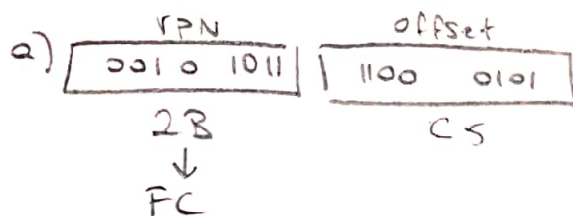Selin Kırmacı
21802177
Section-003
HW #4

1) Page Size $= 256 = 2^8$

VPN: 8 bits | offset: 8 bits

a)

VPN: 0010 1011 | offset: 1100 0101

28 → FC
C5

→ FCC5 physical address

b)

0001 0011 | 0001 1100

13 → a3
1C

→ A31C physical address

c)

1110 0100 | 1010 0011

E4 → e5
A3

→ E5A3 physical address

d)

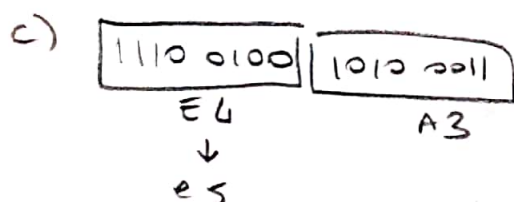0010 1011 | 0001 0111

2B → FC
17
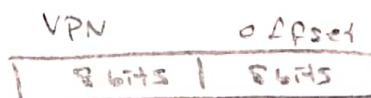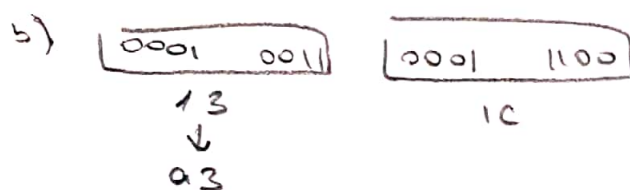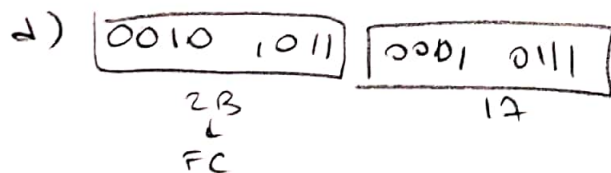
→ FC17 physical address

2)

| 9 | 9 | 9 | 9 | 12 |

page offset → 2^{12} = page size
$2^2 \times 2^{10} = $ 4 KB

1st level: 2^9 entrie
2nd level: 2^9 entrie
3rd level: 2^9 entries
4th level: 2^9 entries per page table
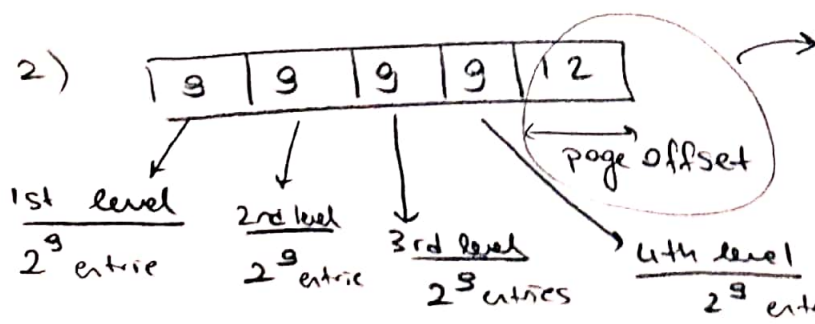
Forth level → $2^9 \times 2^{12} = 2^{21} = 2 \times 2^{20} = 2$ MB

32 MB / 2 MB = 16 forth level page tables required to map.

$16 \times 2^9 \times 2^3 = 2^{4+9+3} = 2^6 \times 2^{10} = $ 64 KB → forth level

$1 \times 2^9 \times 2^3 = 2^{10} \times 2^2 = $ 4KB → third level

$1 \times 2^9 \times 2^3 = 2^{10} \times 2^2 = $ 4KB → second level

$1 \times 2^9 \times 2^2 = 2^{10} \times 2^2 = $ 4KB → first level

64 + 4 + 4 + 4 = 76 KB RAM is enough

3)

```
40, 30, 60, 10, 80, 45
40, 60, 10, 80, 30, 45
40, 60, 10, 30, 80, 45
40, 60, 30, 10, 80, 45
60, 10, 40, 30, 80, 45
60, 10, 40, 80, 30, 45
60, 10, 80, 40, 30, 45
60, 40, 30, 10, 80, 45
60, 40, 10, 30, 80, 45
60, 40, 10, 80, 30, 45
```

Considering there may be context switches.

4)
```
sem  mutex = 1;
sem  tobaccoSem, paperSem, matchSem = 0;

Agent(){
   do{
       wait(mutex);
       put 2 random;
       signal(mutex);
       if (random = tobacco & paper)
           signal(matchSem);
       else if (random = tobacco & match)
           signal(paperSem);
       else
           signal(tobaccoSem);
   } while (1);
}

TobaccoMan(){
   do{
       wait(mutex);
       wait(tobaccoSem);
       take from table;
       signal(mutex);
       Smoke
   } while(1);
}

MatchMan(){
   do{
       wait(mutex);
       wait(matchSem);
       take from table;
       signal(mutex);
       Smoke
   } while(1);
}

PaperMan(){
   do{
       wait(mutex);
       wait(paperSem);
       take from table;
       signal(mutex);
       Smoke
   } while(1);
}
```

6)

a) segment 0 → 1024

$1024 + 50 = \underbrace{64 \times 16}_{\text{page size}} + \overbrace{50}^{\text{page i}}$

$64 \times (\underbrace{16 + 10}_{\text{frame}}) + 50 = \boxed{1714}$

b) segment 1 → $4196 + 0 = \underbrace{65 \times 64}_{} + 36$

$\downarrow +10$

$= 75 \times 64 + 36 = \boxed{4836}$

c) segment 1 → $4196 + \underline{100} = 4296 = 62 \times 64 + 8$

$\downarrow +10$

$\Rightarrow 77 \times 64 + 8 = \boxed{4936}$

d) segment 1 → $4196 + \boxed{700}$ → problem because
Segment 1 has 512 length which is less than 700.
Out of the boundary.

e) segment 2 → $128 + 10 = 2 \times 64 + 10$

$\downarrow +10$

$\Rightarrow 12 \times 64 + 10 = \boxed{778}$

f) segment 3 → $2048 + 200 = 2748 = 35 \times 64 + 8$

$\downarrow +10$

$= 45 \times 64 + 8 = \boxed{2888}$

7)

Existing

A	B	C
9	6	6

Available

A	B	C
2	1	0

Need

4	2	1
2	1	1
4	4	2
2	2	0
2	1	2

State is not safe because none of the processes can be executed.

8)

Available

A	B	C
3	3	0

P4 → 4   3   1

P1 → 5   3   1

After executing P4 and P1 no other process can be executed so deadlock.

9)
```
monitor Semophore
{
    int count;
    condition c;

    void initialize(int N){
        count = N;
    }

    void wait(){
        if (count == 0)
            c.wait();
        count --;
    }

    void signal(){
        count ++;
        c.signal();
    }
}
```