Let's consider a memory system which has 5 bits

- # distinct addresses = 2^5 = 32 distinct addresses
- Each address is a 5-bit number
- First address starts at 00000
- The last address is 11111

Paging System

A paging system is a LOGICAL way of viewing memory addresses.

- Divide the address so that 3 bits are allocated to the page number
- => # distinct pages = 2^3 = 8 distinct pages
- The remaining 2 bits are allocated to the displacement
- => How many displacements are in each page? Answer: 2^2 = 4

Paging implementation is as follows

0	0	0	0	0
p	p	p	d	d

We now have two ways of referring to memory addresses:

1. Using “flat” address of all 5 bits of the address: 00100 for eg.
2. Describing the address as “PAGE Number X” + “DISPLACEMENT Number Y”

For example: Flat address 00101 = Page 001 + Displacement 01 as shown in turquiose in the diagram below...

LOGICAL ADDRESS (Programmer's view)			***PHYSICAL ADDRESS (System's view)***		
Page number	***displacement***	***Flat logical address***	***Page FRAME number***	***displacement***	***Flat Physical address***
000	00	00000	000	00	00000
	01	00001		01	00001
	10	00010		10	00010
	11	00011		11	00011
001	00	00100	001	00	00100
	01	00101		01	00101
	10	00110		10	00110
	11	00111		11	00111
010	00	01000	010	00	01000
	01	01001		01	01001
	10	01010		10	01010
	11	01011		11	1011

In this diagram, we indicate a translation of a logical address (shown in turquoise) to the physical address (shown in red). The translation is as follows:

- LOGICAL address 00101 translated to the PHYSICAL address 01001, or described another way:
- “PAGE Number 001 + DISPLACEMENT 01” translated to “PAGE ***FRAME*** Number 010 + DISPLACEMENT 01”

Our Page Table looks like:

Page number (indexes the table)	***Page FRAME number***
000	
001	010
010	
011	
...	