

①

Simple page tables

Frame = block for below

- ① start with main size (128 rows or 7 bits)
- ② decide block size below we have 8 blocks
- ③ 4 bits to address each block $\log_2(\text{block-size}) = 3 \text{ bits}$
for left we need $\log_2 16 = 4 \text{ bits}$
4 bits to get to every cell in a block

0	05
16	X
32	
64	3
	0
	X
	2
	X
128	1

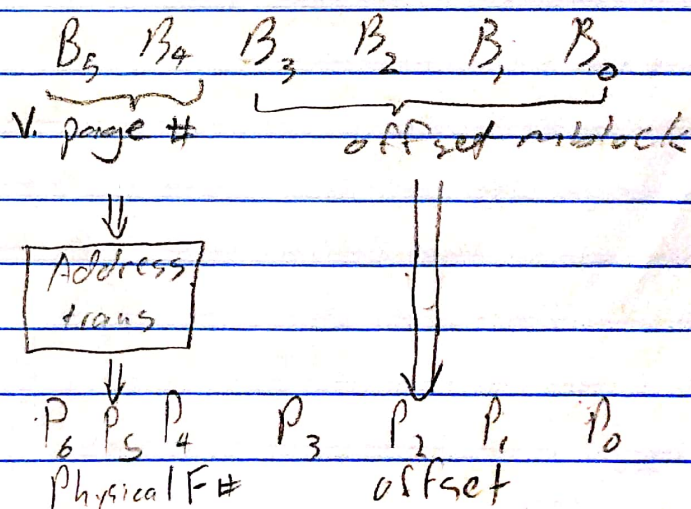
④ Now how about your process?

say, it takes 4 blocks

⑤ each block is 16 bytes long

Page #	0
	16
1	32
2	48
3	64

bits to address it all = $\log_2 64 = 6$



Page table

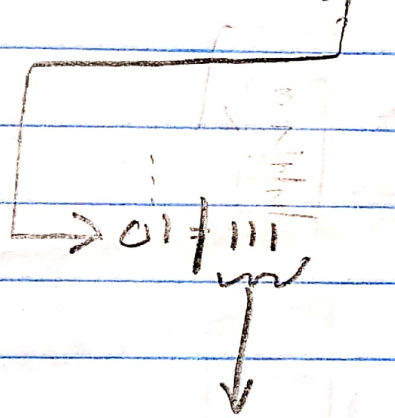
index	VP	PF
0	3	
1	7	
2	5	
3	2	

\Rightarrow

00	011
01	111
10	101
11	010

say we have `mov 21 %eax` load `%eax` from 21

21 \Rightarrow bin \Rightarrow 010101

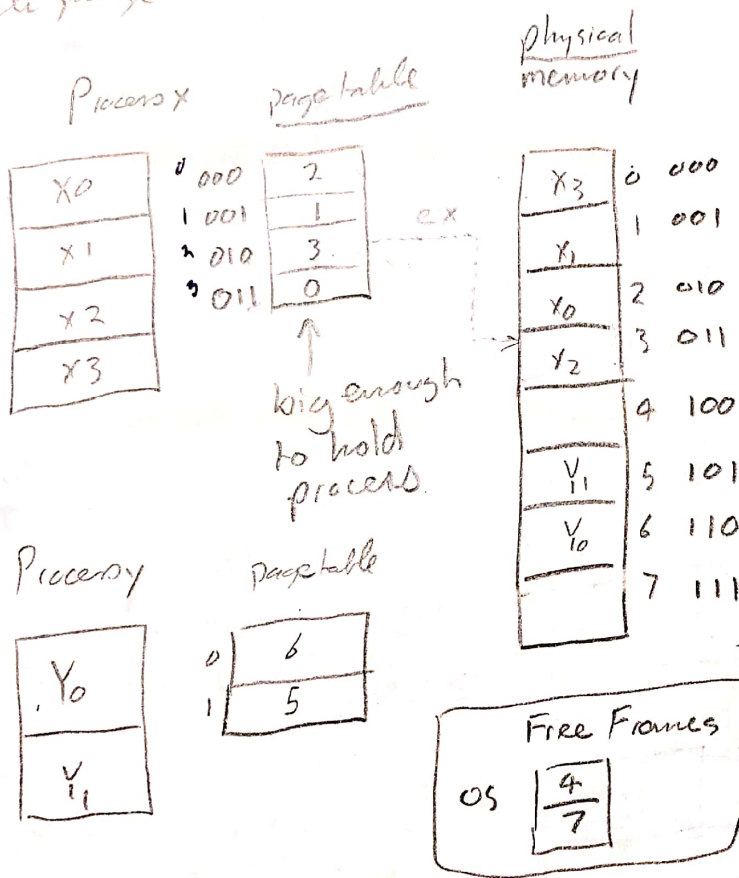


111 0101 \leftarrow physical address

paging

64 bits 64 frames 26 frames each with 2¹¹ words

each page table kept with process control block



say have 8 bit address space

have 8 frames (36 bits)
frames have 2⁵ = 32 words

but still need all of process in mem at once!
also internal frag on last block of process.

ex 8 bit logical address for process X

logical

010 00001 ⇒ 011 00001

page # offset

see above page table