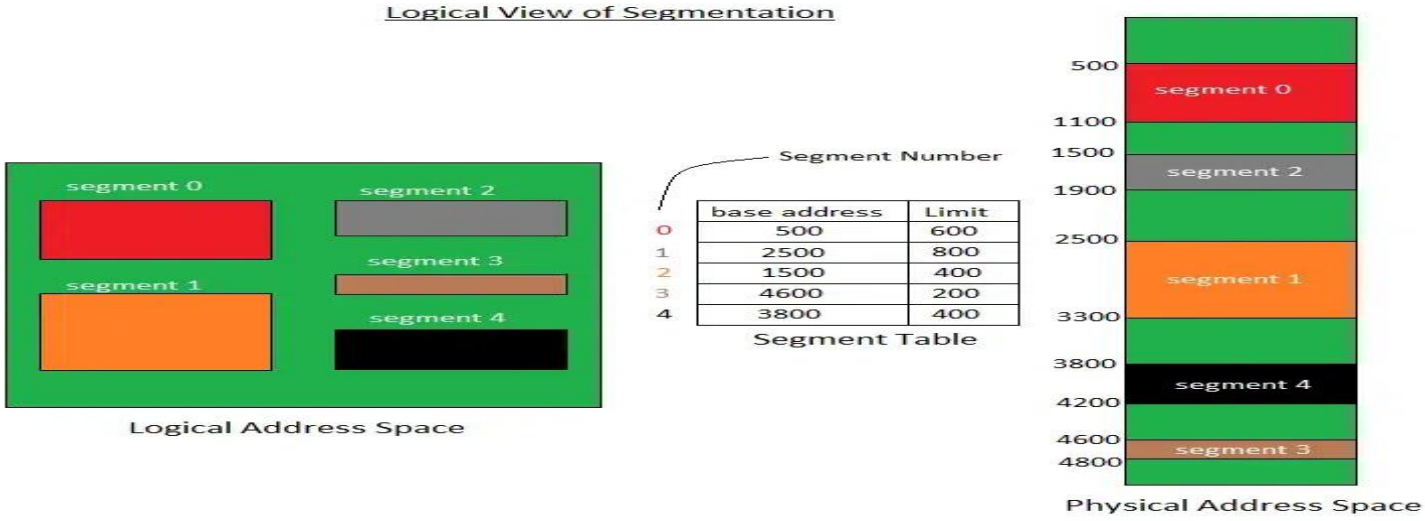
Lecture 5: Memory Management

Slow Learner Activity:-

1.Practice on Segmentation



(i) **Address Translation**

Given the logical address **(Segment 1, Offset 300)**:

1. What is the base address of Segment 1?
2. Is the offset valid? Justify your answer.
3. If valid, calculate the **Physical Address**.

→Base = 2500

Offset = 300 ≤ Limit = 800 → Valid

Physical Address = 2500 + 300 = 2800

(ii) **Labeling Practice**

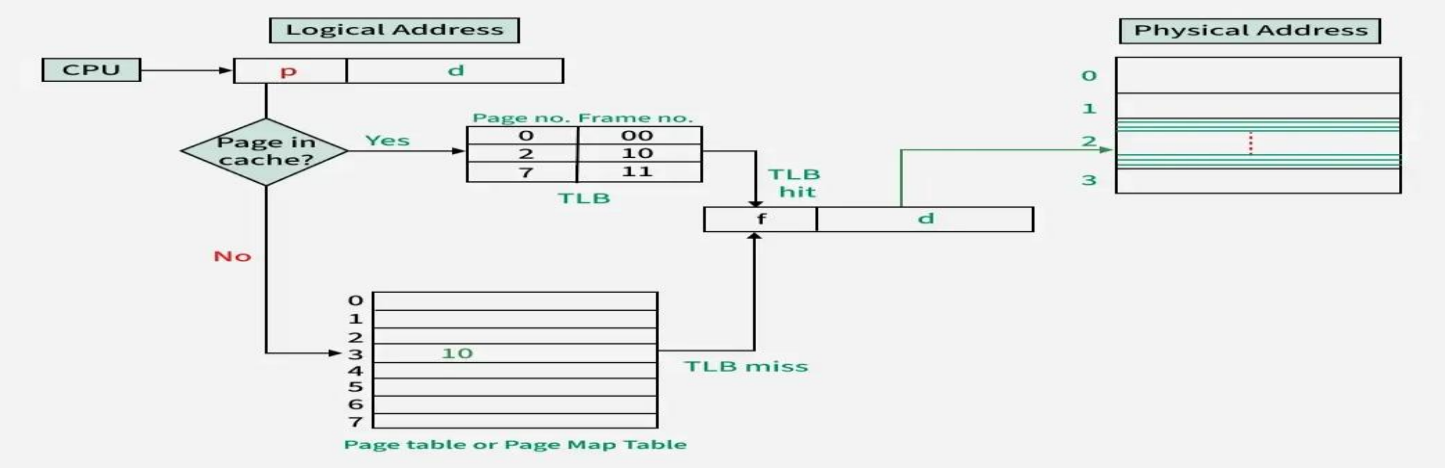
From the diagram:

- Label the **segment** that starts at physical address **3800**.
- Label the **segment** that has a **limit of 400** but starts at **1500**.

→ 3800 = Segment 4

1500 = Segment 2

2.



1.**(A) Logical Address = (Page Number = 2, Offset = d)**

1. **Is there a TLB hit or TLB miss?**
2. **If it's a hit, what is the corresponding frame number?**
3. **Write the final physical address (Frame number + Offset).**

(B) Logical Address = (Page Number = 3, Offset = d)

1. **Will the CPU find the page in the TLB?**
2. **If not, where does it look next?**
3. **From the page table, what is the frame number for Page 3?**

→1(A).Logical Address = (Page Number = 2, Offset = d)

a) **Is there a TLB hit or TLB miss?**
TLB hit — Page 2 is present in the TLB.

b) **Corresponding Frame Number:**
From the TLB, Page 2 maps to **Frame 10**.

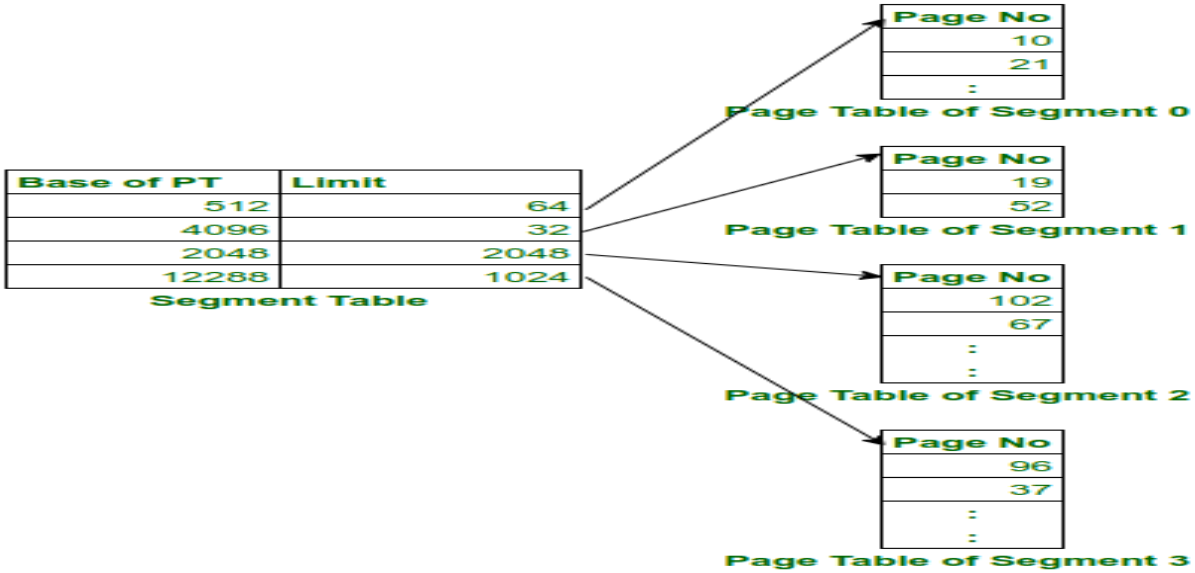
c) **Final Physical Address:**
Combine **Frame 10** with **Offset d**
Physical Address = (10, d)

→1.(B) Logical Address = (Page Number = 3, Offset = d)

1. **Will the CPU find the page in the TLB?**
No — Page 3 is not present in the TLB.
2. **Where does it look next?**
It looks in the Page Table (Page Map Table).
3. **Frame number for Page 3 from Page Table:**
Page Table shows: Page 3 → Frame 10

So, Physical Address = (10, d) again.

3.Two-Level Address Translation – Segmentation with Paging



A system uses segmentation with paging. The Segment Table contains the base addresses and limits for each segment's page table. The Page Table for each Segment maps logical page numbers to physical page frames.

Now, consider a logical address given as a triple:
(Segment Number = 1, Page Number = 1, Offset = 20)

Using the information from the diagram:

Answer the following:

(a) Is the given logical address valid? Justify your answer using the limit value.
(b) If valid, calculate the physical address using the page table of Segment 1. Assume that each page has a size of 64 bytes.

→Given:

- **Logical Address = (Segment Number = 1, Page Number = 1, Offset = 20)**
- **Page Size = 64 bytes**

Step-by-step Analysis:

1. Check if Segment Number 1 is valid:

From the Segment Table:

Segment	Base of PT	Limit
1	4096	32

So, the Page Number must be less than the Limit:
Page Number = 1 → Valid (since 1 < 32)

2. Get the Base Address of Segment 1's Page Table:

- **Base of Page Table for Segment 1 = 4096**

From the diagram:

- **Page Table of Segment 1 → maps:**

- **Page 0 → Frame 19**
- **Page 1 → Frame 52 (this is our required frame)**

3. Get Frame Number from Page Table:

- **Page Number = 1 → Frame Number = 52**

4. Calculate Physical Address:

We use the formula:
Physical Address = (Frame Number × Page Size) + Offset

- **Frame Number = 52**
- **Page Size = 64 bytes**
- **Offset = 20**

So,
Physical Address = (52 × 64) + 20 = 3328 + 20 = 3348

Final Answer:

- **Valid Address**
- **Physical Address = 3348**

Moderate Learner Activity:-

1.

Consider the virtual page reference string

$$1, 2, 3, 2, 4, 1, 3, 2, 4, 1$$

on a demand paged virtual memory system running on a computer system that has main memory size of 3 page frames which are initially empty. Let LRU, FIFO and $\mathrm{OPTIMAL}$ denote the number of page faults under the corresponding page replacement policy. Then

A. $\mathrm{OPTIMAL} < \mathrm{LRU} < \mathrm{FIFO}$
B. $\mathrm{OPTIMAL} < \mathrm{FIFO} < \mathrm{LRU}$
C. $\mathrm{OPTIMAL} = \mathrm{LRU}$
D. $\mathrm{OPTIMAL} = \mathrm{FIFO}$

→

First In First Out (FIFO):

The given virtual page reference string is 1, 2, 3, 2, 4, 1, 3, 2, 4, 1 and the main memory size is 3-page frames.

1	1	1	1	4	4	4	4	4	4
	2	2	2	2	1	1	1	1	1
		3	3	3	3	3	2	2	2
F	F	F	H	F	F	H	F	H	H

Total number of page faults= 6

Optimal Page replacement:

1	1	1	1	1	1	1	1	1	1
	2	2	2	**4**	4	4	4	4	4
		3	3	3	3	3	**2**	2	2
F	F	F	H	F	H	H	F	H	H

Total number of page faults= 5

Least Recently Used :

1	1	1	1	4	4	4	2	2	2
	2	2	2	2	2	3	3	3	1
		3	3	3	1	1	1	4	4
F	F	F	H	F	F	F	F	F	F

Total number of page faults= 9

The number of page faults with OPTIMAL, FIFO, LRU (5,6,9) is **OPTIMAL < FIFO < LRU.**

2. An operating system supports a paged virtual memory. The central processor has a cycle time of 1 microsecond. It costs an additional 1 microsecond to access a page other than the current one. Pages have 1,000 words, and the paging device is a drum that rotates at 3,000 revolutions per minute and transfers 1 million words per second. The following statistical measurements were obtained from the system:

One percent of all instructions executed accessed a page other than the current page.

•Of the instructions that accessed another page, 80 percent accessed a page already in memory.

• When a new page was required, the replaced page was modied 50 percent of the time.

Calculate the effective instruction time on this system, assuming that the system is running one process only and that the processor is idle during drum transfers.

→

effective access time = 0.99 × (1 µsec + 0.008 × (2 µsec) + 0.002 × (10,000 µsec + 1,000 µsec) + 0.001 × (10,000 µsec + 1,000 µsec) = (0.99 + 0.016 + 22.0 + 11.0) µsec = 34.0 µse

3.

Consider the following page-replacement algorithms. Rank these algorithms on a five-point scale from "bad" to "perfect" according to their page-fault rate. Separate those algorithms that suffer from Belady's anomaly from those that do not.

a. LRU replacement
b. FIFO replacement
c. Optimal replacement
d. Second-chance replacement

→

Rank	Algorithm	Suffer from Belady's anomaly
1	Optimal	no
2	LRU	no
3	Second-chance	yes
4	FIFO	yes

Fast Learner Activity:-

1. You are simulating a program with the following memory reference pattern:
1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

- TLB can store 3 entries (LRU replacement)
- RAM has 3 page frames
- Page replacement follows FIFO policy

Question:
Calculate the number of page faults and TLB misses. Analyze how different TLB and page replacement strategies (e.g., LRU, Optimal) would change the outcome. Suggest the best configuration for minimizing both.
→ **Given:**

- Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5
- TLB size: 3 (LRU)
- Page frames: 3 (FIFO)

Page Faults (FIFO):
Every time a new page not in memory is accessed:

- Total page faults = **12**
  (All pages are replaced too soon due to FIFO)

TLB Misses (LRU):

- First 3 accesses are misses.
- Repeated pages are evicted due to limited size.
- Approx. **10–12 TLB misses**

Better Configurations:

- **Use LRU or Optimal replacement** → Fewer page faults (Optimal gives **8**).
- **Increase TLB size** to reduce TLB misses.
- Apply **Working Set Model** to avoid unnecessary page replacements.
  2.

In the IBM/370, memory protection is provided through the use of *keys*. A key is a 4-bit quantity. Each 2K block of memory has a key (the storage key) associated with it. The CPU also has a key (the protection key) associated with it. A store operation is allowed only if both keys are equal, or if either is zero. Which of the following memory-management schemes could be used successfully with this hardware?

a. Bare machine
b. Single-user system
c. Multiprogramming with a fixed number of processes
d. Multiprogramming with a variable number of processes
e. Paging
f. Segmentation

Answer:

a. Protection not necessary, set system key to 0.
b. Set system key to 0 when in supervisor mode.
c. Region sizes must be fixed in increments of 2k bytes, allocate key with memory blocks.
d. Same as above.
e. Frame sizes must be in increments of 2k bytes, allocate key with pages.
f. Segment sizes must be in increments of 2k bytes, allocate key with segments.

3.Why are page sizes always powers of 2?

Answer: Recall that paging is implemented by breaking up an address into a page and offset number. It is most efficient to break the address into X page bits and Y offset bits, rather than perform arithmetic on the address to calculate the page number and offset. Because each bit position represents a power of 2, splitting an address between bits results in a page size that is a power of 2.

4. Consider a logical address space of 64 pages of 1024 words each, mapped onto a physical memory of 32 frames.

a. How many bits are there in the logical address?

b. How many bits are there in the physical address? Answer: a. Logical address: 16 bits b. Physical address: 15 bits