

Virtual Memory (H.1)

www.wikiversity.org

The necessities in Computer Organization

2015.06.09

Copyright (c) 2015 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Virtual Memory 가상 memory

Virtual Memory ----- O.S provides
↔ Physical Memory ----- M.M. (DRAM)

어떤 OS에서 exe file을 실행시키면

· process가 한개 이상 생성된다.

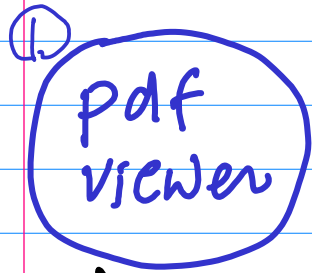
각 process가 쓰는 memory → Virtual memory

다른 process / 다른 user 다

시스템 자원을 공유 share. (time sharing)

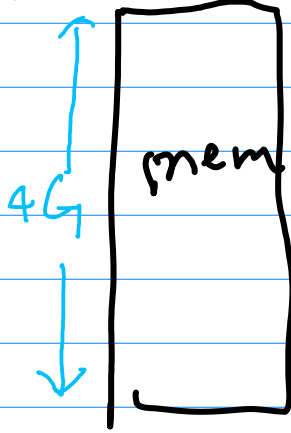
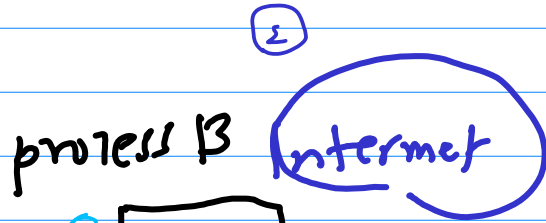
CPU, MEM, HDD, I/O

실용적인 응용 프로그램 2개가 3개의 example

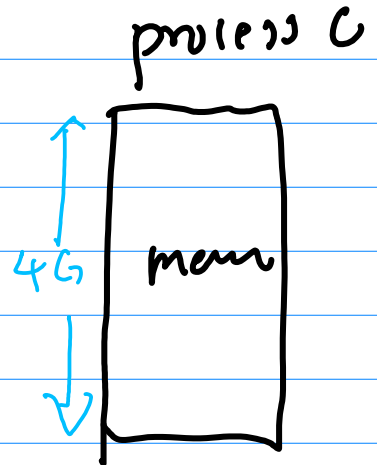
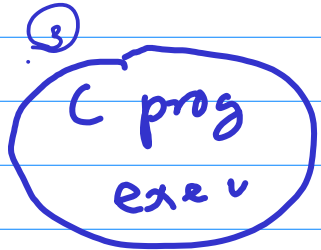


32-bit
addr
↓
4GB

$$2^{32} = \frac{2^2}{4} \cdot \frac{2^{30}}{G}$$

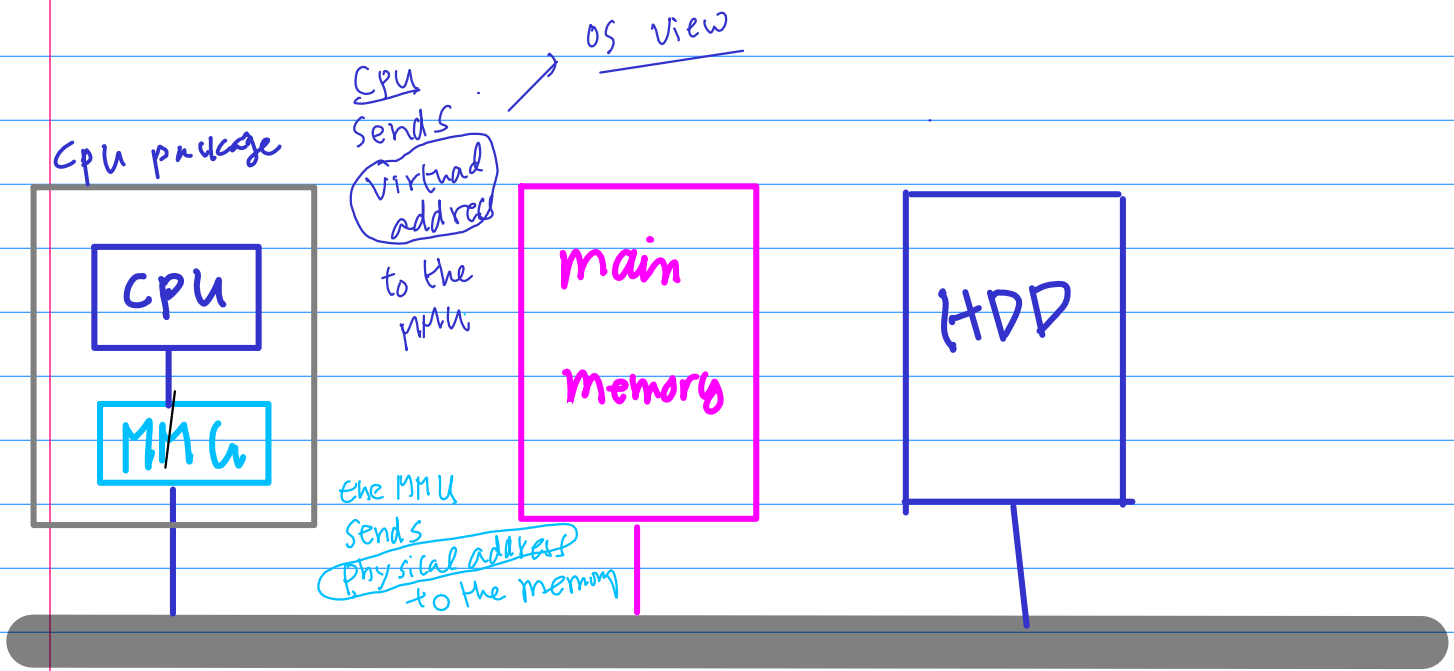


logical address



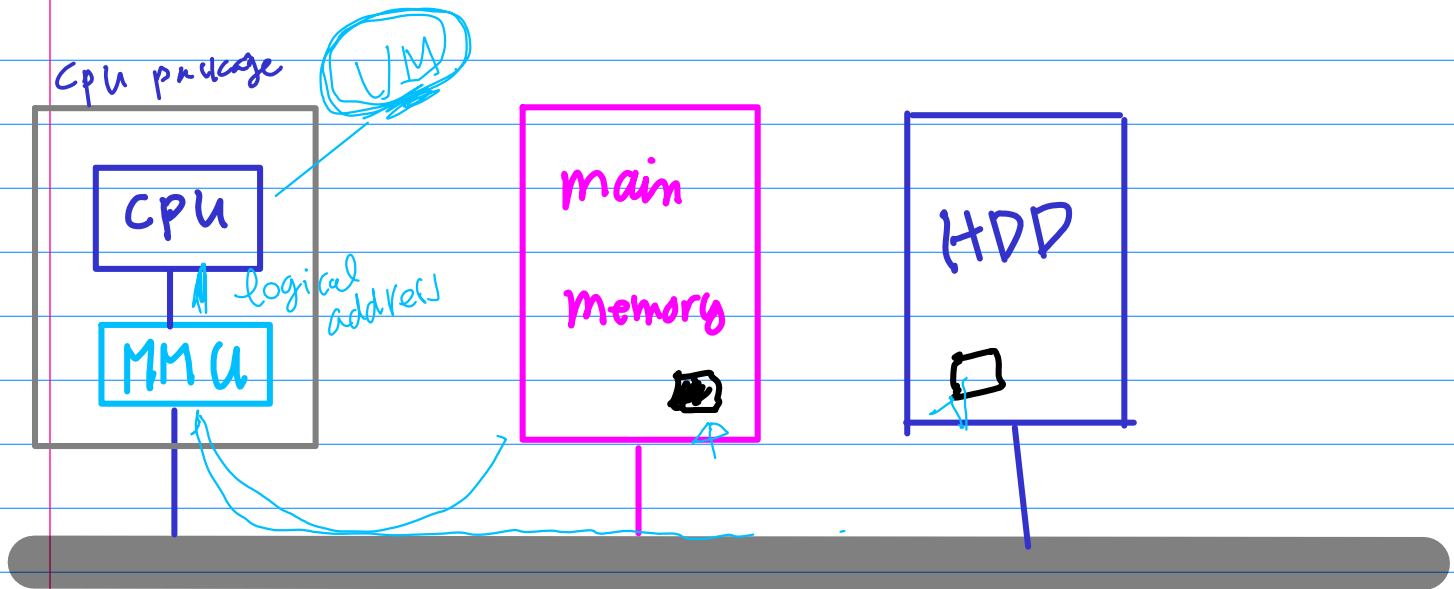
address
10 bit → 1K
20 bit → 1M
30 bit → 1G
32 bit → 4G

MMU, OS



OS keeps track of mapping V addr to P addr.

MMU translate V addr to P addr

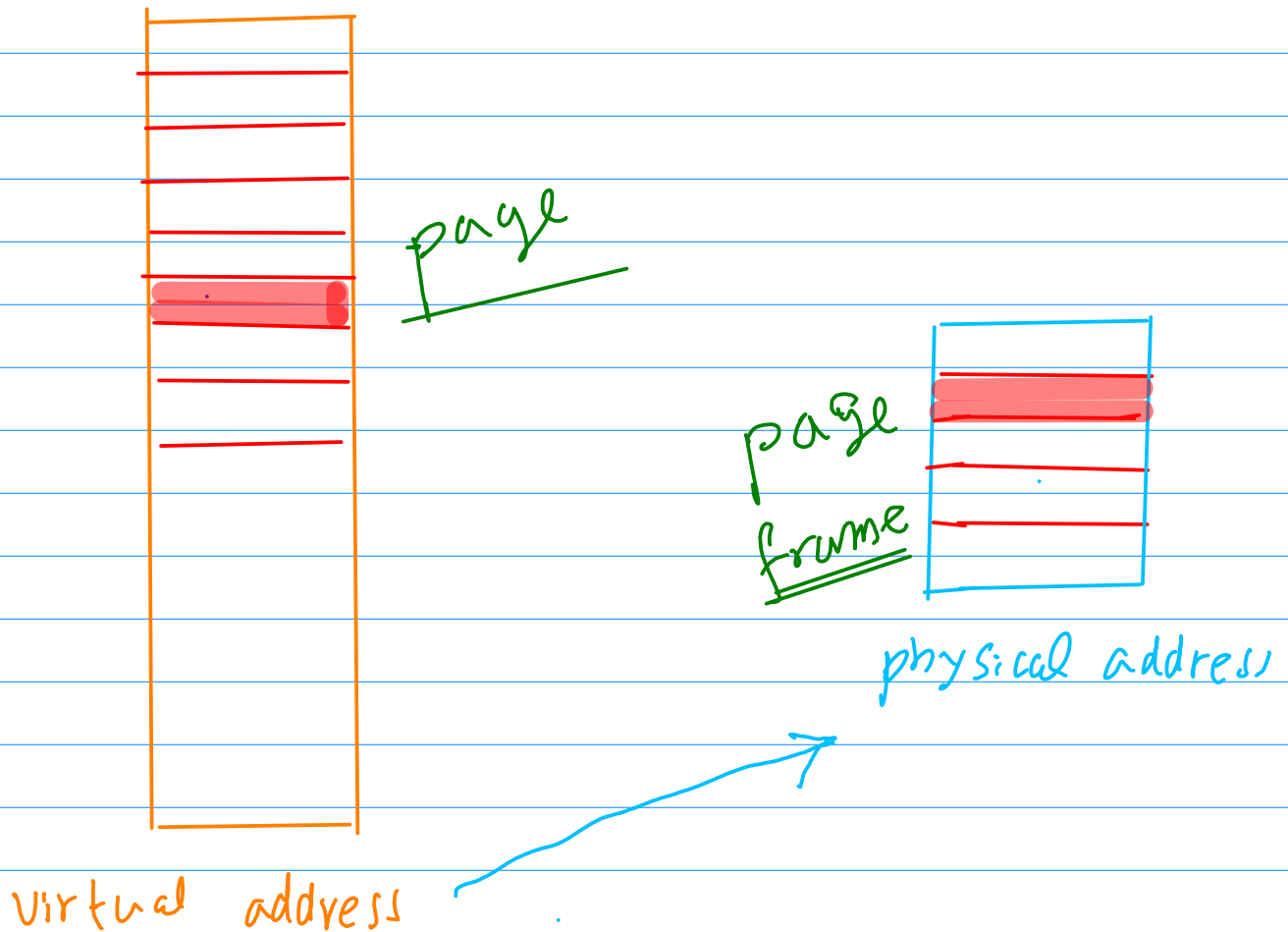


Page & page frame

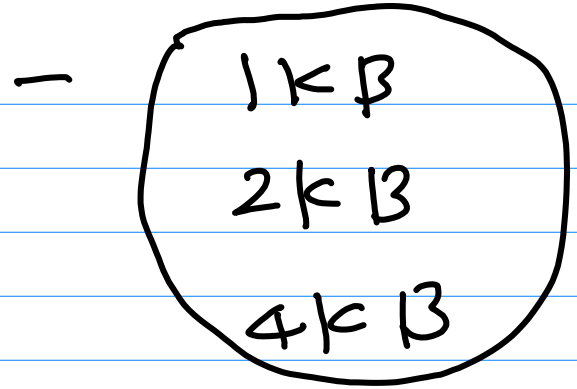
Fixed
size
memory
portion

Virtual Mem

Physical Mem

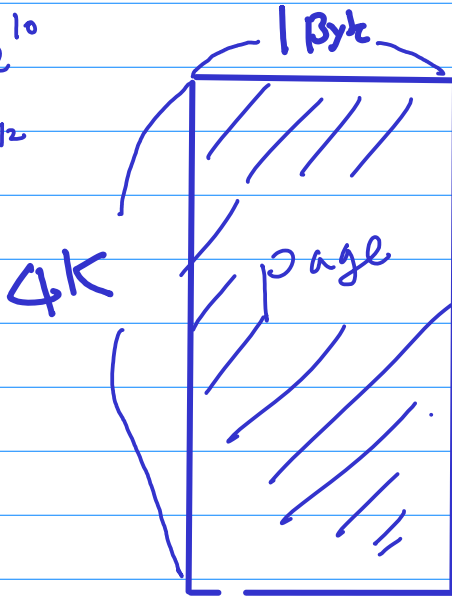


Page size



12-bit

$$2^2 \cdot 2^{10} \\ = 2^{12}$$



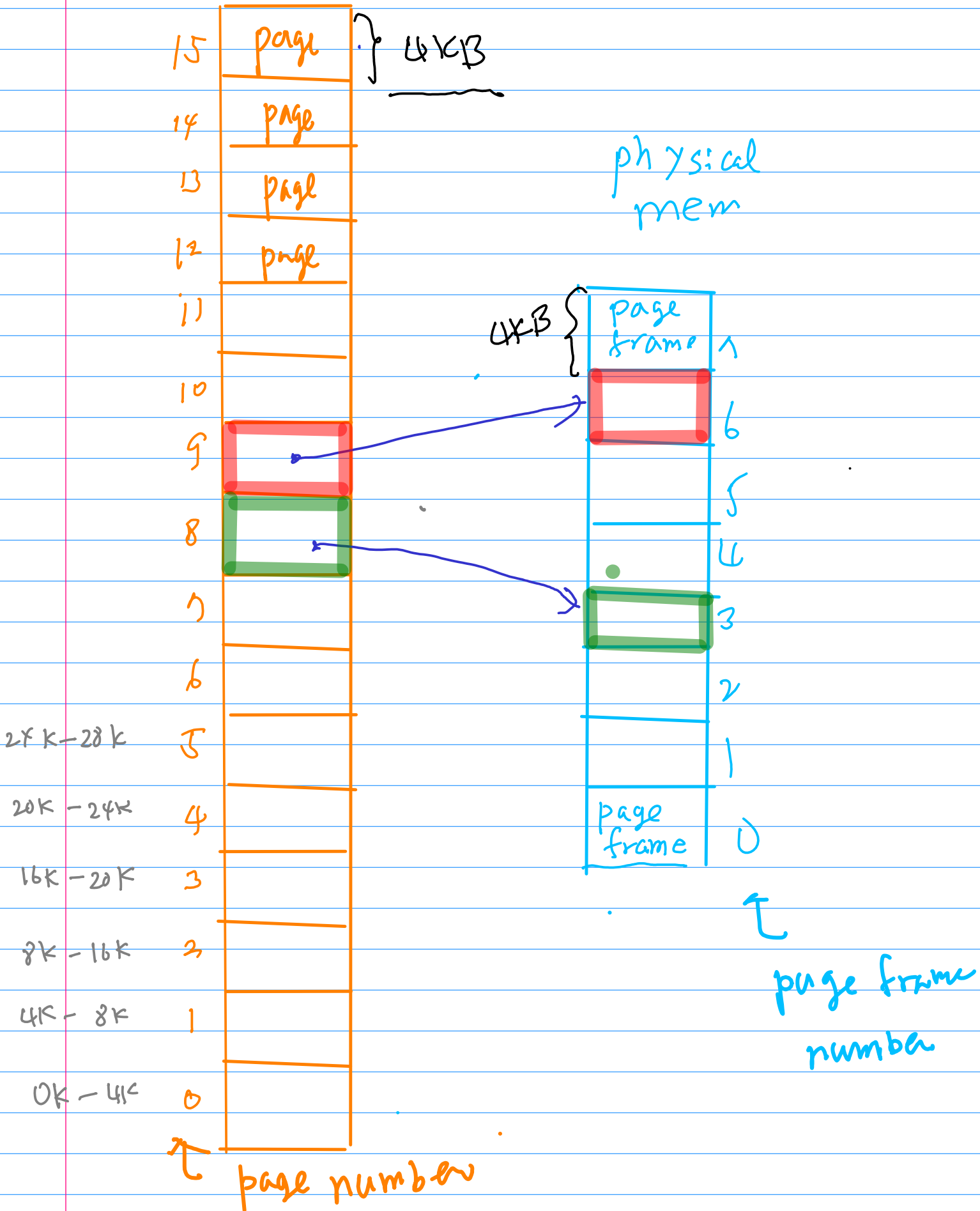
$$1K = 2^{10}$$

$$2K = 2^{11}$$

$$4K = 2^{12}$$

fixed size

Virtual Mem

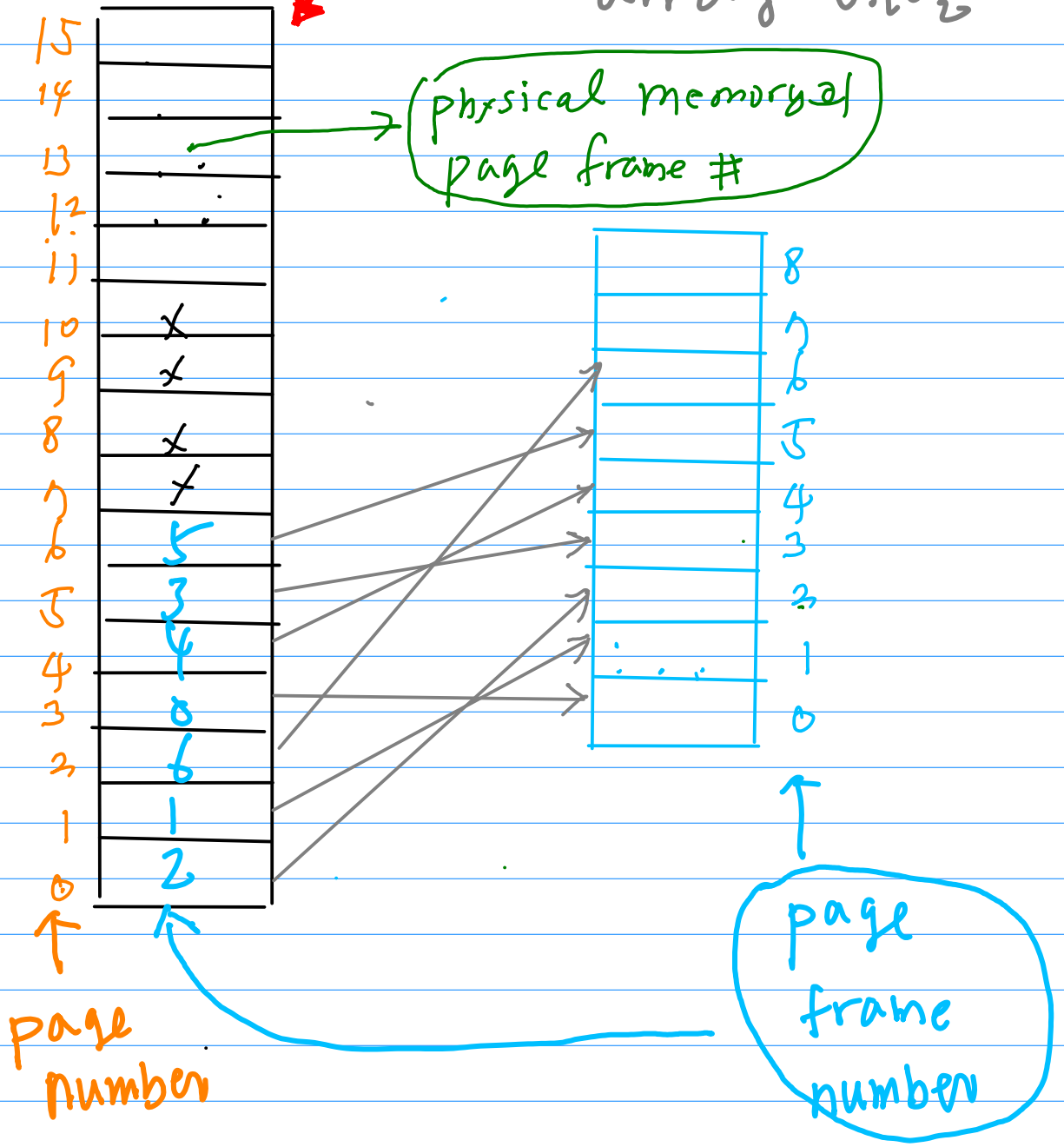


Page table

Θησζ

Virtual Mem

Array Θησζ



page table: virtual memory address와 physical memory address의 관계를 나타내는 테이블

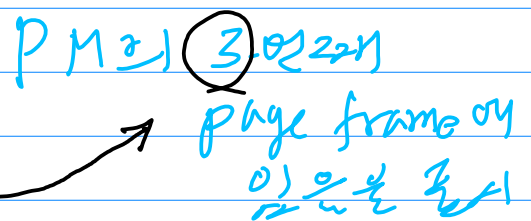
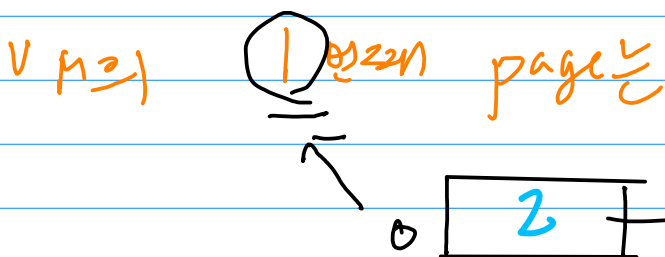
3	0
2	6
1	1
0	2

page table.
테이블 index

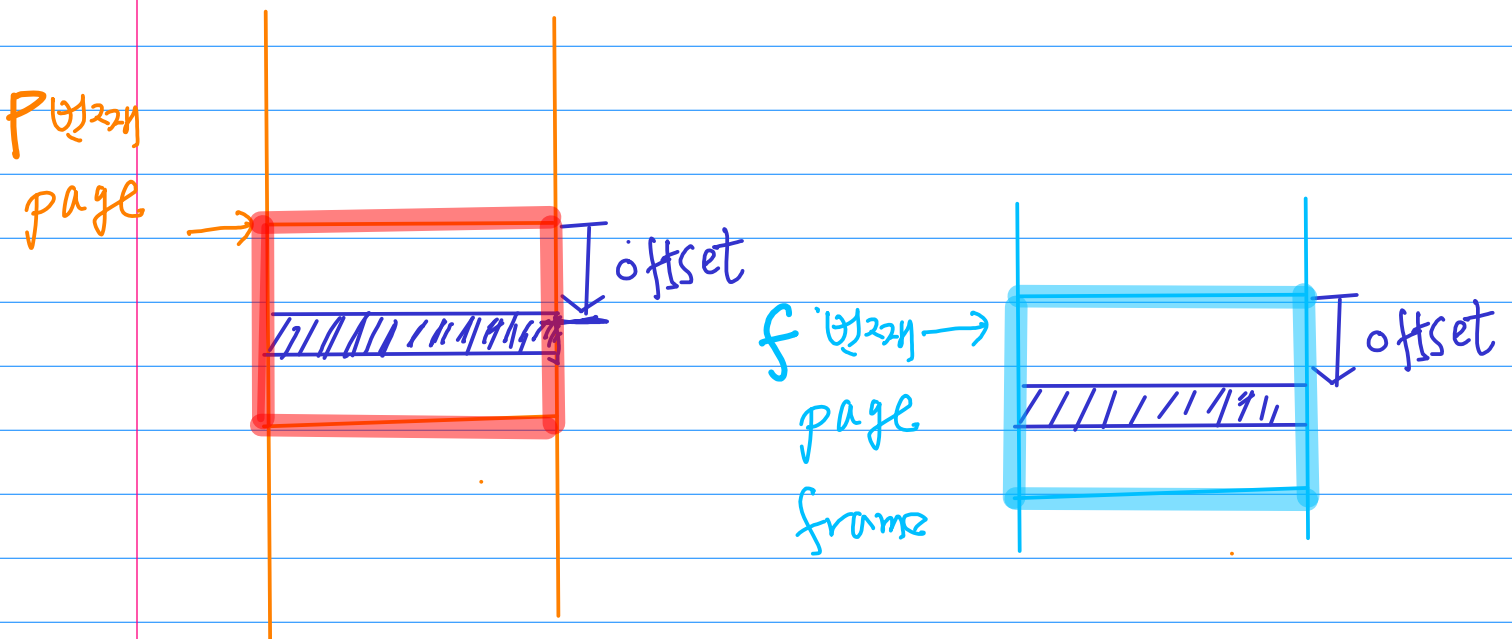
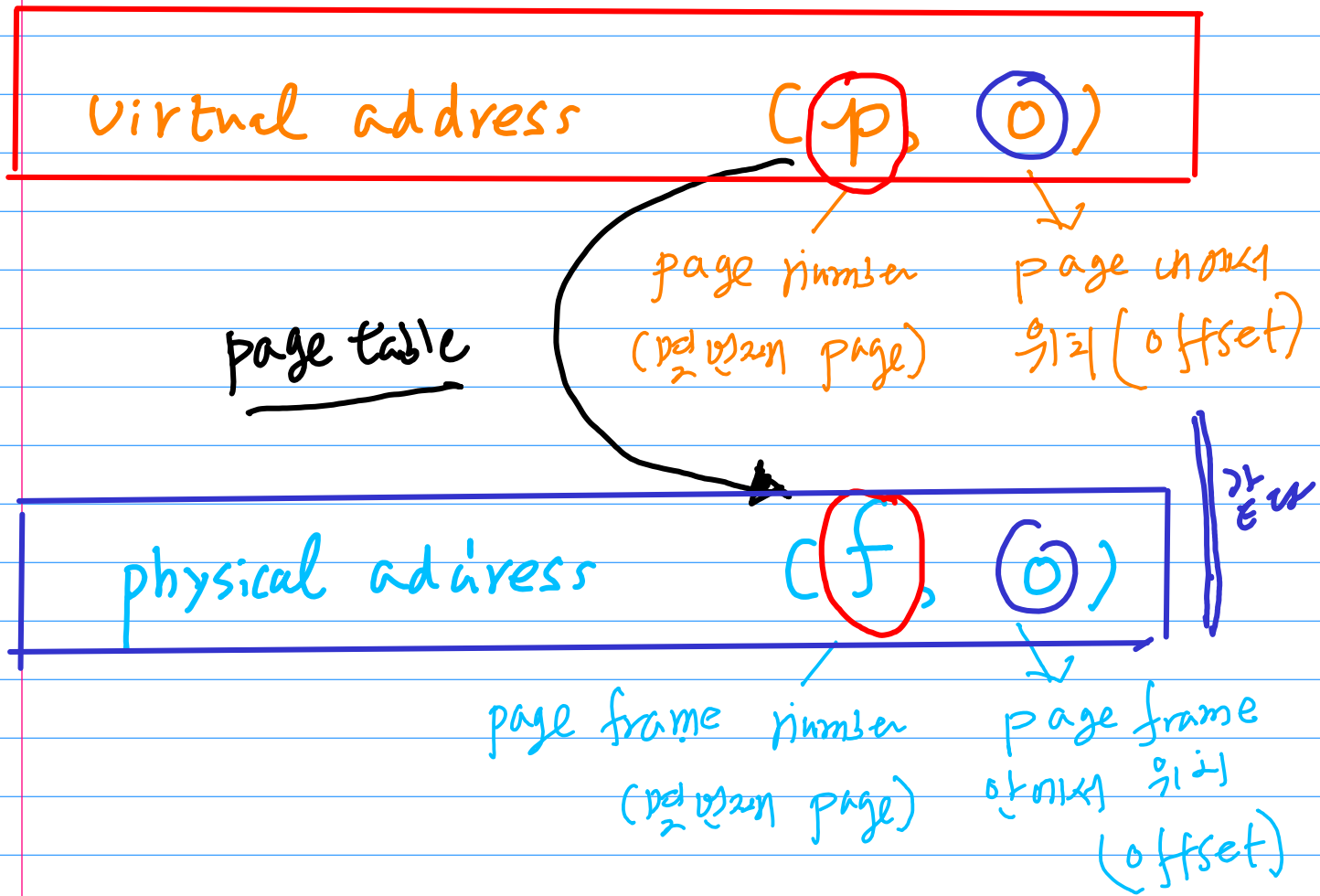
page table.
테이블 index에
존재하는 값

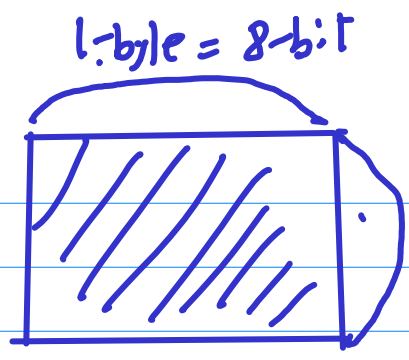
Virtual memory의
몇 번째 page인지를
나타낸다.

physical memory의
몇 번째 page frame
인지를 표시함

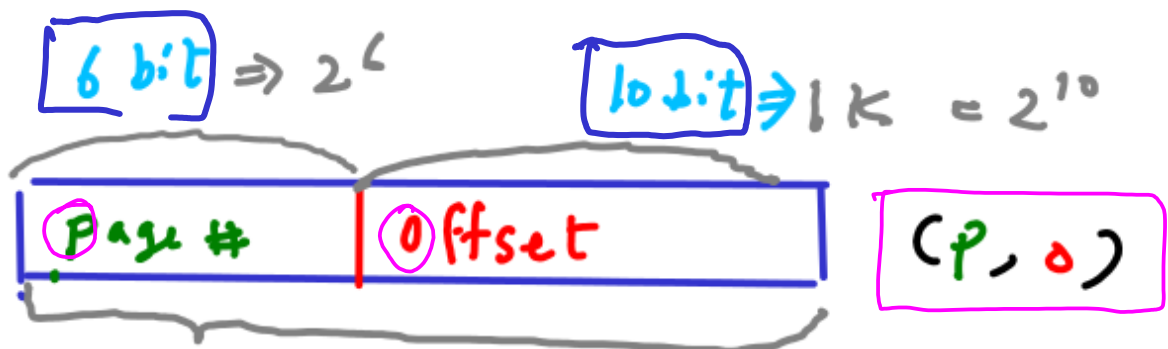


Paging

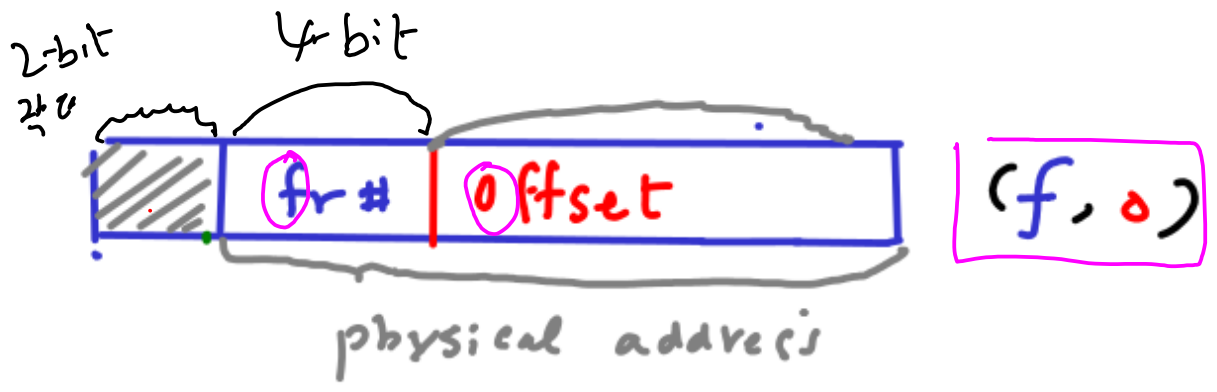




offset $\frac{1}{2}$ - 10-bit 2^{10}
 page size = $1K \cdot B$

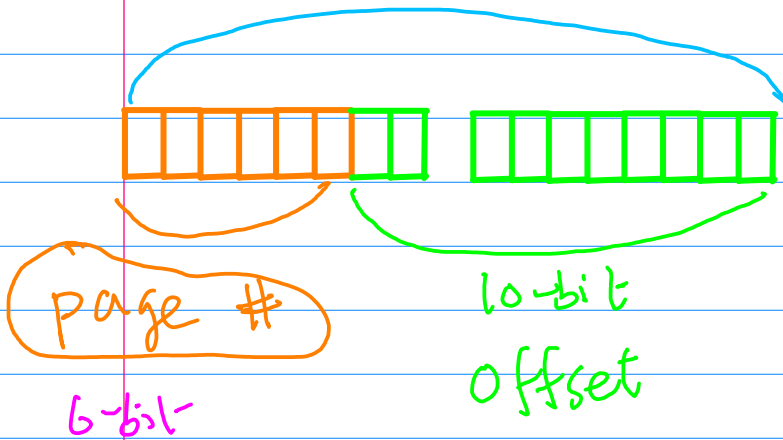


Virtual address = 16 bits = $2^6 K$



VM size \gg PM size
 16-bit address 14-bit address

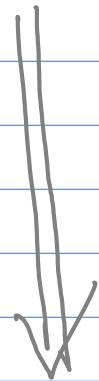
virtual memory
16-bit address



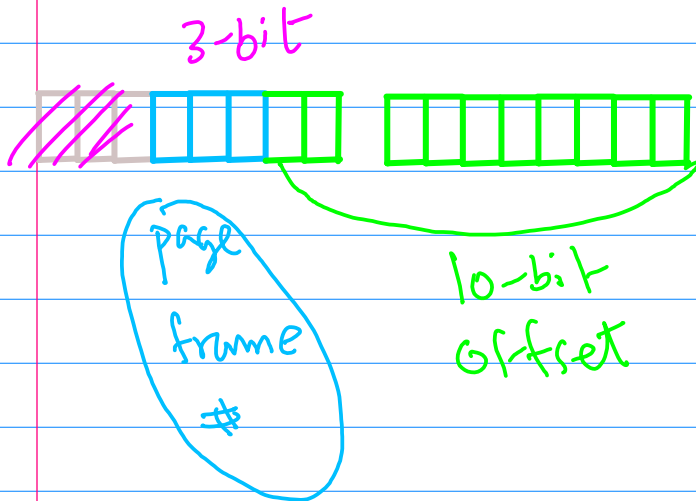
page size = 16 B \Rightarrow 10 bit

(P, 0)

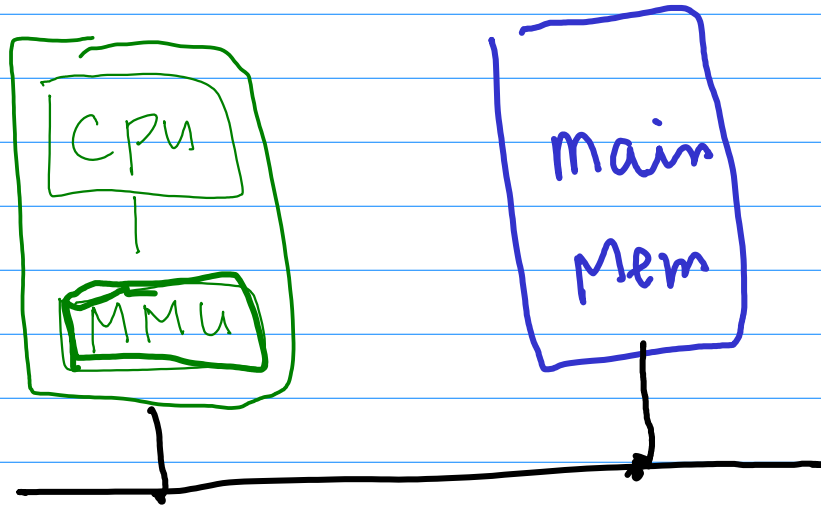
MMU
(Memory Management Unit)



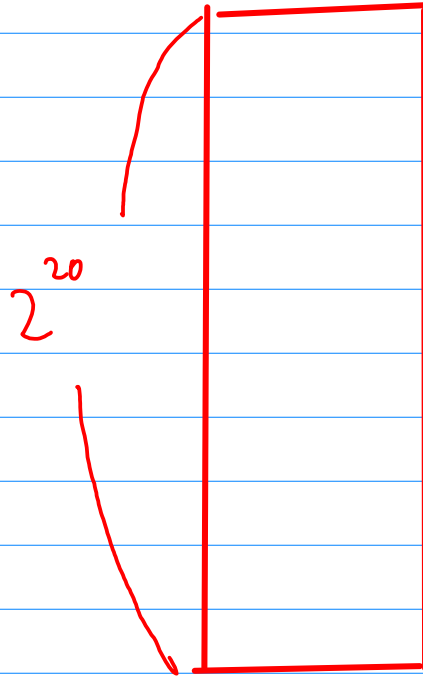
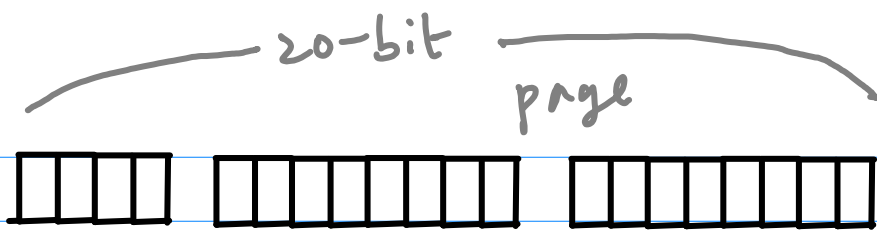
(F, 0)

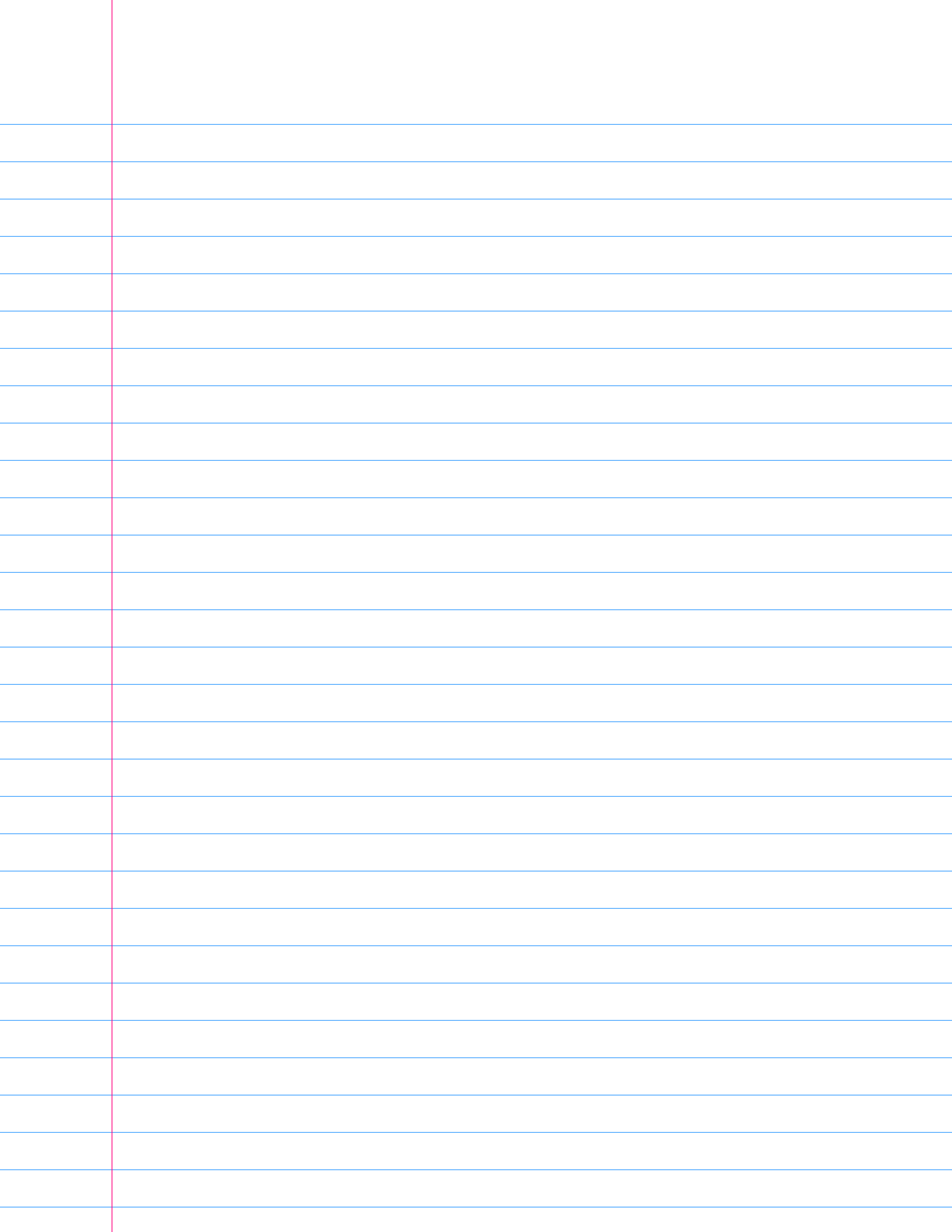


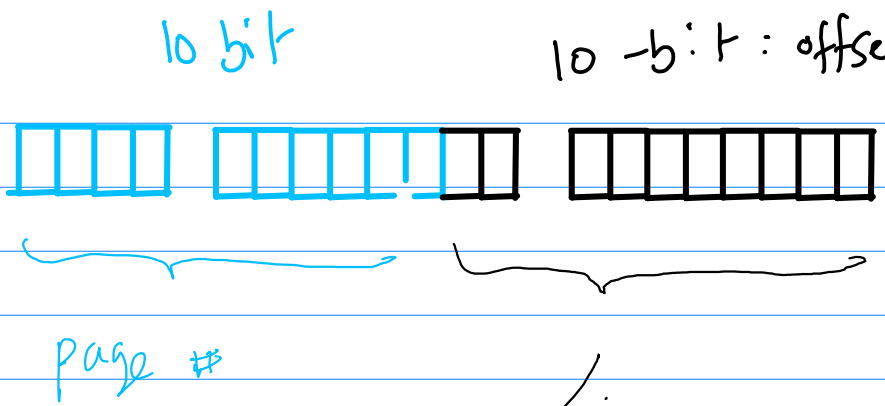
VM의 PM은 018h 시
 $2^3 = 8$ 의 13다.



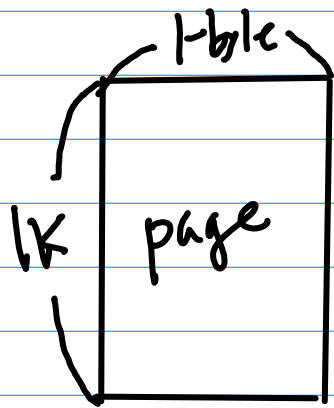
Virtual
mem





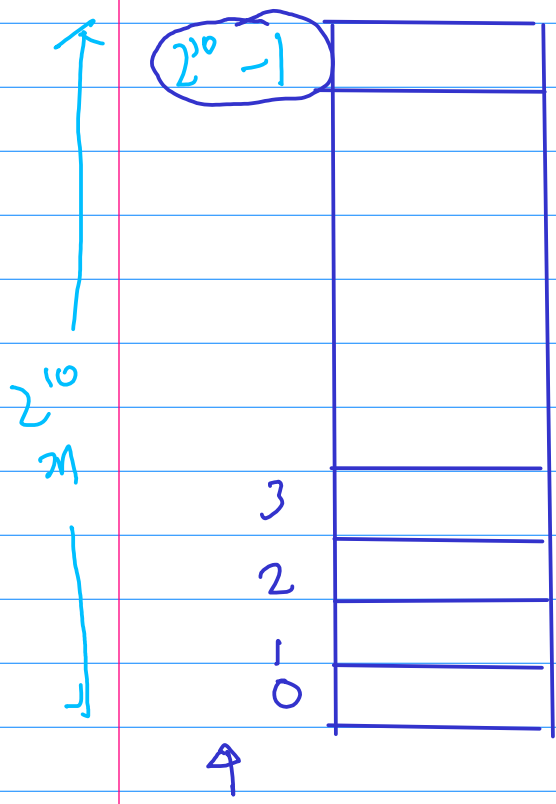


page size
 $1K \times 8bit$
1KB



page size 1KB

page table \rightarrow array



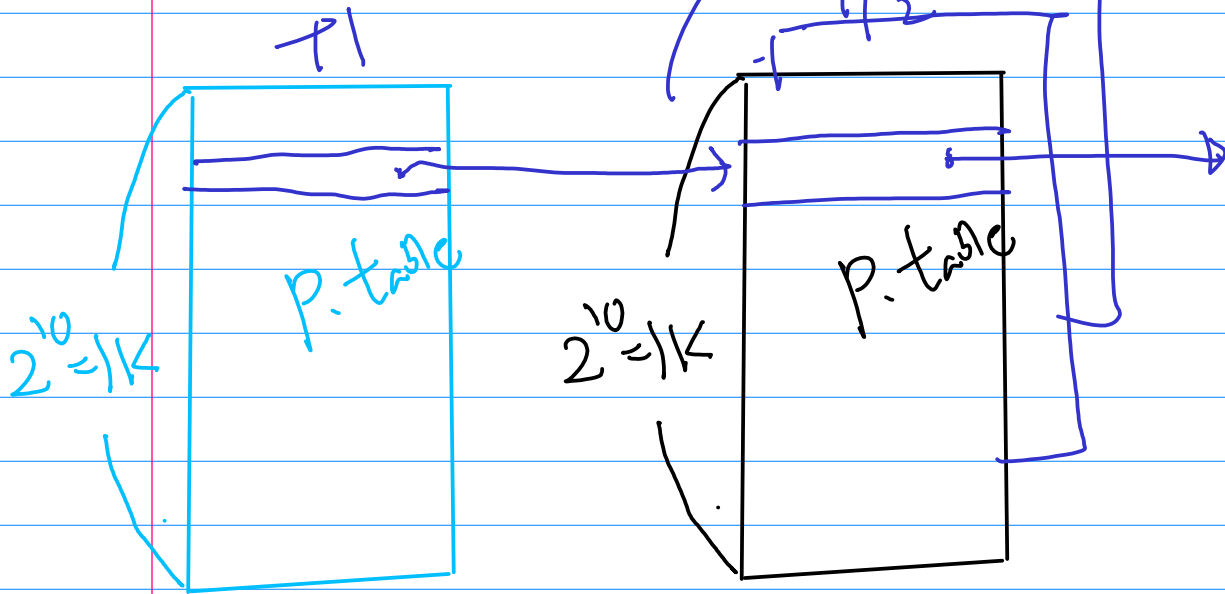
array of page #

10 bit

10-bit: offset \rightarrow page size

$$\frac{1K \times 8bit}{1KB}$$

1KB



page table must be stored in memory

① Size

page table을 저장할 때
메모리를 너무 많이 차지하면 안된다

⇒ 2-level, 3-level Paging

② access time

$(v. address) \rightarrow (p. address) \rightarrow (\text{memory RD/WR})$

· page table을 access
memory access

⇒ TLB (Translation Lookaside Buffer)

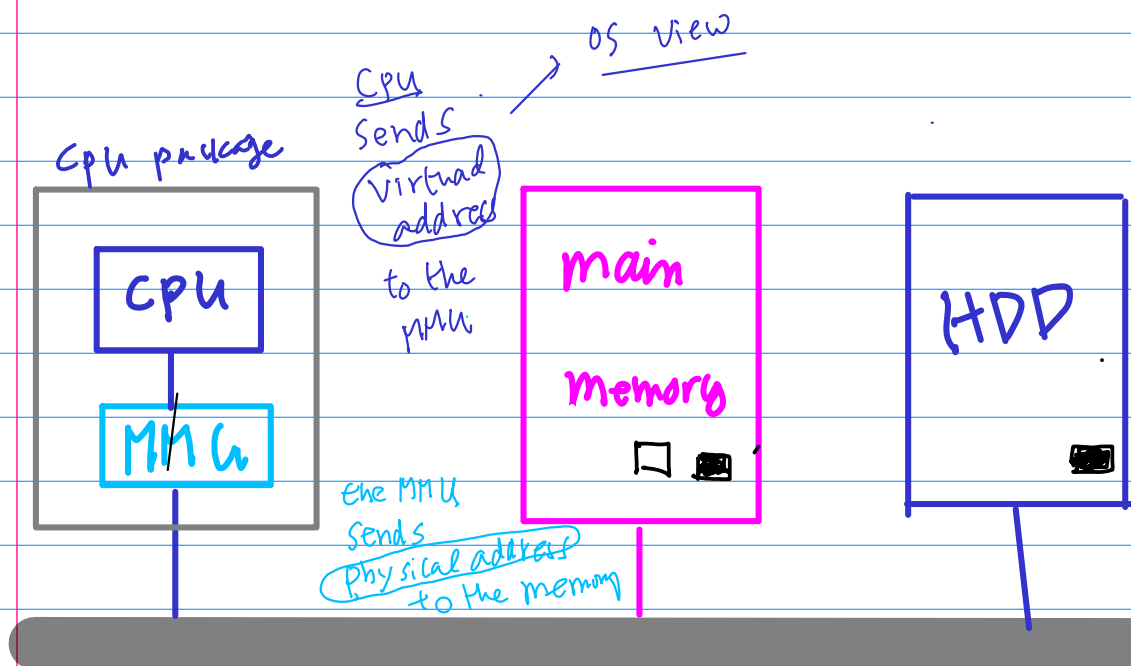
Page Fault

원하는 page가 physical memory에
load 되어 있지 않은 상태

원하는 page가 disk에 있어서
그 page를 physical memory에 load 해야 함

→ replace 기존의 load된 page를 다시 disk로
교체 필요 replacement policy

MMU, OS

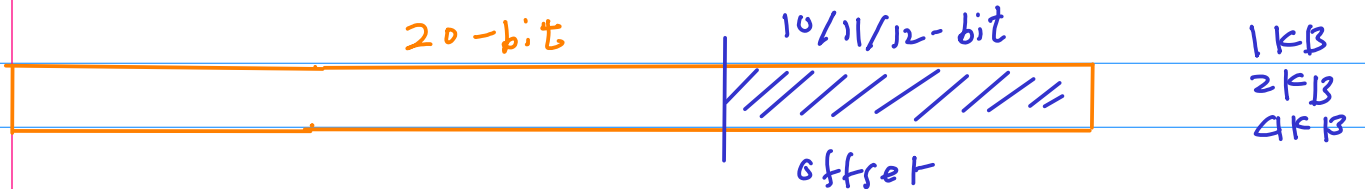


OS keeps track of mapping \textcircled{V} addr to \textcircled{P} addr.

MMU translate \textcircled{V} addr to \textcircled{P} addr

Multi-level Page Table

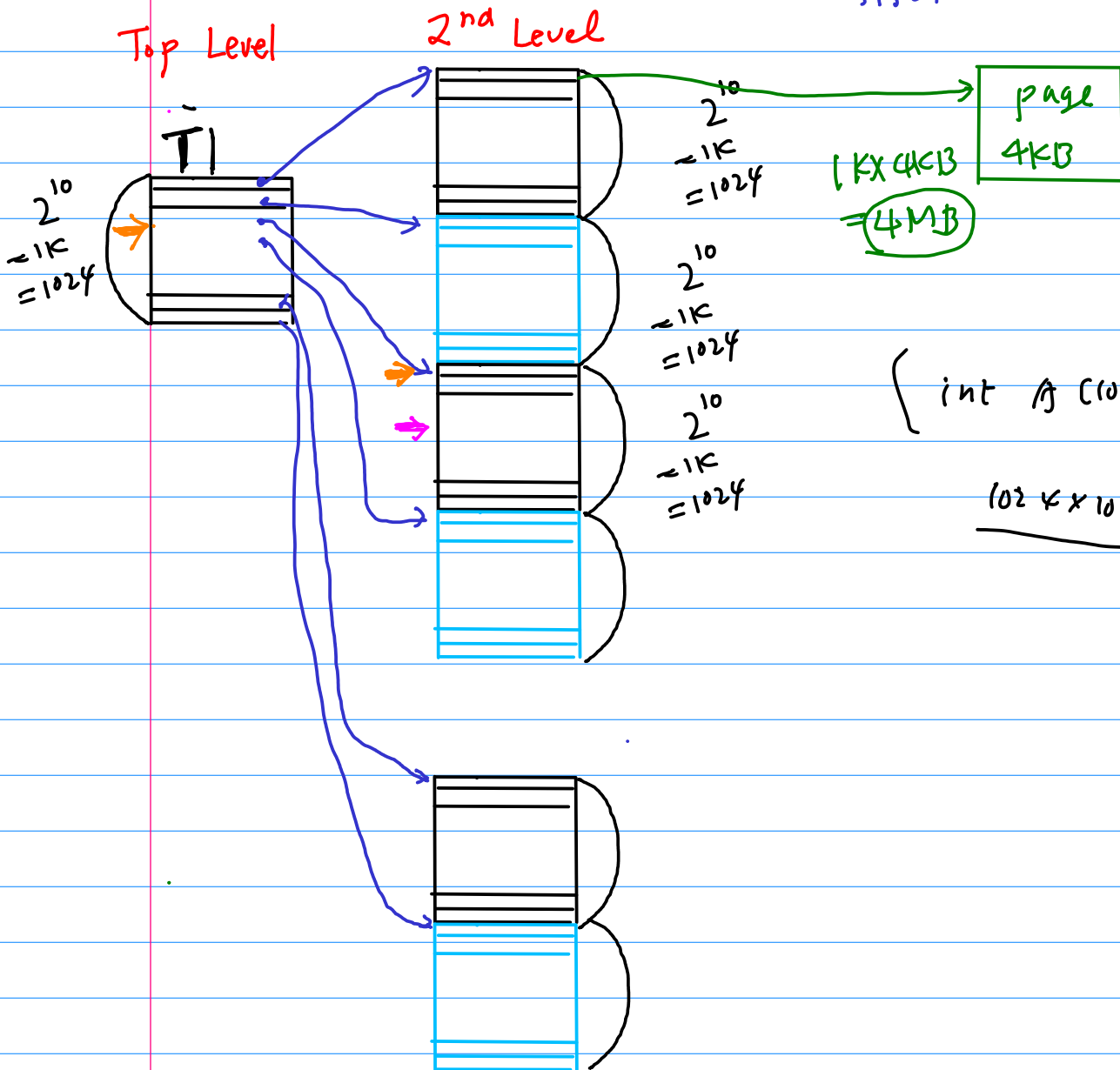
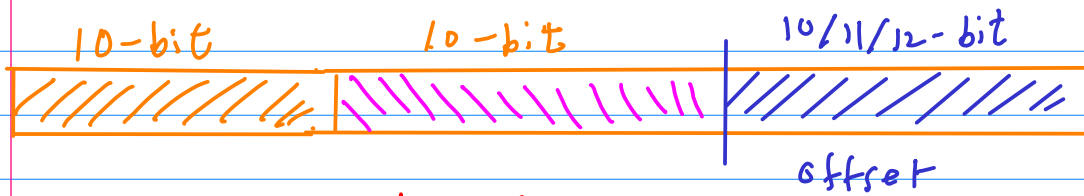
Virtual address (p, 0)



$2^{20} \rightarrow 1M$

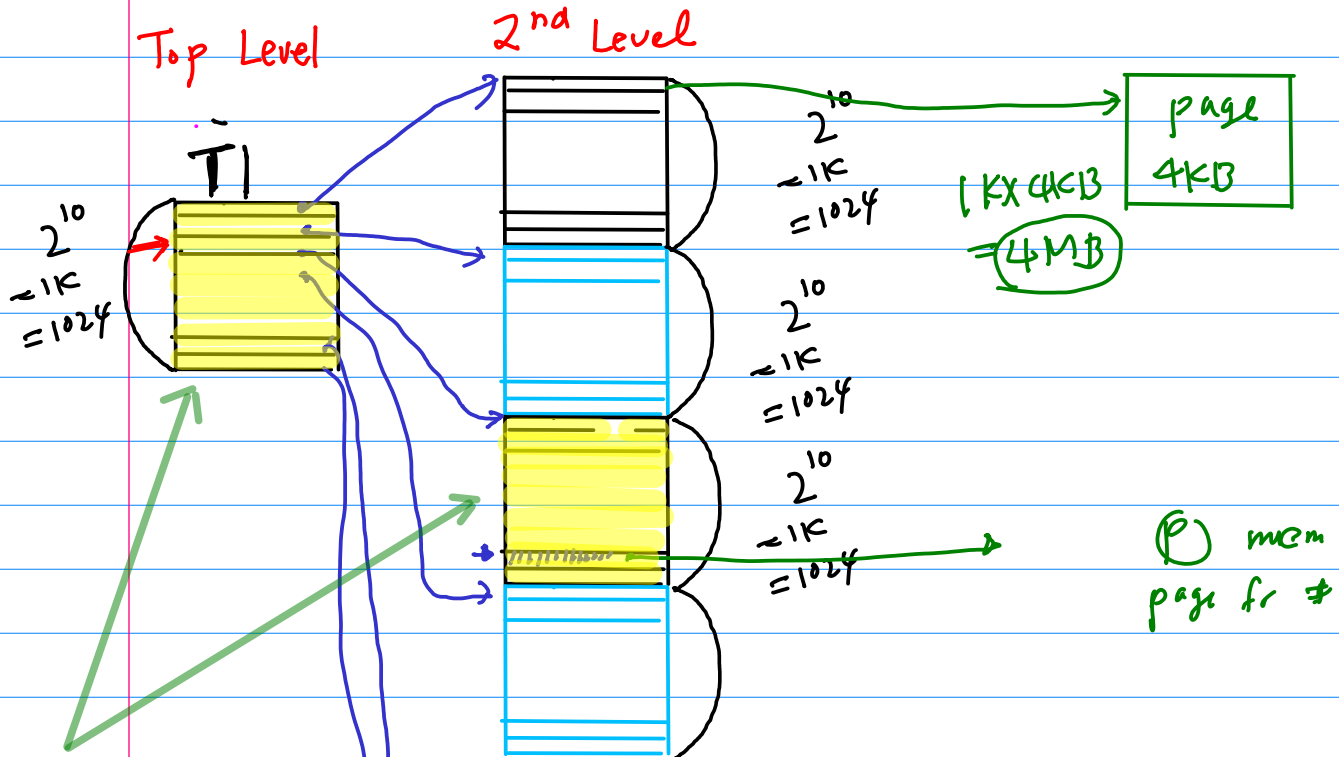
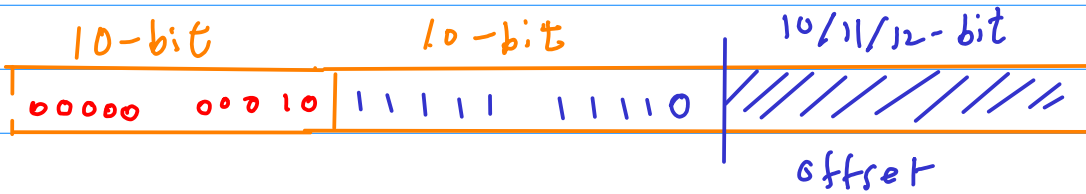
```
int A [220]  
int A [1048,516]
```

2-level Paging



$$(int\ A[1024]) \times 1024$$

$$\underline{1024 \times 1024 = 1024}$$



Only these two tables

are loaded in the memory.

