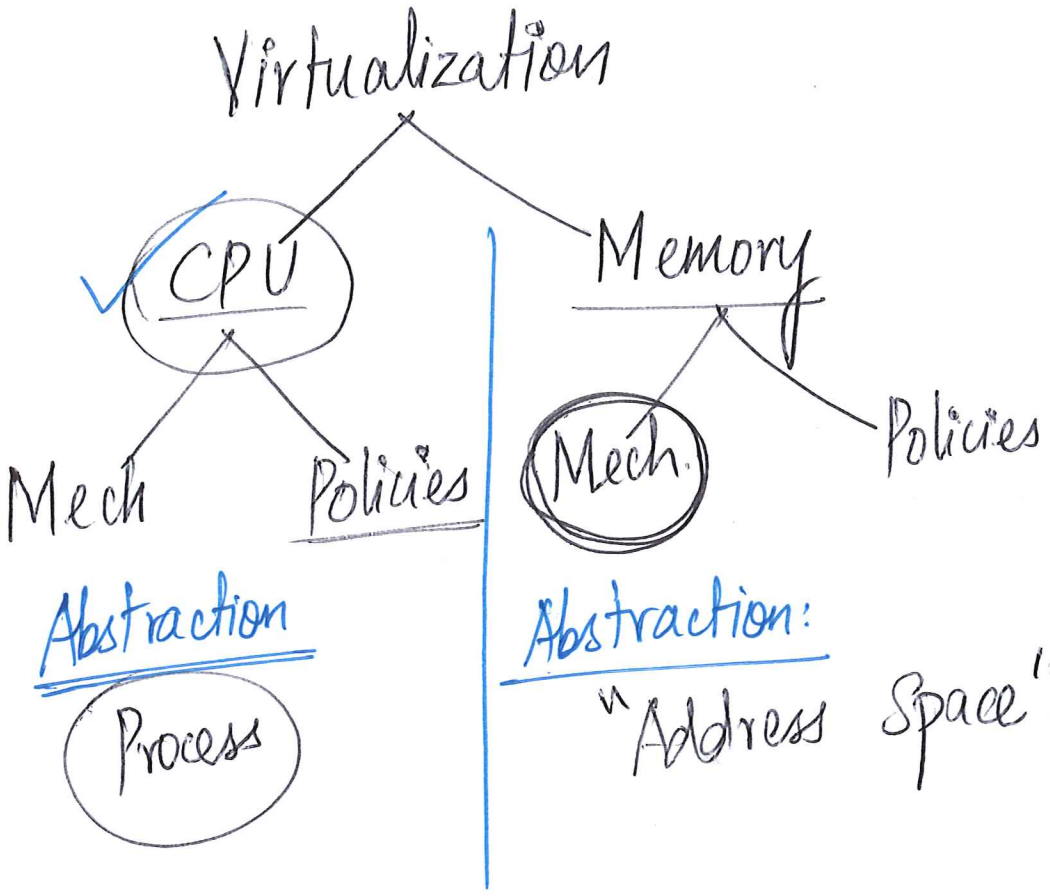
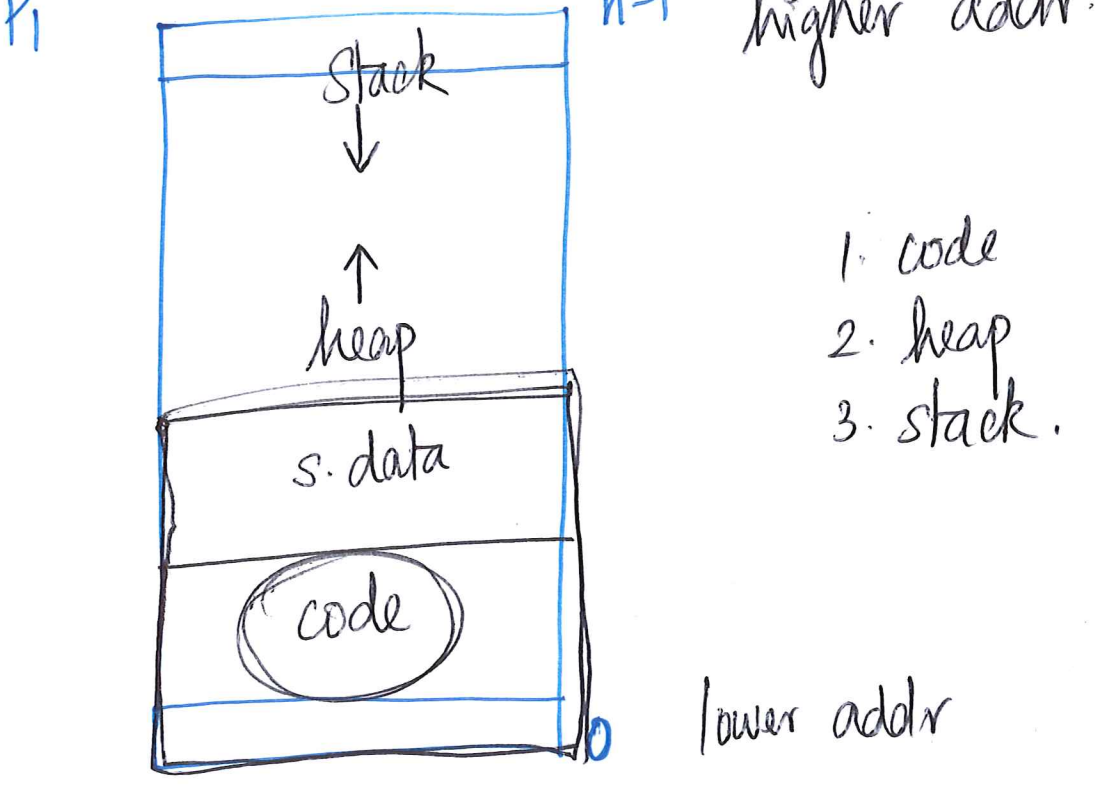
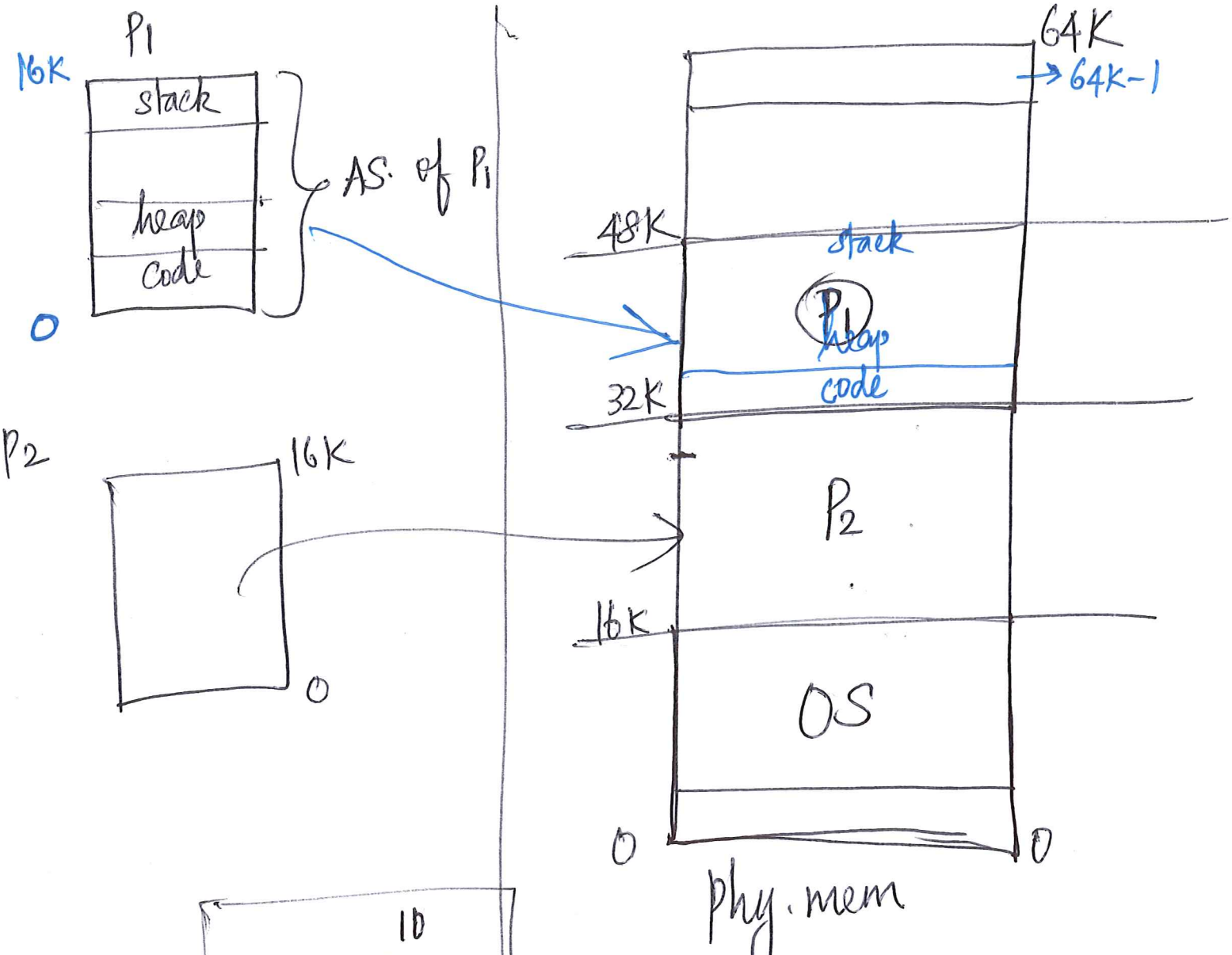


Address Space



1. code
2. heap
3. stack.



$K = 2^{10}$

$M = 2^{20}$

$G = 2^{30}$

Assumptions

1. process' A.S. is contiguous in phy. mem
2. A.S. of a process < phy mem.
3. All process have the same size of A.S.

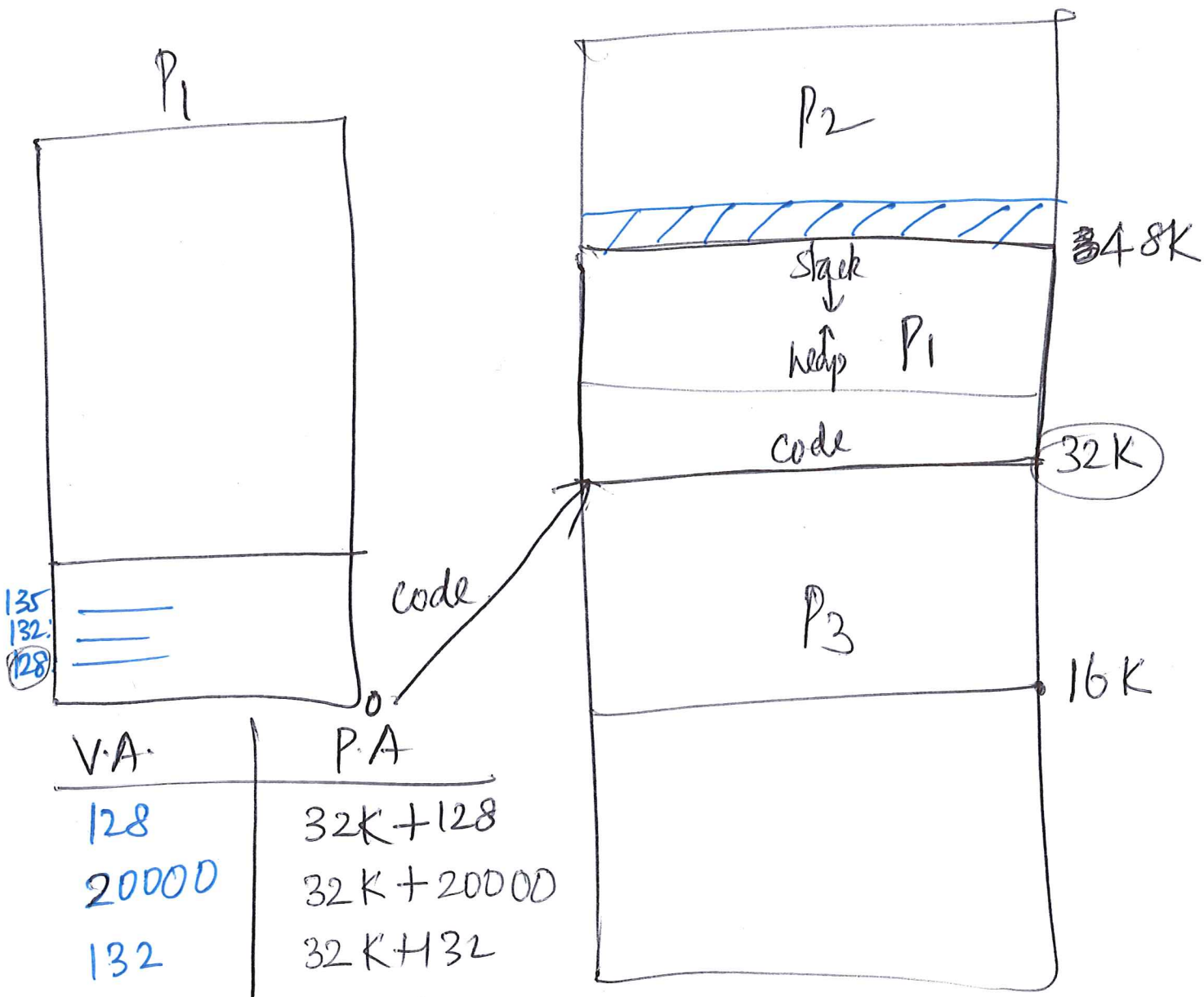
```
void foo() {
    int x = 100;
    x = x + 10;
}
```

20000

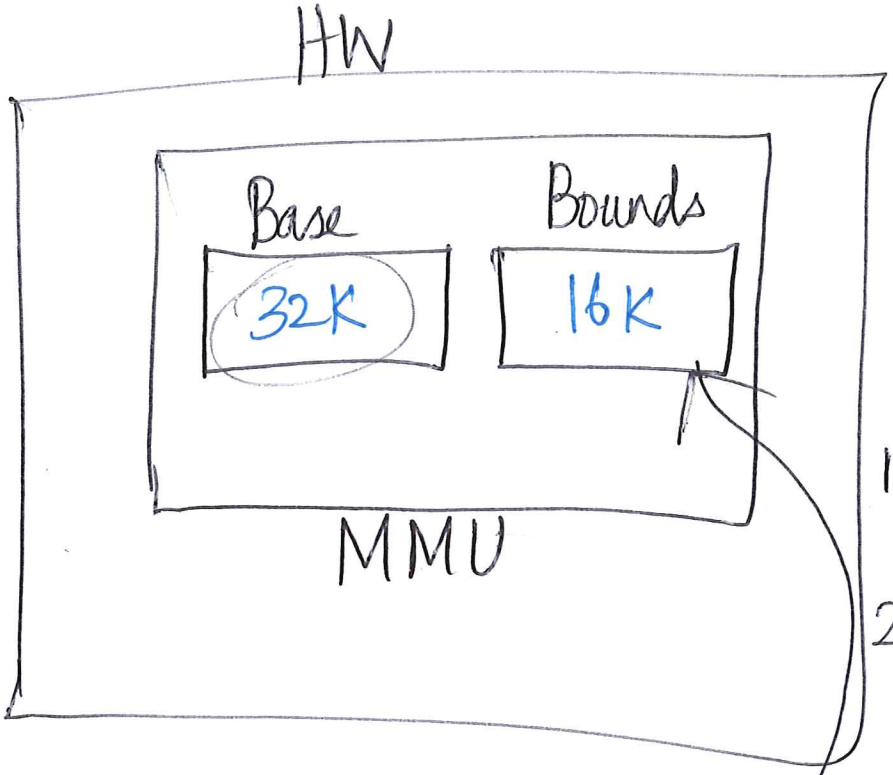
128: movl ~~0x1004~~, %eax

132: addl $10, %eax

135: movl %eax, ~~0x1004~~ 20000



V.A.	P.A
128	32K+128
20000	32K+20000
132	32K+132
135	" +135
20000	" +20000



1. 1024 < 16K ✓

2. PA = 32K + 1024

P1

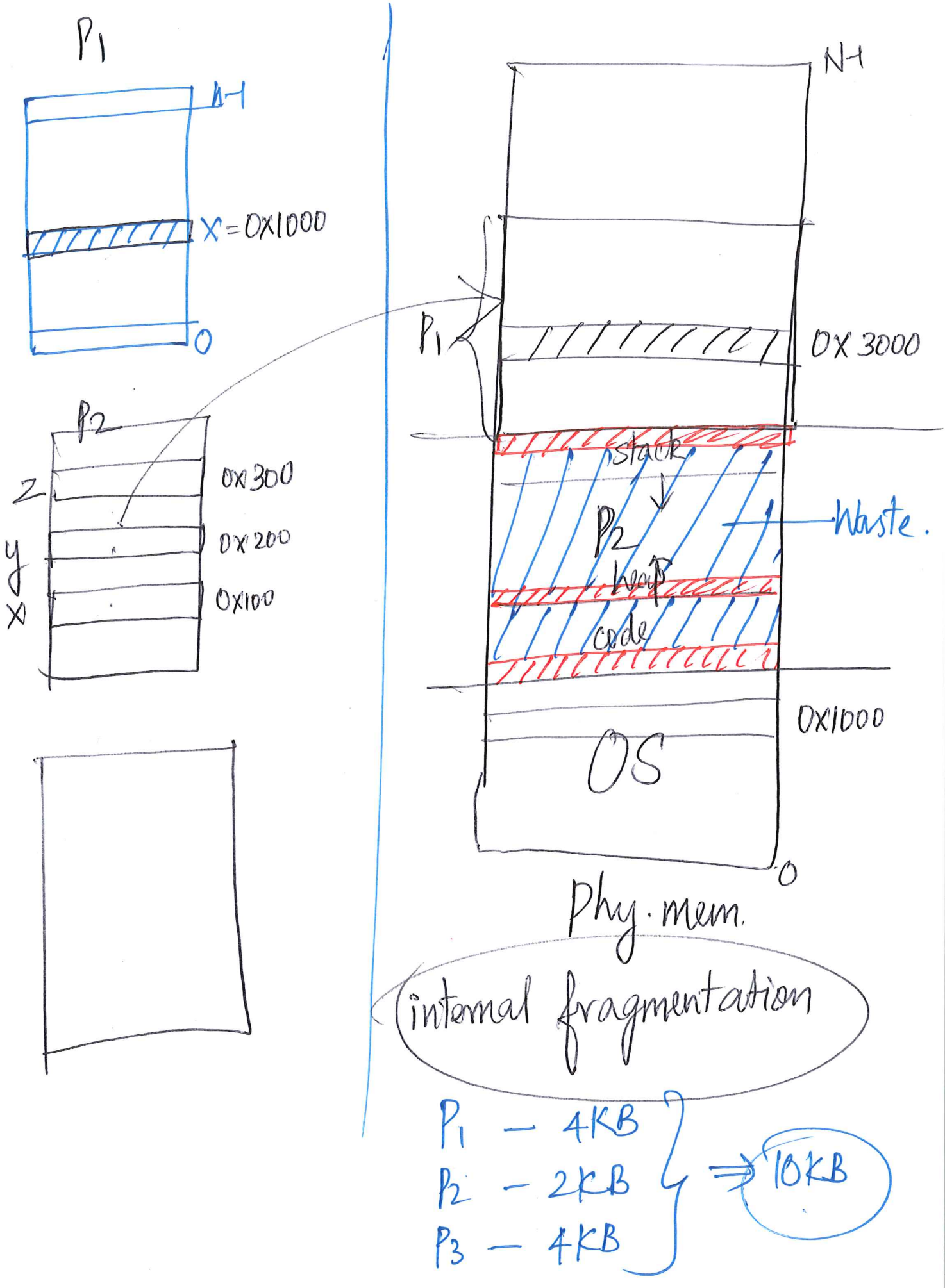
V.A: 1024

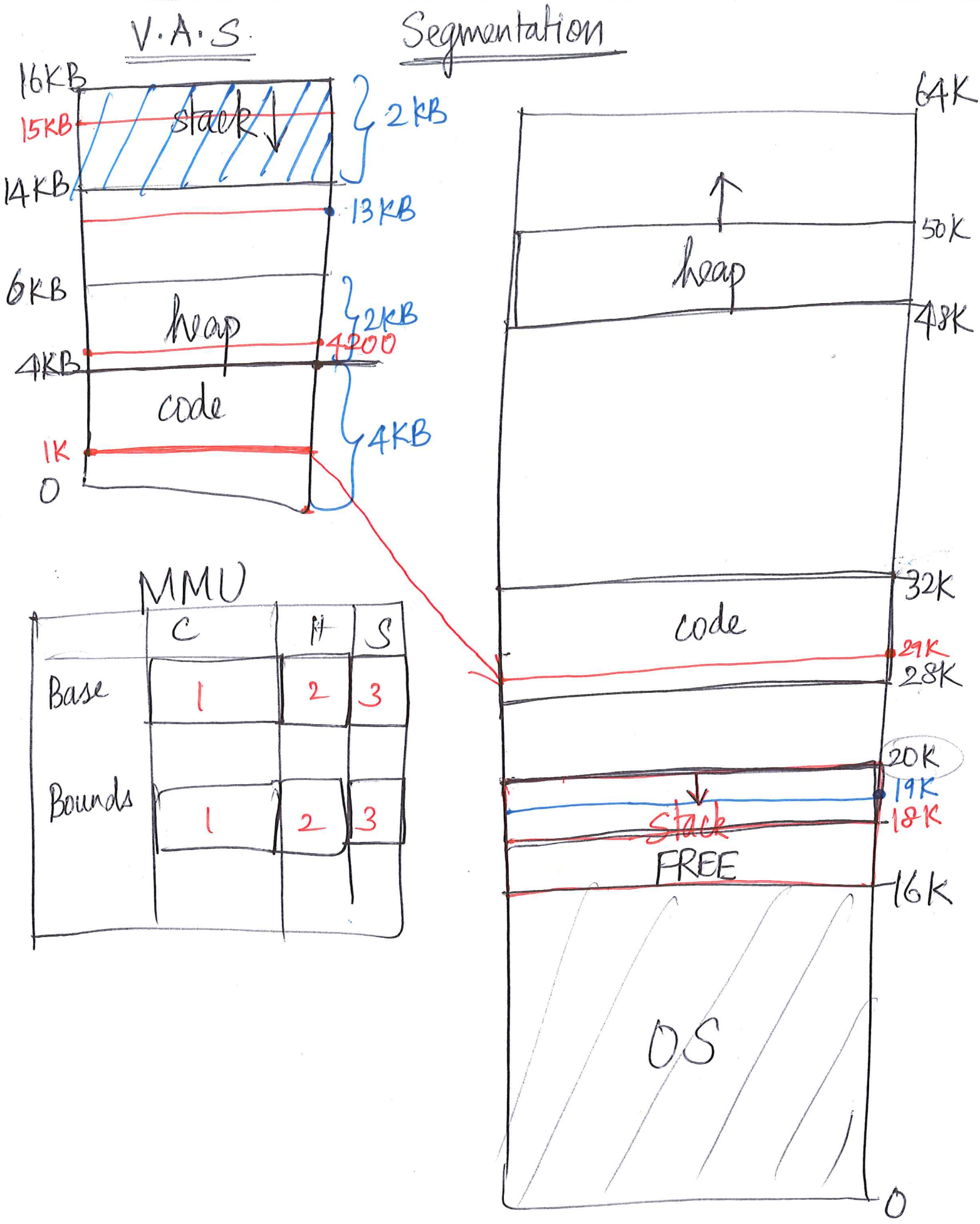
P.A. = Base + V.A.

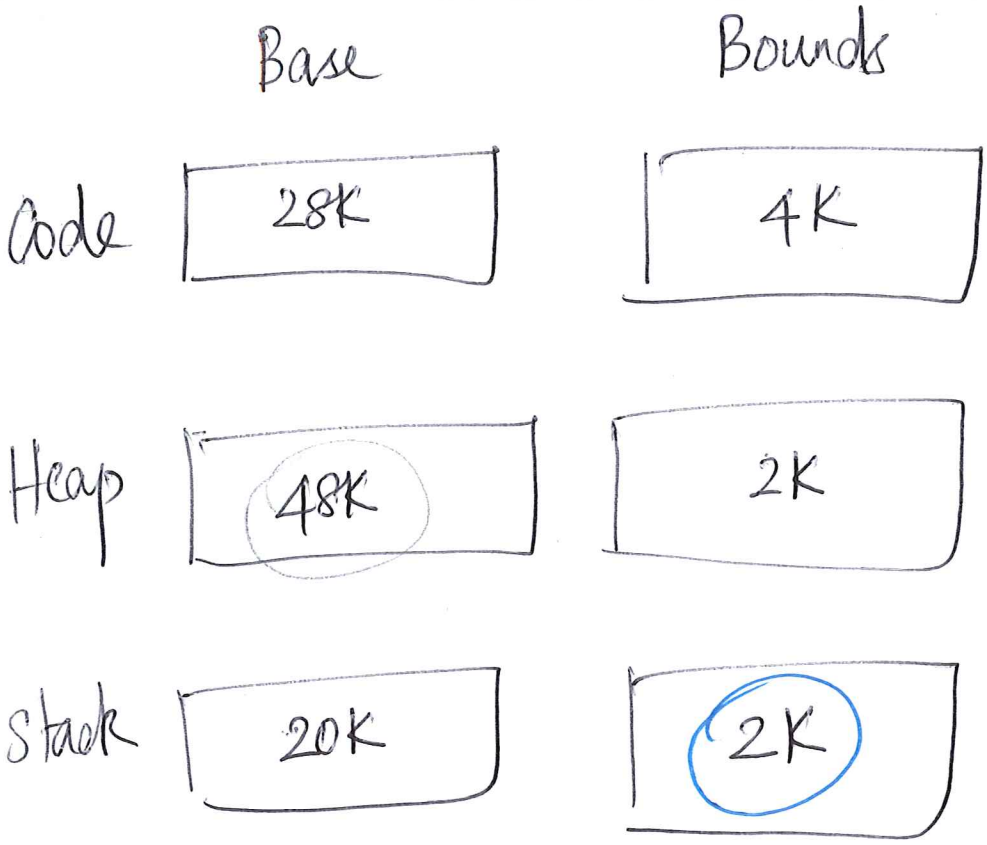
Base and Bounds Relocation

Dynamic

HW-based.


P1
N-1
X = 0X1000
0
P2
Z
y
x
0X300
0X200
0X100
N-1
P1
0X 3000
Stack
P2
Waste.
heap
code
0X1000
OS
0
Phy. mem.
internal fragmentation
P1 – 4KB
P2 – 2KB
P3 – 4KB
⇒ 10KB


V.A.S.
Segmentation
16KB
15KB
stack
2KB
14KB
13KB
6KB
heap
2KB
4200
4KB
code
4KB
1K
0
MMU
C
H
S
Base
1
2
3
Bounds
1
2
3
64K
heap
50K
48K
32K
code
29K
28K
20K
19K
18K
stack
FREE
16K
OS
0



Code

V.A: 1024 = 1K (code)

P.A

28K + 1K = 29K

V.A: 4200 (Heap)

offset = 4200 − 4096

= 104

P.A.

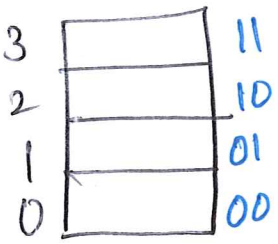
48K + 104

Size (V.A. Space) = 16 KB = 16×2^{10} bytes

$= 2^4 \times 2^{10} = 2^{14}$ bytes

V.A:

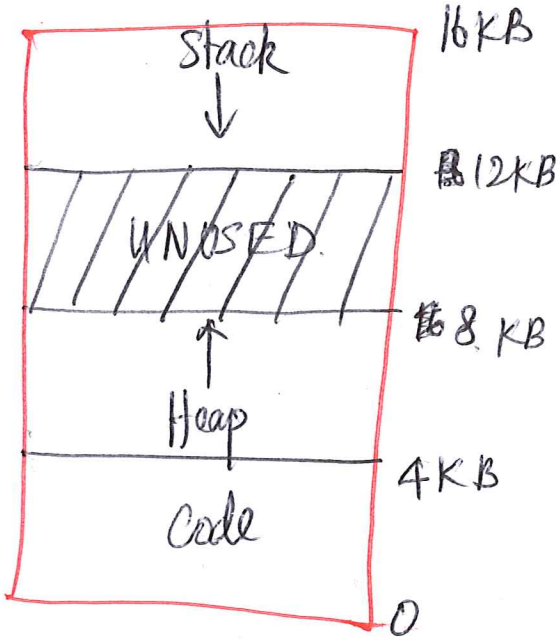
0 0	0 1 0 0	0 0 0 0	0 0 0 0
2^{13} 2^{12}	1024 512 256	128 64 32 16	8 4 2 1
0	4	0	0



1024 = 0x 0400

4 = 2^2 bytes

2 MSBs	Seg
00	Code
01	Heap
11	Stack



16 | 4200

16 | 262 − 8

16 | 16 − 6

1 − 0

4200 = 0x 1068

0 1	0 0 0 0	0 1 1 0	1 0 0 0
			8 4 2 1
1	0	6	8

$4GB = 4 \times 2^{30} = 2^{32}$

Stack

1 1	1 1 0 0	0 0 0 0	0 0 0 0
2^{13} 2^{12}	2^{11} 2^{10} 2^{9} 2^{8}	2^{7} 2^{6} 2^{5} 2^{4}	2^{3} 2^{2} 2^{1} 2^{0}

$15\,KB = 0x\,3C00$

$2^{11} + 2^{10}$

$= 2^{10}(2+1)$

$= 3K$

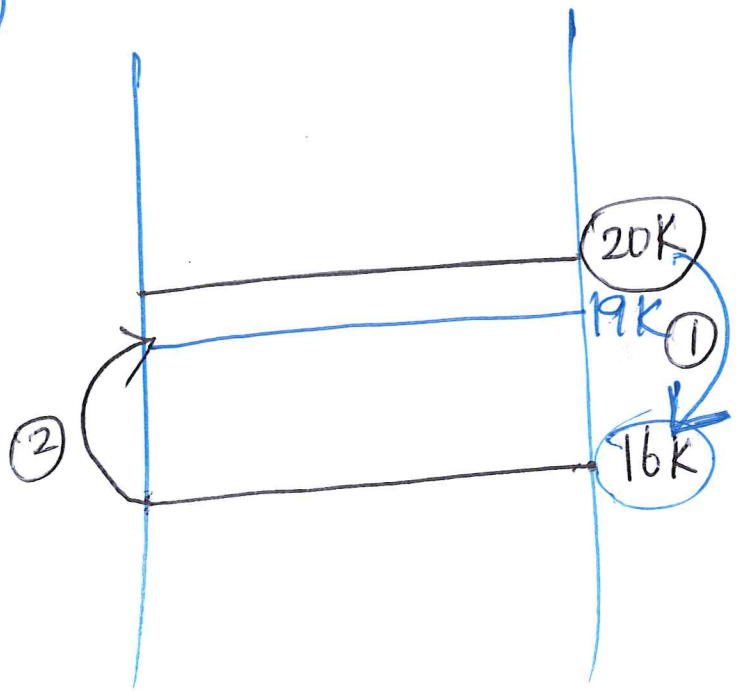
$P.A. = \text{Base} - \text{max size of stack} + \text{initial offset}$

$20K \quad 4K \quad 3K$

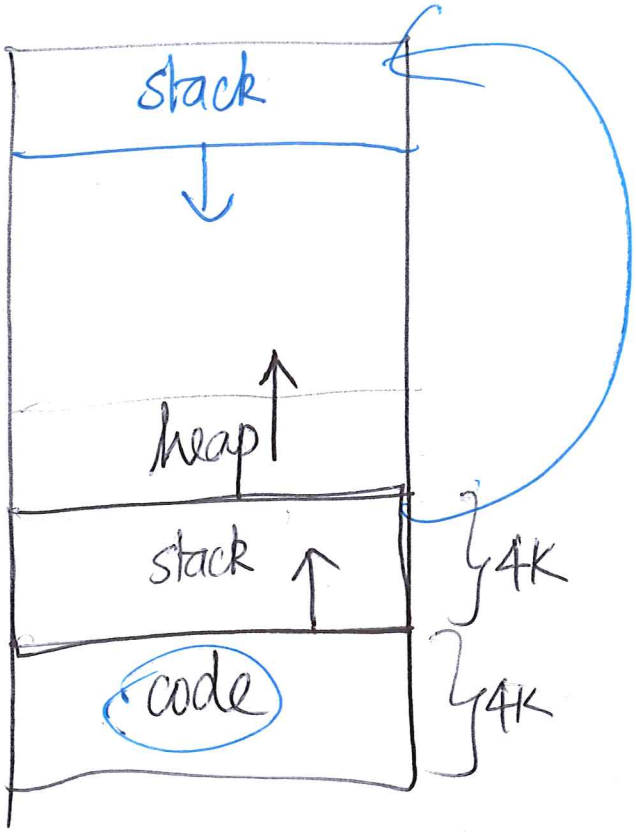
$= 20K - 4K + 3K$

$= 20K - 1K$

$= 19K$



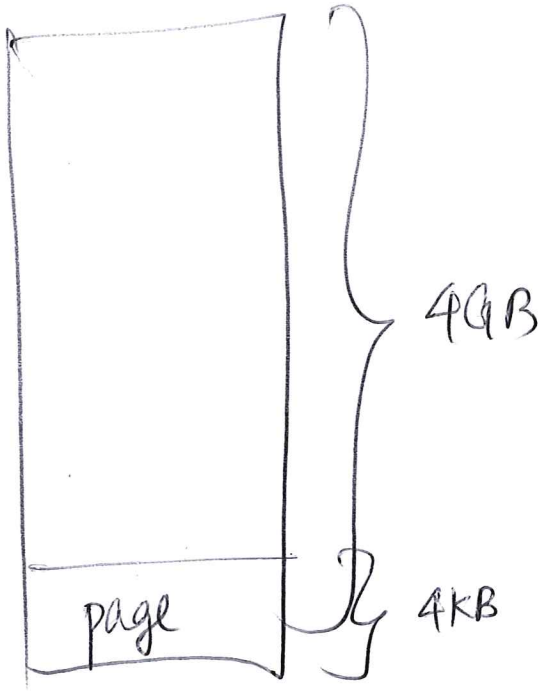
XV6



Seg	Base	Bounds	Growing up?	Perm
Code			1	r x
H			1	r w
S			0	r w

4GB ⟶ 32 bits V.A.

page size = 4KB.

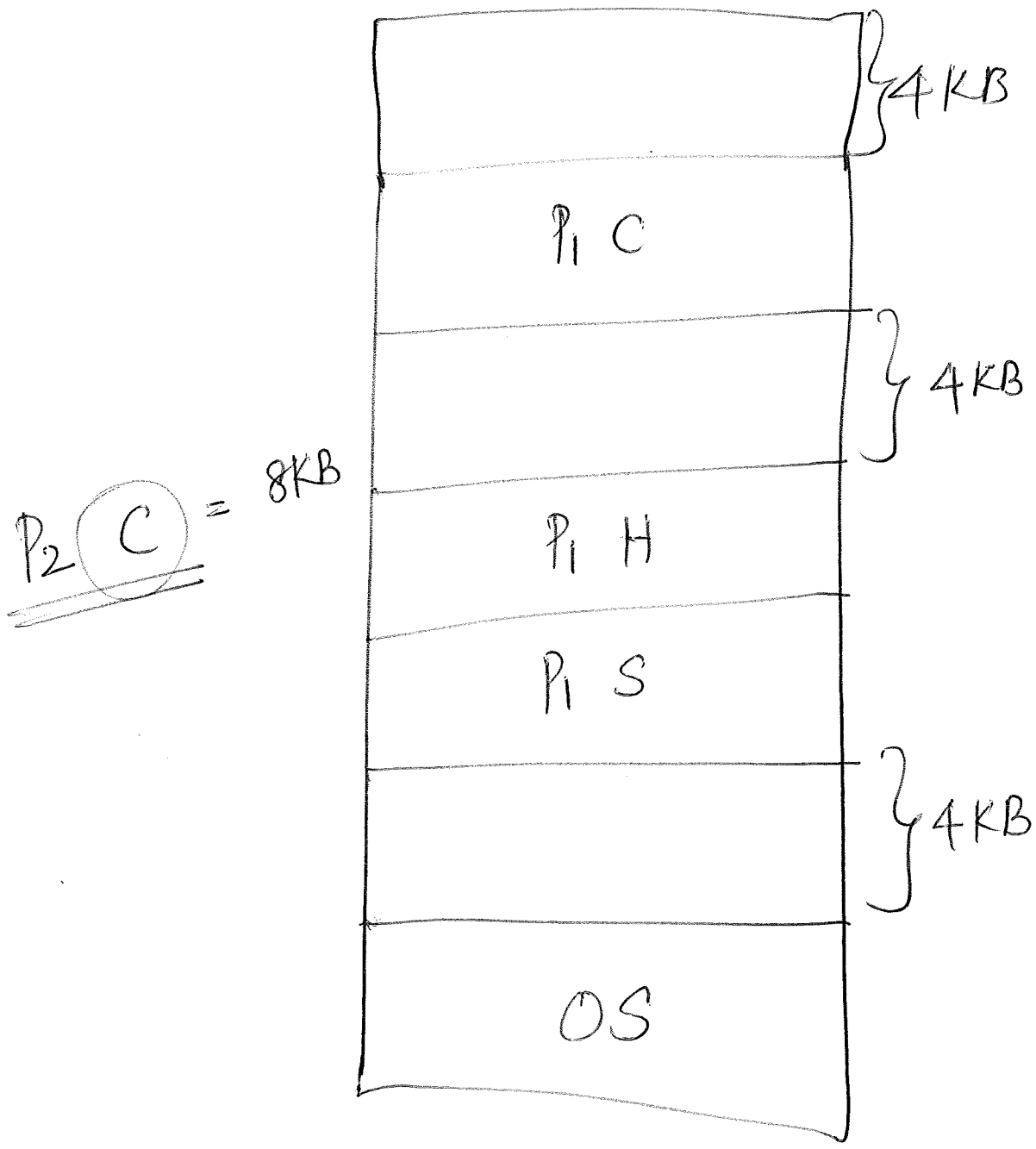


$$\#\text{pages} = \frac{4GB}{4KB}$$

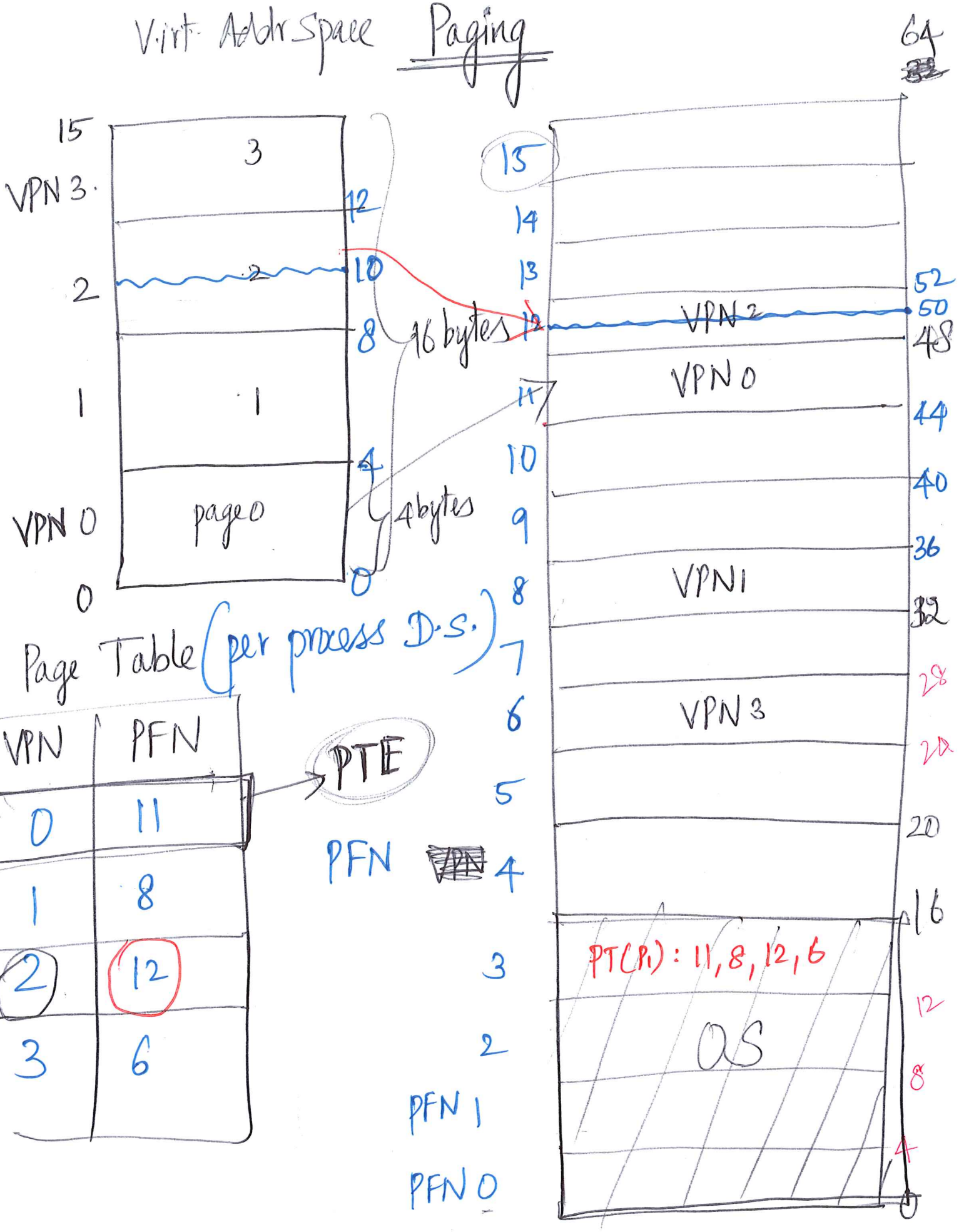
$$= \frac{4 \times 2^{30}}{4 \times 2^{10}}$$

$$= 2^{20}$$

$\simeq$ 1 million V. Pages.


4KB
P1 C
4KB
P2 C = 8KB
P1 H
P1 S
4KB
OS

Virt Addr Space — Paging



Page Table (per process D.S.)

VPN	PFN
0	11
1	8
2	12
3	6

PTE

PFN

V.A. = 10

16 bytes = 2^{4} bytes

VPN | offset

1 0 | 1 0

8 4 | 2 1

4 bits

page size = 4 bytes = 2^{2} bytes

VPN = 2

Addr Trans (using P.T.)

PFN = 12

P.A. = 1 1 0 0 | 1 0 = 50

32 16 8 4 | 2 1

8 4 2 1

PFN (4 bits) | offset