OPERATING SYSTEMS

Memory Management -4

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

OPERATING SYSTEMS

Course Syllabus - Unit 3

UNIT 3: Memory Management

Main Memory: Hardware and control structures, OS support, Address translation, Swapping, Memory Allocation (Partitioning, relocation), Fragmentation, Segmentation, Paging, TLBs context switches .Virtual Memory - Demand Paging, Copy-on-Write, Page replacement policy - LRU (in comparison with FIFO & Optimal), Thrashing, design alternatives - inverted page tables, bigger pages. Case Study: Linux/Windows Memory.

Course Outline - Unit 3

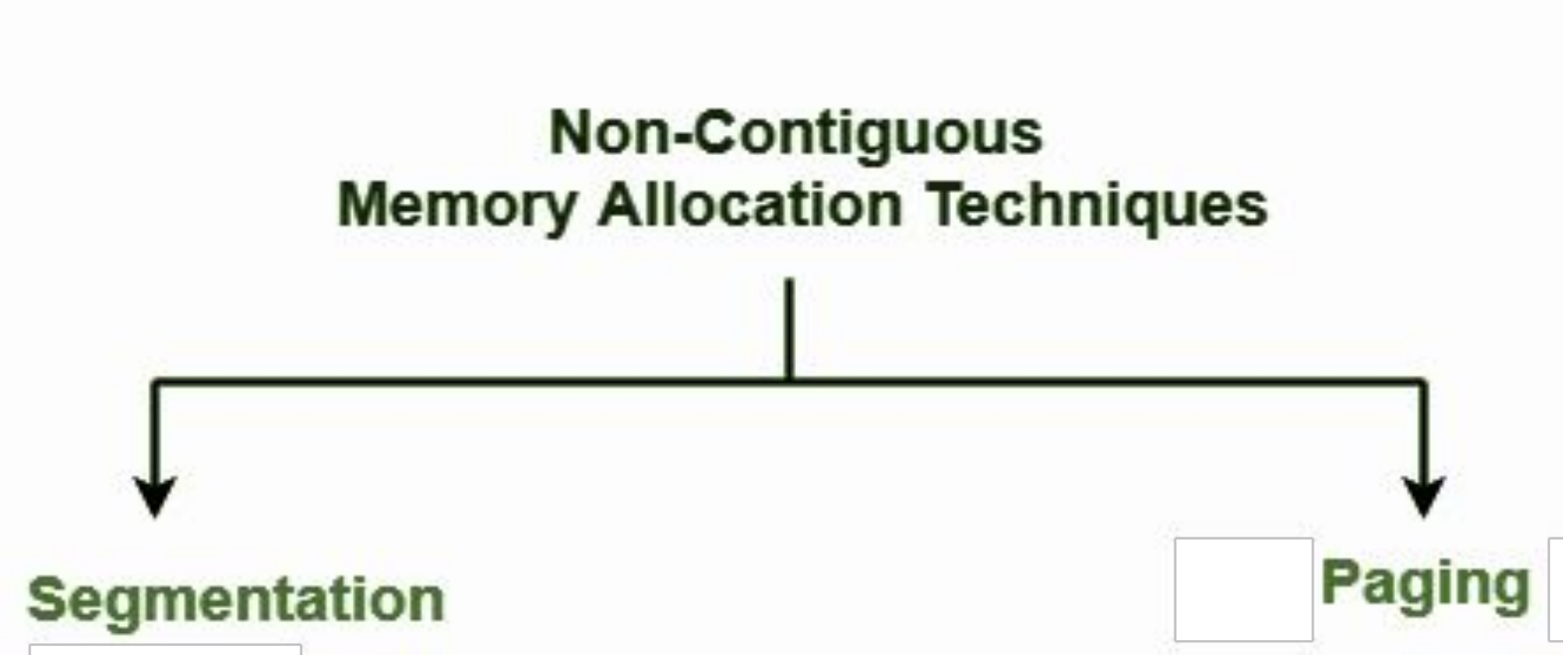
25	8.1	Main Memory: Hardware and control structures, OS support, Address translation,	8	64.2
26	8.2-8.3	Swapping, Memory Allocation (Partitioning, relocation), Fragmentation,	8	
27	8.4	Segmentation	8	
28	8.5	Paging	8	
29	8.5	TLBs context switches	8	
30	8.6	Structure of page tables	8	
31	8.6.3,8.7	design alternatives - Inverted page tables, bigger pages	8	
32	9.1-9.2	Virtual Memory - Demand Paging	9	
33	9.3,9.4.1-9.4.3	Copy-on-Write, Page replacement: Basic page replacement (FIFO page replacement and optimal page replacement)	9	
34	9.4.4, 9.5	LRU Page replacement, Allocation of frames	9	
35	9.6	Thrashing	9	
36	9.10	Case Study: Linux/Windows Memory	9	

Topic Outline

- Non Contiguous memory allocation
- Segmentation
- User View of the Program
- Logical View of Segmentation
- Segmentation Architecture
- Segmentation Hardware

Non - Contiguous Memory Allocation

- Non-contiguous memory allocation is a memory allocation technique.
- It allows to store parts of a single process in a non-contiguous fashion.
- Thus, different parts of the same process can be stored at different places in the main memory.



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

Segmentation

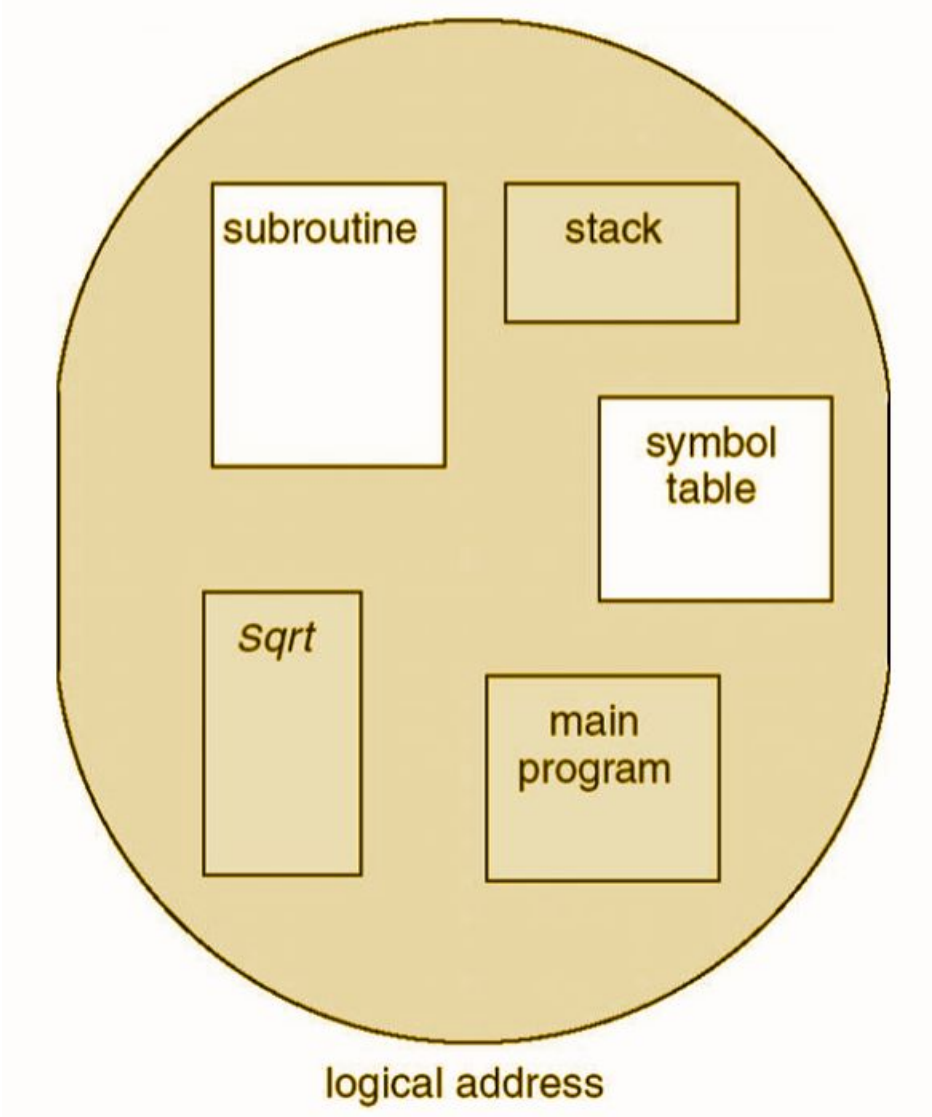
- Segmentation is a memory management technique which supports user's view of memory. This technique of division of a computer's primary memory into sections called segmentation leads to segments.

- Types of Segmentation

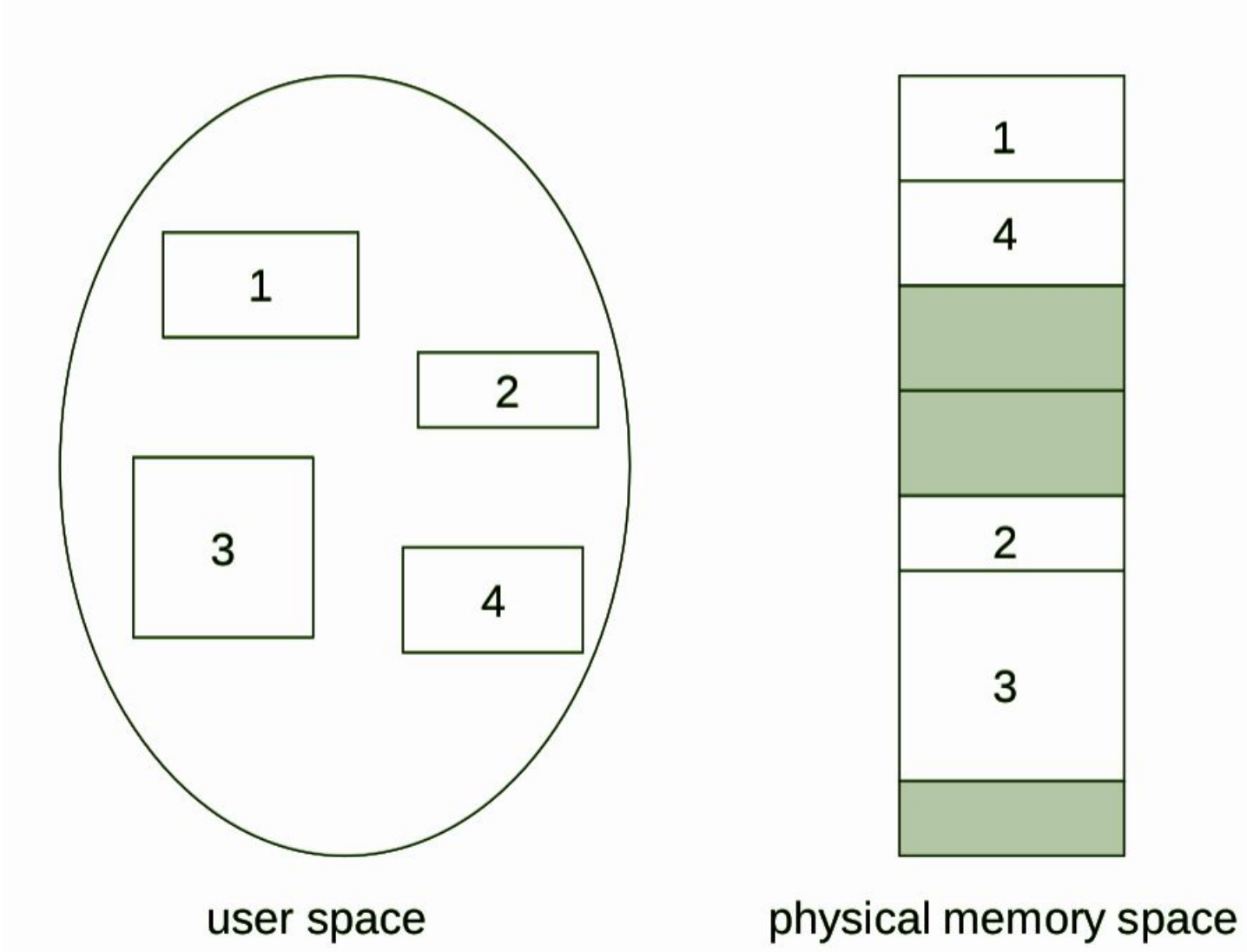
  - **Virtual memory segmentation**
    Each processor job is divided into several segments, It is not essential all of which are resident at any one point in time.

  - **Simple segmentation**
    Each process is divided into many segments, and all segments are loaded into the memory at run time, but not necessarily contiguously.

User View of the Program



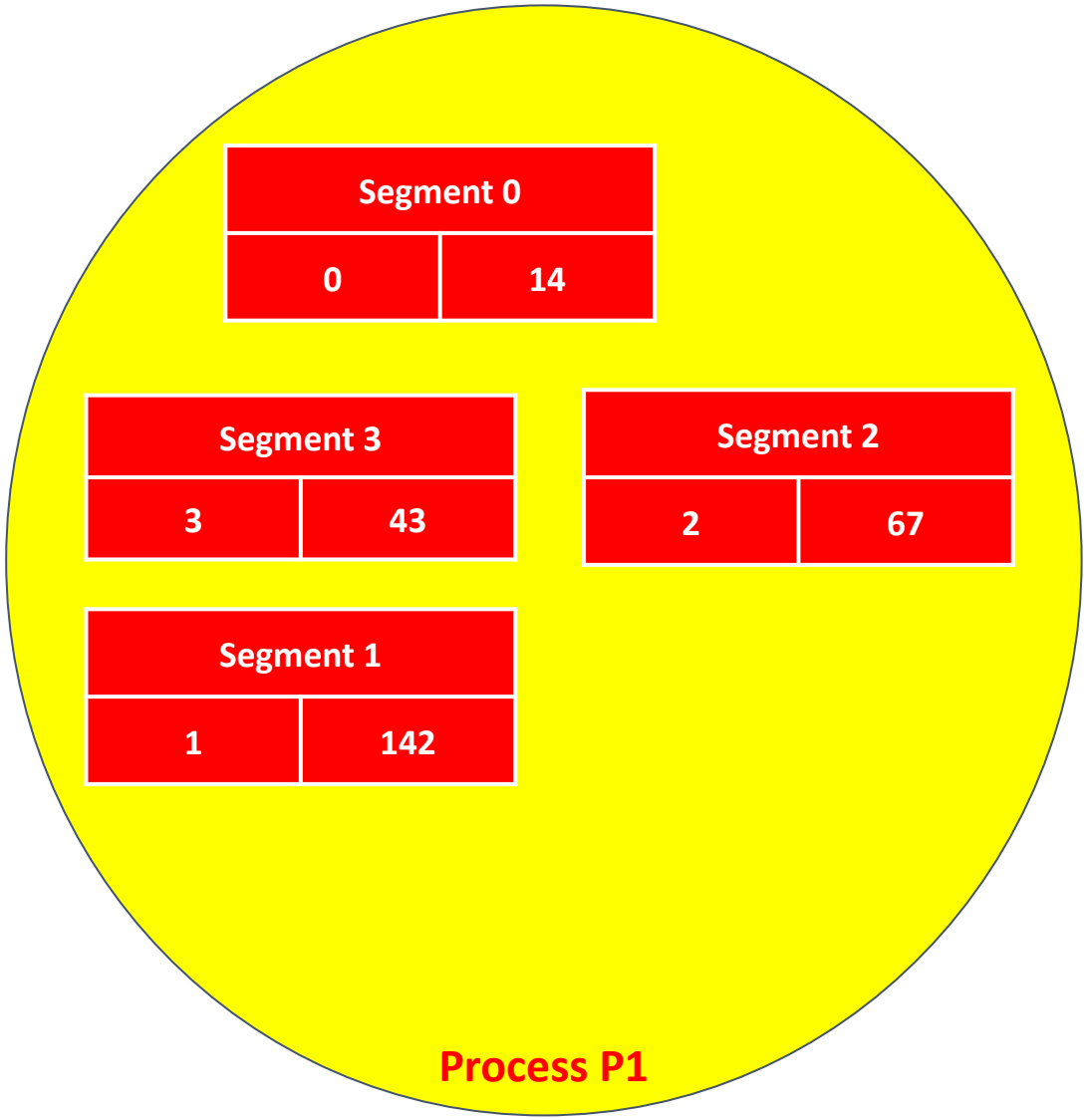
Logical View of Segmentation



Segmentation Architecture

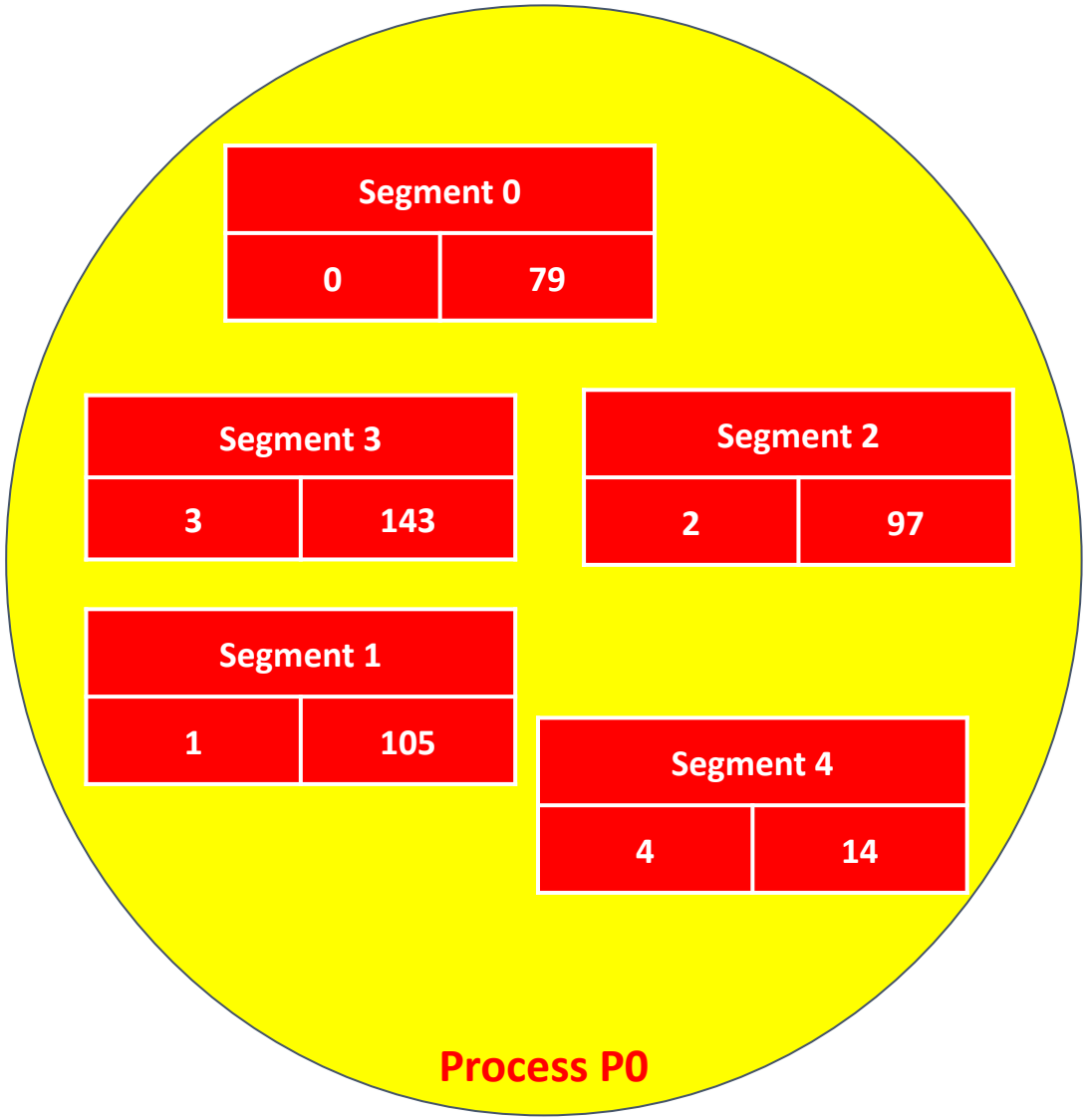
- Logical address consists of a two tuple:
  - <segment-number, offset>
- Segment table => maps two-dimensional physical addresses
- Each segment table entry has:
  - base => contains the starting physical address where the segments reside in memory
  - limit => specifies the length of the segment

Segmentation Example 1



Segment #	Base	Limit
0	Undefined	14
1	Undefined	142
2	Undefined	67
3	Undefined	43
Total=>4	Undefined	Total=>266

Segmentation Example - 2



Segment #	Base	Limit
0	Undefined	79
1	Undefined	105
2	Undefined	97
3	Undefined	143
4	Undefined	14
Total=>5	Undefined	Total=>438

Segmentation Architecture

- 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

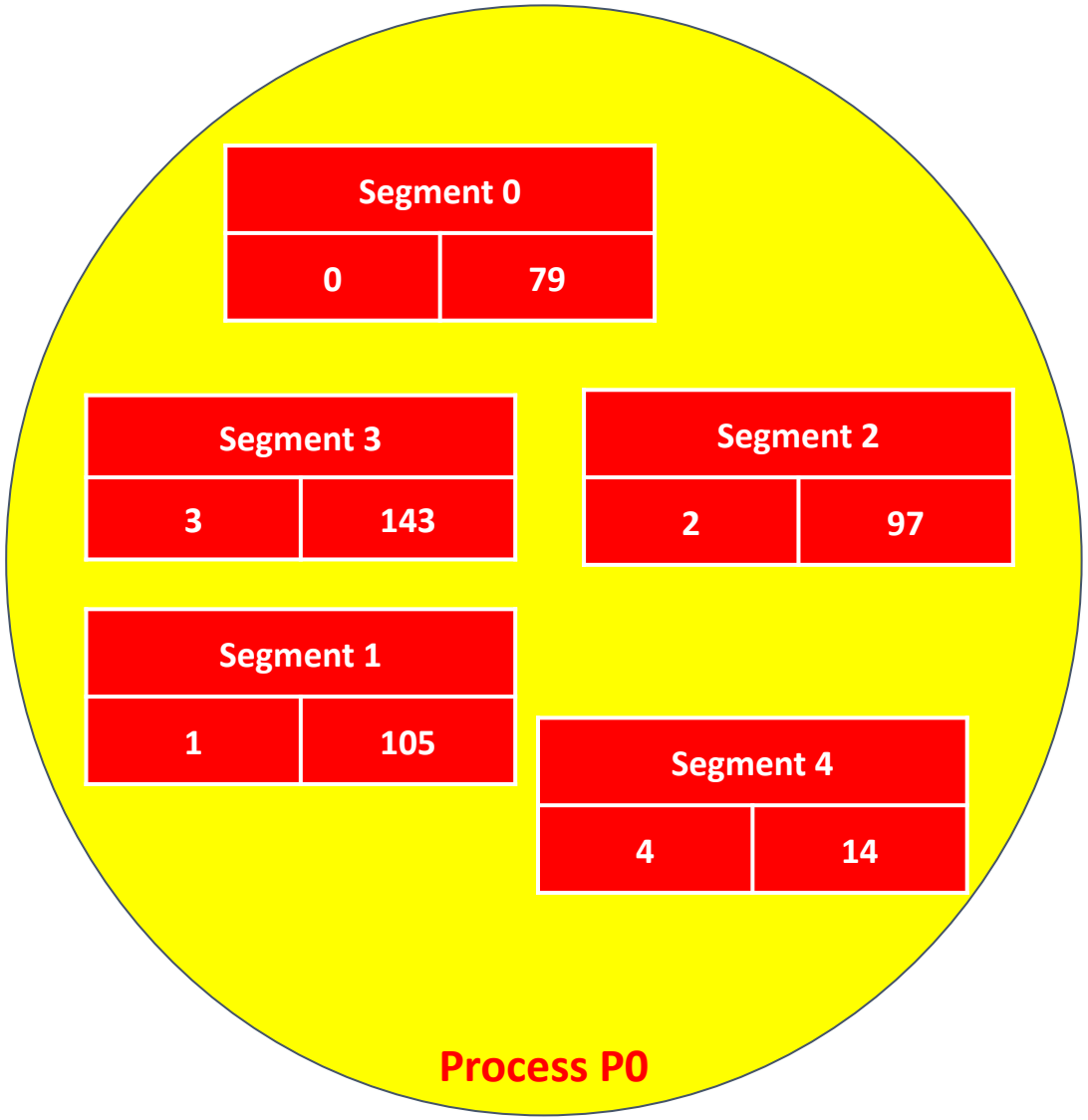
Segmentation Architecture

- 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

Segmentation Example - 2

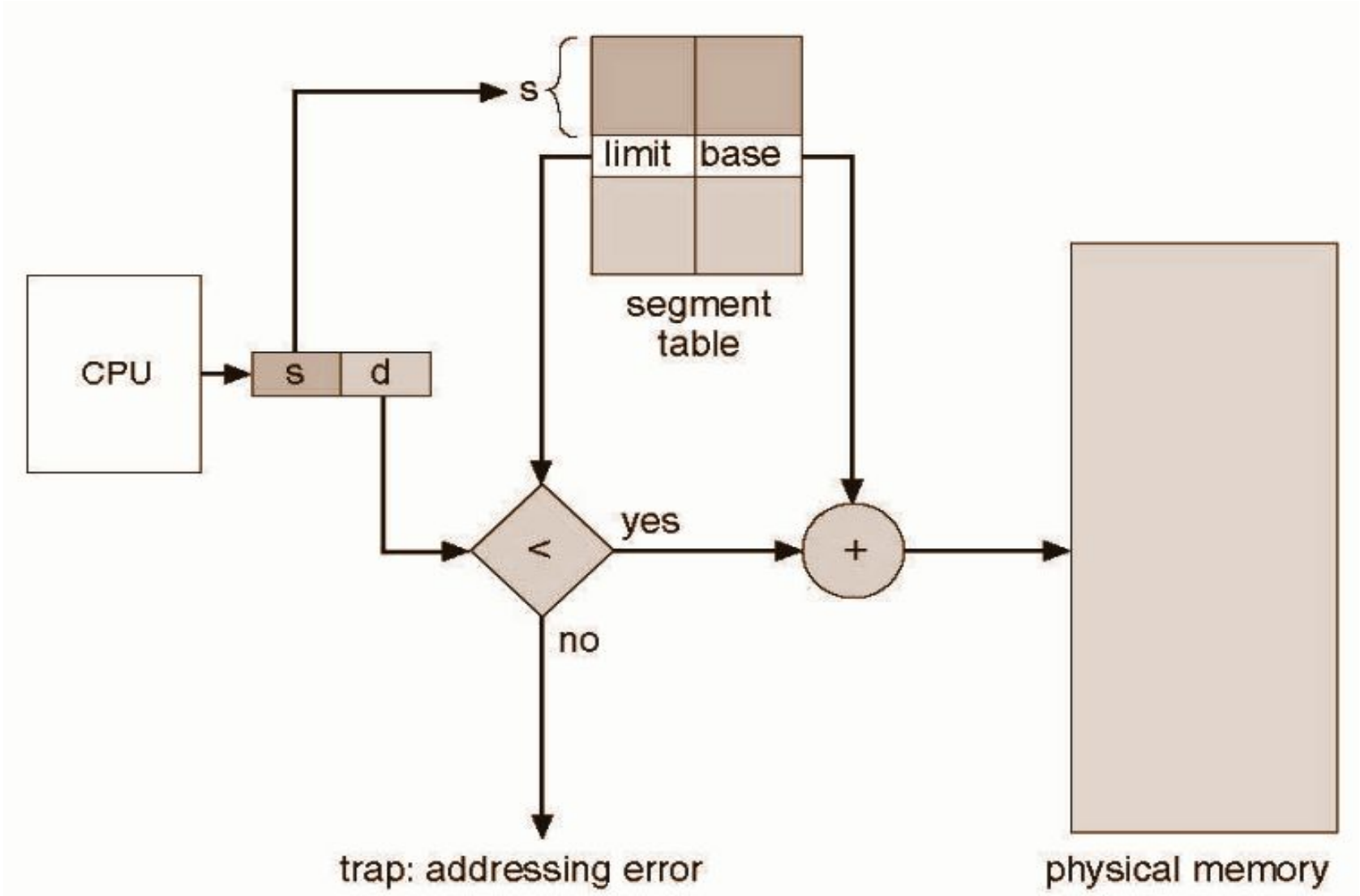


Segment Table		
Segment #	Base	Limit
0	Undefined	79
1	Undefined	105
2	Undefined	97
3	Undefined	143
4	Undefined	14
Total=>5	Undefined	Total=>438

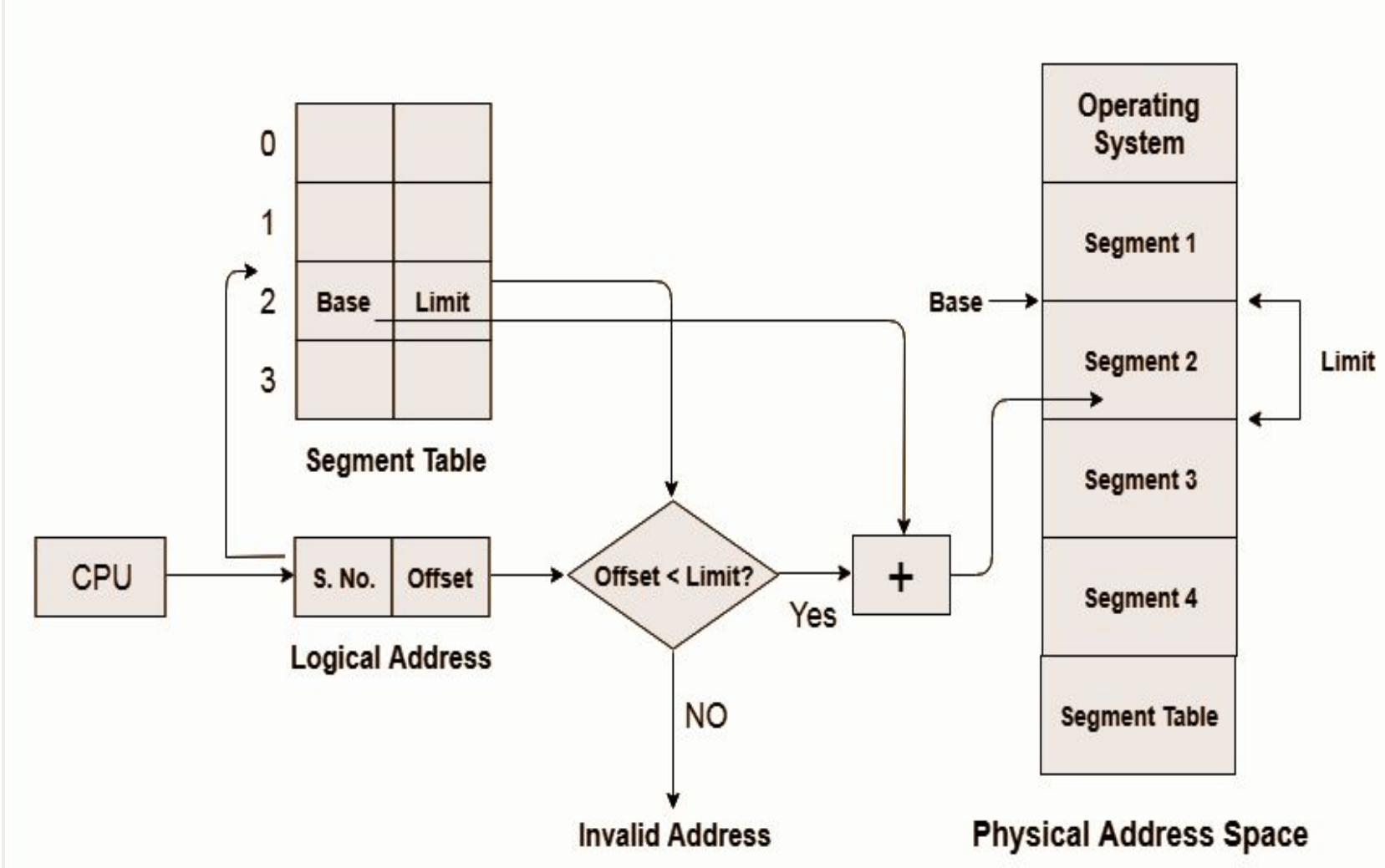
Segment Table Base Register - STBR
Undefined

Segment Table Length Register - STLR
5
Legal range of STLR => {0,1,2,3,4}

Segmentation Hardware



Segmentation Hardware - Additional Input



Segmentation Example - 2

Segment Table		
Segment #	Base	Limit
0	Undefined	79
1	Undefined	105
2	Undefined	97
3	Undefined	143
4	Undefined	14
Total=>5	Undefined	Total=>438

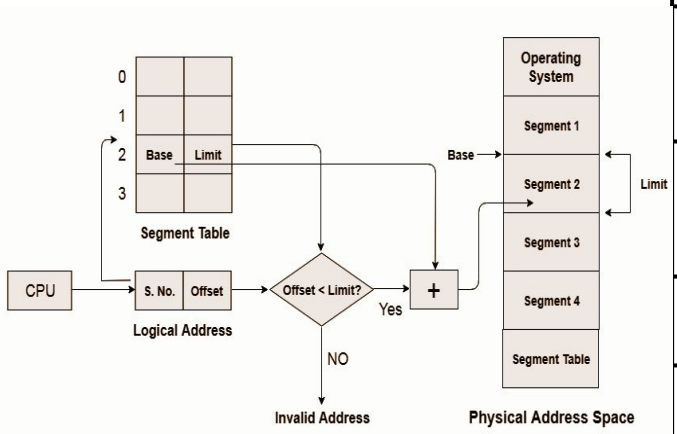
Segment Table		
Segment #	Limit	Base
0	79	4000
1	105	5012
2	97	1024
3	143	1197
4	14	9014

Segment Table Base Register - STBR
CPU / OS

Segment Table Length Register - STLR => CPU / OS
5
Legal range of STLR => {0,1,2,3,4}

Segmentation Example - 2

Segment Table		
Segment #	Limit	Base
0	79	4000
1	105	5012
2	97	1024
3	143	1197
4	14	9014



Segment Table Base Register - STBR
CPU / OS

Segment Table Length Register - STLR => CPU / OS
5
Legal range of STLR => {0,1,2,3,4}

Physical Memory	
Absolute Address	Content
0 to 1023	OS
1024 to 1120	Segment #2
1197 to 1339	Segment #3
4000 to 4078	Segment #0
5012 to 5116	Segment #1
9014 to 9027	Segment #4
9028	Other
10098	Other

Segmentation Example - 2

Segment Table		
Segment #	Limit	Base
0	79	4000
1	105	5012
2	97	1024
3	143	1197
4	14	9014

Segment Table Base Register - STBR
CPU / OS

Segment Table Length Register - STLR => CPU / OS
5
Legal range of STLR => {0,1,2,3,4}

Physical Memory	
Absolute Address	Content
0 to 1023	OS
1024 to 1120	Segment #2
1197 to 1339	Segment #3
4000 to 4078	Segment #0
5012 to 5116	Segment #1
9014 to 9027	Segment #4
9028	Other
10098	Other

Legal Address Range (inclusive)
0=>{ 4000 .. 4078 }
1=>{ 5012 .. 5116 }
2=>{ 1024 .. 1120 }
3=>{ 1197 .. 1339 }
4=>{ 9014 .. 9027 }

Are the following addresses Legal w.r.t Segment Table ?
(0, 4012) => Yes
(4, 9116) => No
(2, 1119) => Yes
(1, 5150) => No
(3, 1317) => Yes

Segmentation

Segmentation Advantages

- No internal fragmentation.
- Average segment sizes are larger, which allows segments to store more process data.
- Less processing overhead.
- Simpler to relocate segments than to relocate contiguous address spaces on disk.
- Segment tables are smaller than their counterpart tables, and takes up less memory.

Segmentation Disadvantages

- Uses legacy technology in x86-64 servers.
- Linux only supports segmentation in 80x86 microprocessors
- Porting Linux to different architectures is problematic because of limited segmentation support.
- Requires programmer intervention.
- Subjected to external fragmentation.

Topic Uncovered in this Session

- Non Contiguous memory allocation
- Segmentation
- User View of the Program
- Logical View of Segmentation
- Segmentation Architecture
- Segmentation Hardware

THANK YOU

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com and complete reading assignments provided by the Anchor on Edmodo