

FACULTY OF ENGINEERING AND TECHNOLOGY UNIVERSITY OF LUCKNOW LUCKNOW



Operating System AI-602

Dr. Zeeshan Ali Siddiqui
Assistant Professor
Deptt. of C.S.E.

PAGING CONCEPT

Paging Concept^{1/3}

- Segmentation permits the physical address space of a process to be *non-contiguous*.
- *Paging* is another memory-management scheme that offers this advantage.
- *Paging* avoids external fragmentation and the need for compaction, whereas *segmentation* does not.

Paging Concept^{2/3}

- *Paging* solves the problem of fitting memory chunks of varying sizes onto the backing store.
- Paging is implemented through *cooperation* between the operating system and the computer hardware.
- **Problem:** *Internal fragmentation*

Paging Concept^{3/3}

- *Physical address space* of a process can be non-contiguous.
- Divide physical memory into fixed-sized blocks called *frames* (size is power of 2, between 512 and 8,192 bytes).
- Divide logical memory into blocks of same size called *pages*.
- To *run* a program of size n pages, need to find n free frames and load program.
- Set up a *page table* to translate logical to physical addresses.

Paging Model Of Logical And Physical Memory

page 0
page 1
page 2
page 3

logical
memory

0	1
1	4
2	3
3	7

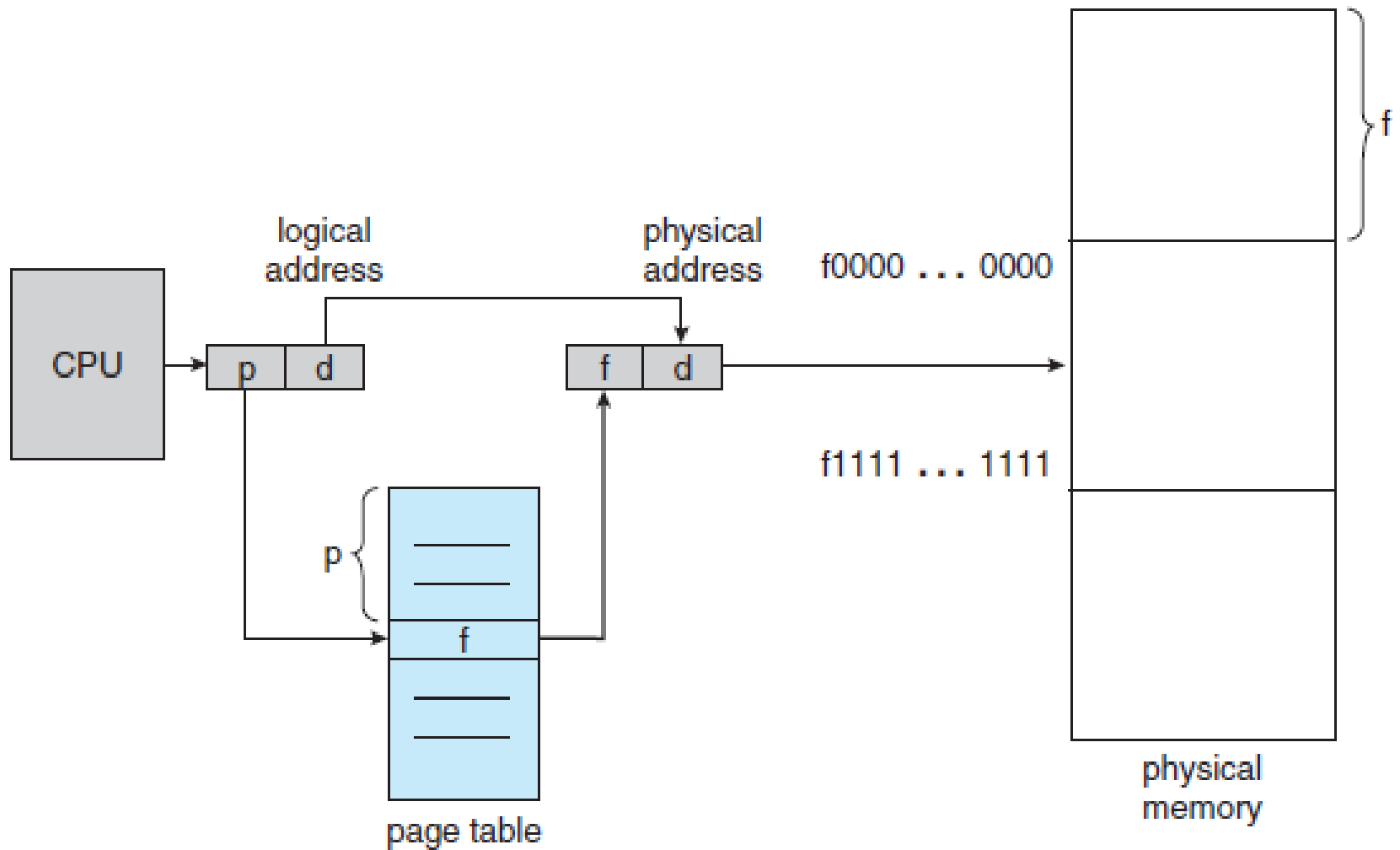
page table

frame
number

0	
1	page 0
2	
3	page 2
4	page 1
5	
6	
7	page 3

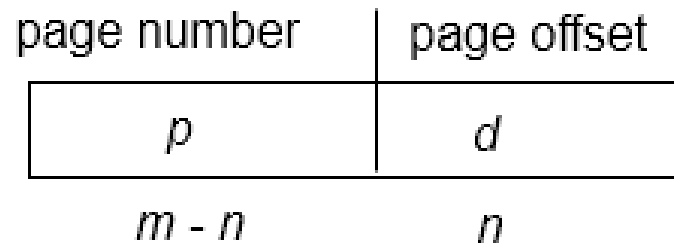
physical
memory

Paging Hardware



Address Translation Scheme

- Address generated by CPU is divided into:
 - *Page number (p)* – used as an index into a page table which contains base address of each page in physical memory.
 - *Page offset (d)* – combined with base address to define the physical memory address that is sent to the memory unit.



- Where p is an index into the page table and d is the *displacement* within the page.
 - For given logical address space 2^m and page size 2^n .

Example_{1/2}

- Let the logical address, $n = 2$ and $m = 4$.
- Using a
 - page size of 4 bytes and
 - a physical memory of 32 bytes (8 pages).
- Logical address 0 is page 0, offset 0.

Example^{2/2}

0	a
1	b
2	c
3	d
4	e
5	f
6	g
7	h
8	i
9	j
10	k
11	l
12	m
13	n
14	o
15	p

logical memory

0	5
1	6
2	1
3	2

page table

0	
4	i j k l
8	m n o p
12	
16	
20	a b c d
24	e f g h
28	

physical memory

Example: Analysis^{1/3}

- We find that page 0 is in frame 5.
- Thus, logical address 0 maps to
- physical address 20 [= (5 × 4) + 0].

0	a
1	b
2	c
3	d
4	e
5	f
6	g
7	h
8	i
9	j
10	k
11	l
12	m
13	n
14	o
15	p

logical memory

0	5
1	6
2	1
3	2

page table

0	
4	i j k l
8	m n o p
12	
16	
20	a b c d
24	e f g h
28	

physical memory

- Logical address 3 (page 0, offset 3) maps to physical address 23 [= (5 × 4) + 3].

Example: Analysis^{2/3}

0	a
1	b
2	c
3	d
4	e
5	f
6	g
7	h
8	i
9	j
10	k
11	l
12	m
13	n
14	o
15	p

logical memory

0	5
1	6
2	1
3	2

page table

0	
4	i j k l
8	m n o p
12	
16	
20	a b c d
24	e f g h
28	

physical memory

- Logical address 4 is page 1, offset 0; according to the page table, page 1 is mapped to frame 6. Thus, logical address 4 maps to physical address 24 [= (6 × 4) + 0].

Example: Analysis^{3/3}

0	a
1	b
2	c
3	d
4	e
5	f
6	g
7	h
8	i
9	j
10	k
11	l
12	m
13	n
14	o
15	p

logical memory

0	5
1	6
2	1
3	2

page table

0	
4	i j k l
8	m n o p
12	
16	
20	a b c d
24	e f g h
28	

physical memory

- Logical address 13 maps to physical address ?

References

1. Silberschatz, Galvin and Gagne, “Operating Systems Concepts”, Wiley.
2. William Stallings, “Operating Systems: Internals and Design Principles”, 6th Edition, Pearson Education.
3. D M Dhamdhere, “Operating Systems: A Concept based Approach”, 2nd Edition, TMH.

Thank You.

