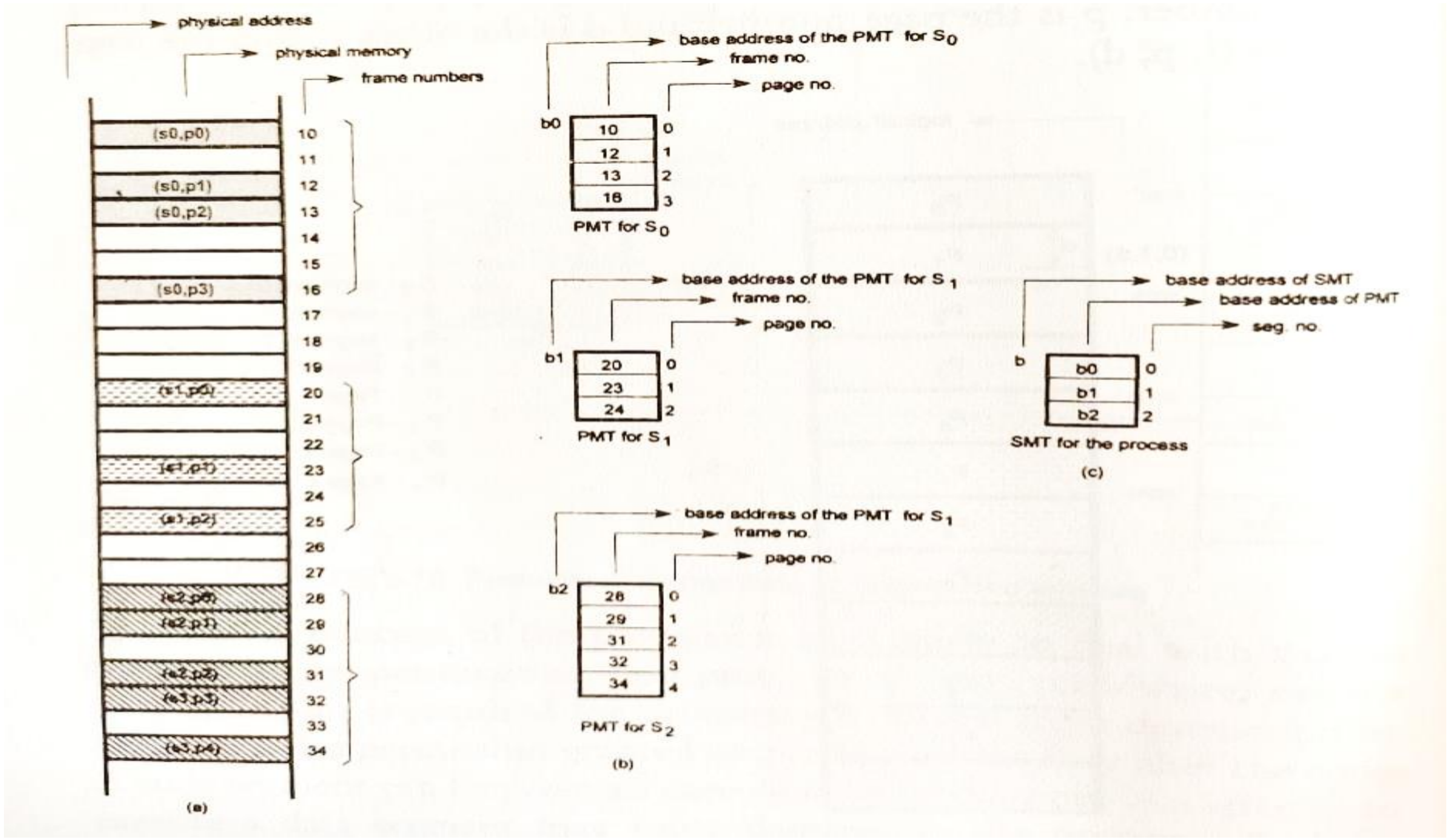
Segmented Paging



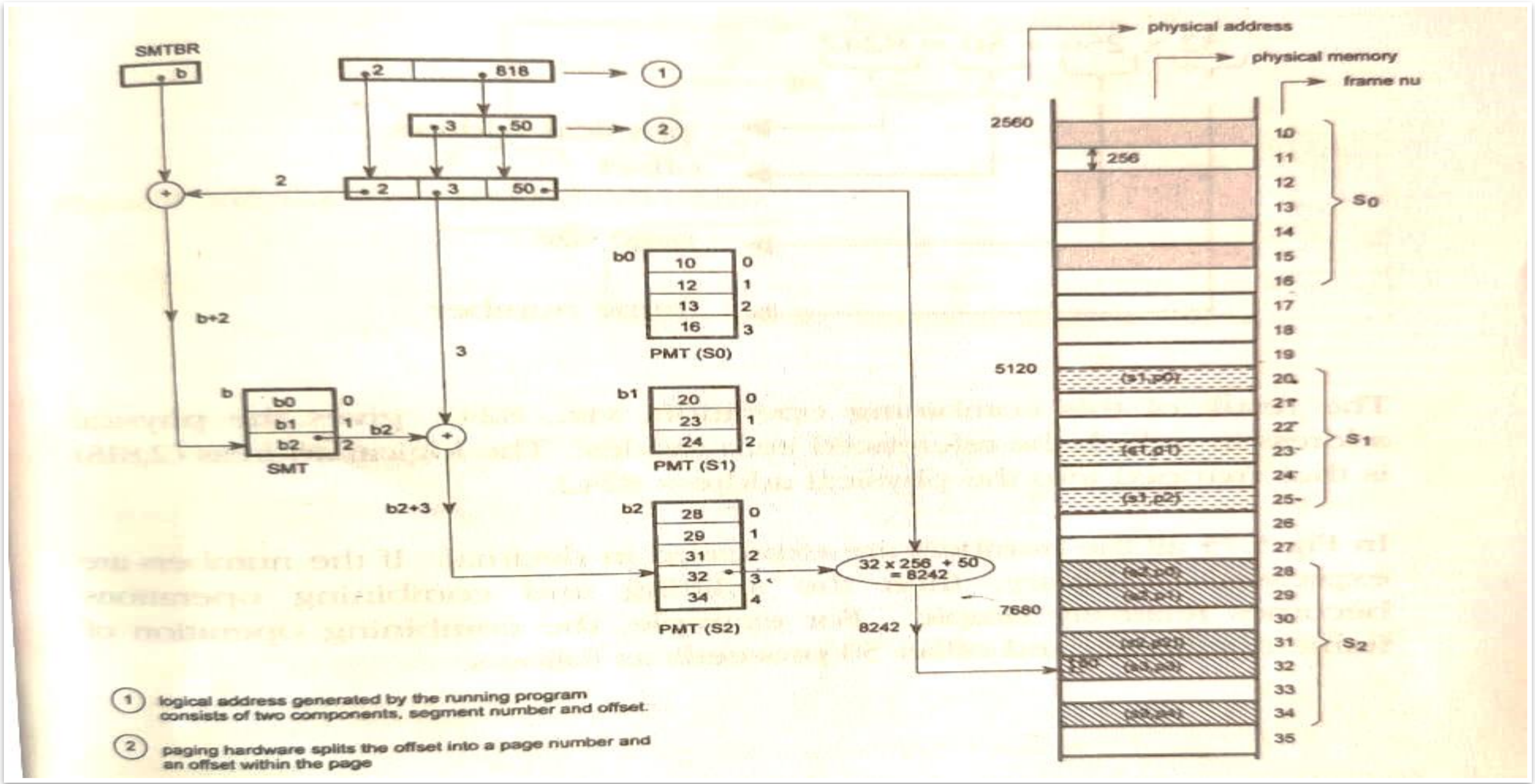
Segmented Paging



Segmented Paging



Segmented Paging



① logical address generated by the running program consists of two components, segment number and offset.

② paging hardware splits the offset into a page number and an offset within the page

Segmented Paging Translation

- Logical_address = segment_number:page_number:offset
- page_table = segment_table[segment_number]
- physical_page_number =page_table[page_number]
- physical_address = physical_page_number:offset

Pros and Cons of Segmented Paging

- + Code sharing
- + Reduced memory requirements for page tables
- - Higher overhead and complexity
- - Page tables still need to be contiguous
- - Each memory reference now takes two lookups

Virtual Memory

Objective

- To describe the benefits of a virtual memory system
- To explain the concepts of demand paging, page-replacement algorithms, allocation of page frames and thrashing.
- To discuss the principle of the working-set model.

Background

- In practice, most real processes do not need all their pages, or at least not all at once, for several reasons:
  - **Arrays** are often over-sized
  - Certain features of **programs are rarely used.**
- The ability to load only the portions of processes that were actually needed has several benefits:
  - Programs could be written for a much **larger address space.**
  - **more memory left for other programs**, **improving CPU utilization and system throughput.**
  - **Less I/O is needed for swapping** processes in and out of RAM, speeding things up.

Virtual Memory Concept

- Virtual Memory is a memory management scheme that supports the execution of partially loaded program.
- Virtual memory allow the execution of a process whose logical address space exceeds far beyond the physically address ace available on the machine on which the program is being executed.
- Virtual memory manager create an illusion that the physical memory is stretched far beyond the actual physical memory available on the machine.

Virtual Memory Concept

- Code needs to be in memory to execute, but entire program rarely used
  - Error code, unusual routines, large data structures
- Entire program code not needed at same time
- Consider ability to execute partially-loaded program
  - Program no longer constrained by limits of physical memory
  - Each program takes less memory while running -> more programs run at the same time
    - Increased CPU utilization and throughput with no increase in response time or turnaround time
  - Less I/O needed to load or swap programs into memory -> each user program runs faster

Virtual Address Space

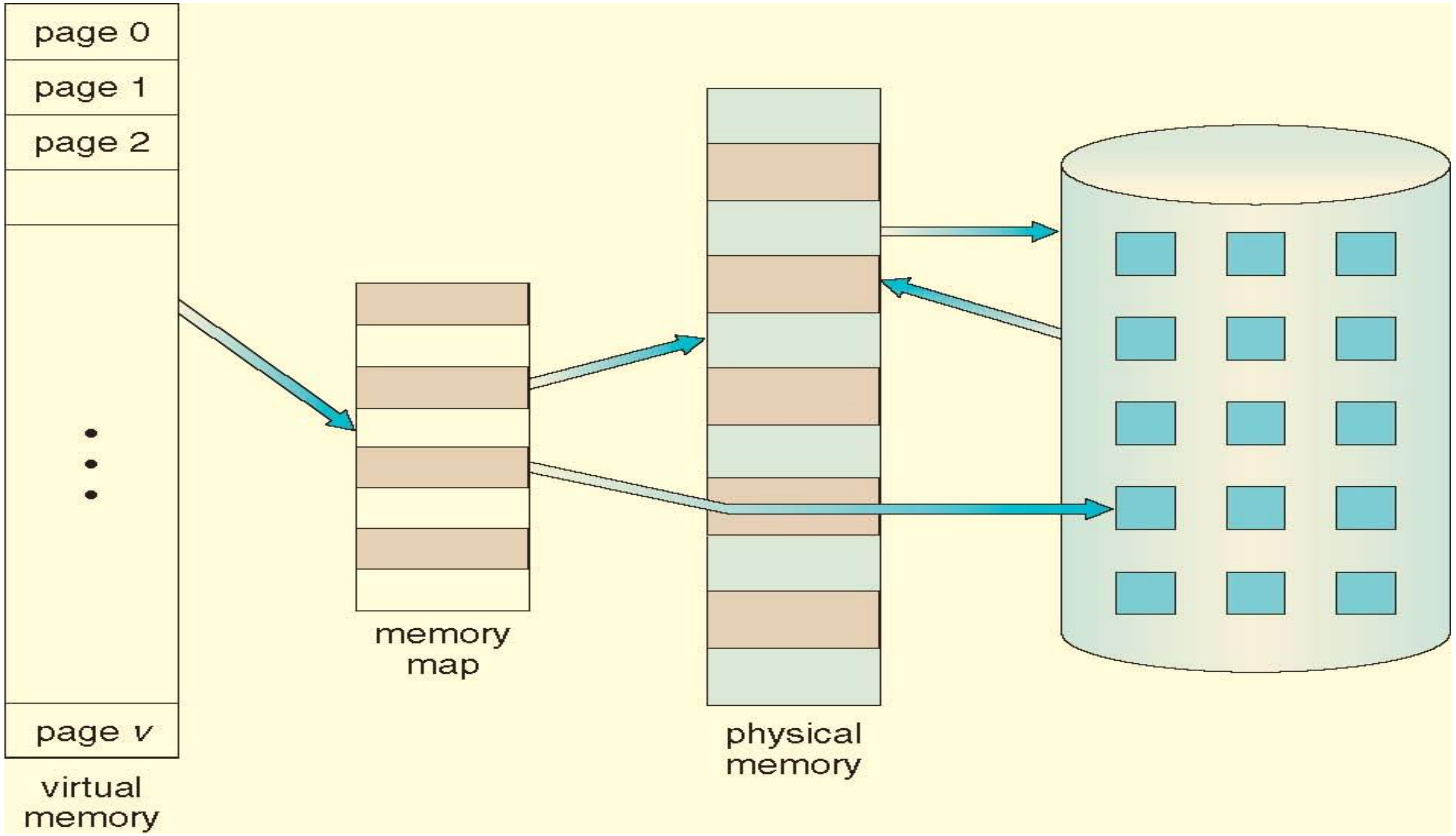
Virtual address space – logical view of how process is stored in memory

- Usually start at address 0, contiguous addresses until end of space
- Meanwhile, physical memory organized in page frames
- MMU must map logical to physical

Virtual memory can be implemented via:

- Demand paging
- Demand segmentation

Virtual Memory That is Larger Than Physical Memory



Thanks