21600880

Sec-3

CS 342 - Hw2

Q1) There are N # of processes. The head process has a duration of N units time. The tail has 1 time unit.

a) For FCFS:

First process waits : 0

Second " " : N

Third " " : N + (N-1)

$\vdots$

N^{th} process waits : N + (N-1) + (N-2) + ... + 2

N-1 processes

$$(N-1)\cdot N + (N-2)\cdot(N-1) + (N-3)\cdot(N-2) + \cdots 2\cdot 1 =$$

$$\Rightarrow \sum_{k=0}^{N-1} (N-k-1)\cdot(N-k) = \frac{(N+1)\cdot N\cdot(N-1)}{3} = \frac{(N^2-1)\cdot N}{3} \Rightarrow \text{Total waiting time.}$$

$$\text{Avg. waiting time} = \frac{(N^2-1)}{3}$$

b) For SJF:

First process waits: 0

Second process waits: 1

Third process waits : 1 + 2

$\vdots$

$(N-1)^{th}$ process waits: 1 + 2 + ... + N-2 wait

N^{th} process waits: 1 + 2 + ... + N-2 + N-1 wait

$$\Rightarrow \sum_{k=1}^{N-1} (N-k)\cdot k = \frac{1}{6}\cdot(N+1)\cdot(N-1)\cdot N \Rightarrow \frac{N^2-1}{6} \text{ avg. wait time}$$

For Round Robin: $(q = 1)$

wait for round 1: $1+2+3+4+ \cdots + N-1 = \frac{N.(N-1)}{2}$

wait for round 2; $1+2+3+ \cdots + N-2 = \frac{(N-1)(N-2)}{2}$

⋮

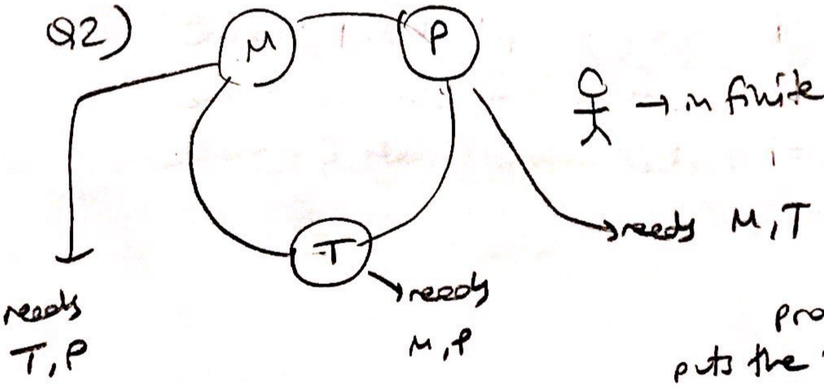
wait for round $(N-1)$: 1

$$\Rightarrow \frac{N.(N-1)}{2} + \frac{(N-1).(N-2)}{2} + \frac{(N-2)(N-3)}{2} + \cdots + 1$$

$$\Rightarrow \sum_{k=0}^{N-1} \left(\frac{(N-k).(N-1-k)}{2} \right) = \frac{1}{6} . N.(N^2-1)$$

Average wait time $\Rightarrow \frac{1}{6}.(N^2-1)$

Q2)



→ infinite provider

* we should arrange the provider in a way that he/she puts the two materials at the same time.

---

tp → items[0]
mp → items[1]
tm → items[2]

Semaphore items[3];
Semaphore tableEmpty;

Code for Smokers

```
do{
    //smokers wait for items[i]
    wait(items[i]);
    //smokers smoke
    signal(items[i]);
}while(true);
```

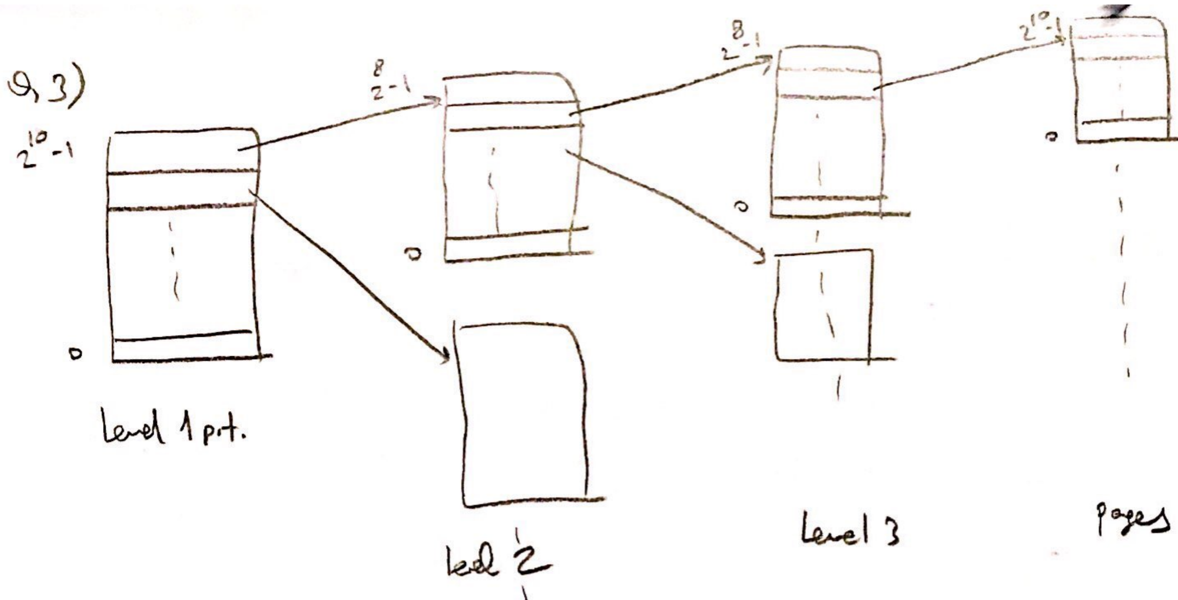
Code for provider

```
do{
    wait (isEmpty)
    int k = rand() % 3;
    signal(items[k]);
}while(true);
```

Scanned with CamScanner

a)

Q,3)

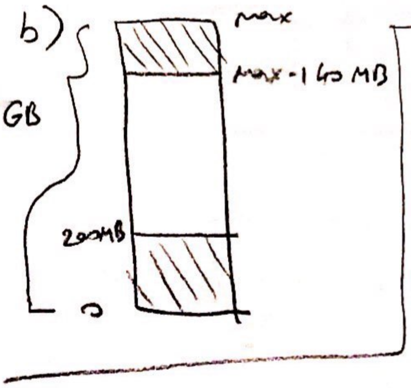


10	8	8	10
level 1 page #	level 2 page #	level 3 page #	page offset

a) Pagesize is 2^{10} Bytes because 10 is the page offset. We can have 2^{10} entries in one page.

2^{10} Bytes = $\underline{\underline{1KB}}$

b)



One of the level 3 page tables can map $2^8 \cdot 2^{10} = 2^{18}$ Bytes of logical address space.

For 200 MB $\quad \dfrac{200\,MB}{2^{18}\,B} = \dfrac{2 \cdot 100 \cdot 2^{20}\,B}{2^{18}\,B} = 800$ level 3 pages needed.

→ we need to find how many level 2 pages are needed to map 800 level 3 page tables. A level 2 page table can map 2^8 level 3 page tables.

$$\frac{800}{2^8} = \frac{800}{256} \simeq 3.12 \Rightarrow \lceil 3.12 \rceil = 4 \text{ level 2 pages are needed.}$$

For 140 MB

$$\frac{140\,MB}{2^{18}\,B} = \frac{2 \cdot 70 \cdot 2^{20}\,B}{2^{18}} = 560 \text{ level 3 pages are needed.}$$

$$\Rightarrow \frac{560}{256} \approx 2.19 \Rightarrow \lceil 2.19 \rceil = 3 \text{, level 2 pages are needed.}$$

In total: $3+4=\underline{\underline{7}}$ level 2 pages, $800+560=\underline{\underline{1360}}$ level 3 pages are needed.

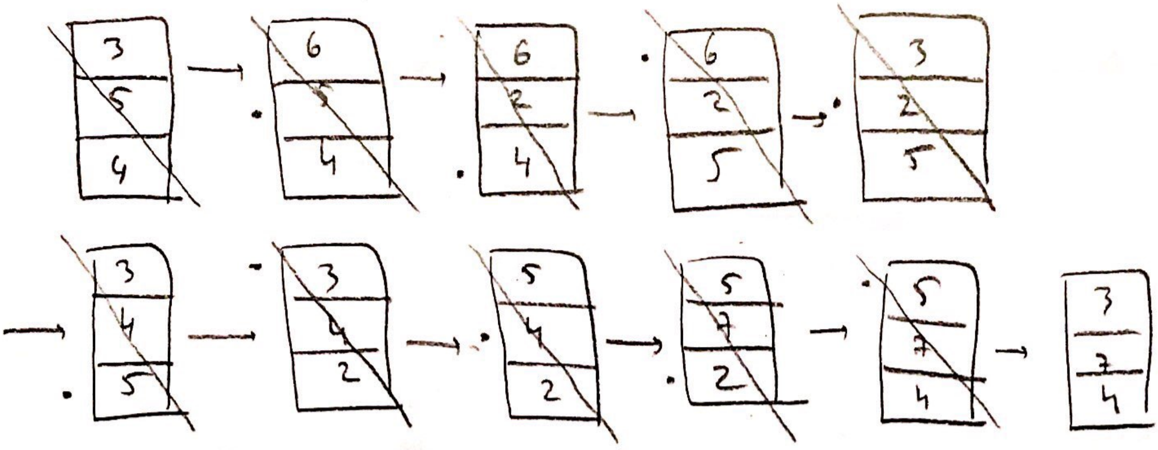
Q4)

a) FIFO

3 5 4 3 5 6 2 5 2 3 4 2 5 4 2 7 4 7 3

× × × ✓ ✓ × × × ✓ × × × × ✓ ✓ × × ✓ ×

⇒ 13 page faults

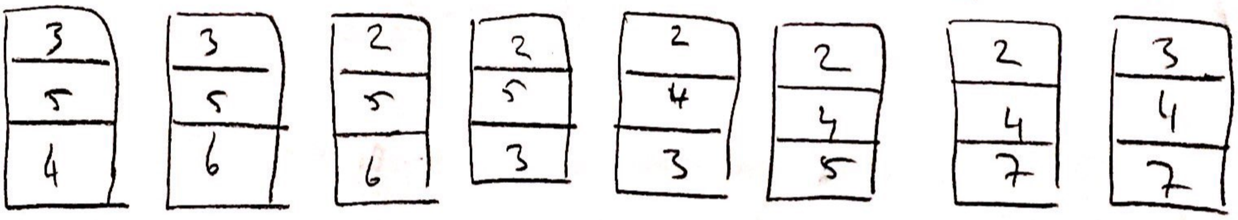


b) LRU

3 5 4 3 5 6 2 5 2 3 4 2 5 4 2 7 4 7 3

× × × ✓ ✓ × × ✓ ✓ × × ✓ × ✓ ✓ × ✓ ✓ ×

⇒ 10 page faults

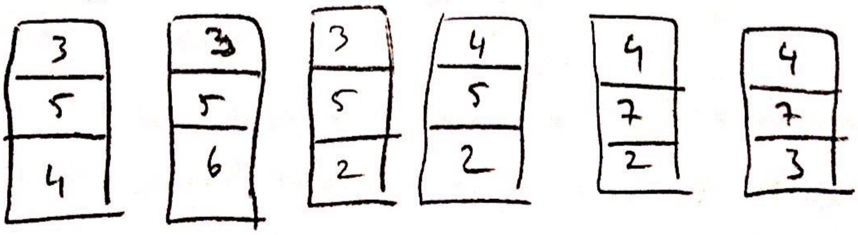


c) OPT

3 5 4 3 5 6 2 5 2 3 4 2 5 4 2 7 4 7 3

× × × ✓ ✓ × × ✓ ✓ ✓ × ✓ ✓ ✓ ✓ × ✓ ✓ ×

⇒ 8 page faults

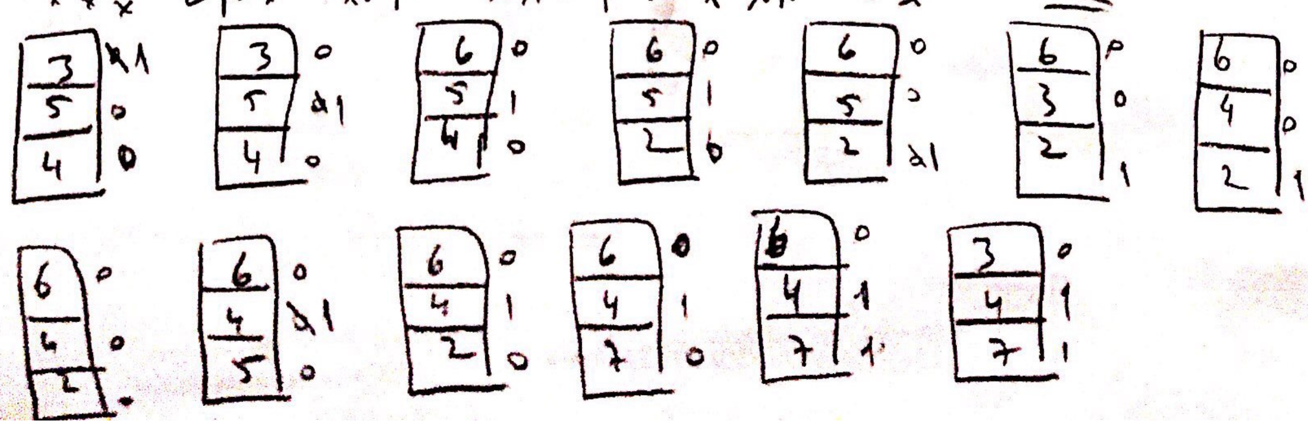


d) R-bit

3 5 4 3 5 6 2 5 2 3 4 2 5 4 2 7 4 7 3

× × × ✓ ✓ × × ✓ ✓ × × ✓ × ✓ × × ✓ ✓ ×

⇒ 11 page faults



Q4 continued...

e) Second Chance

3 5 4 3 5 6 | 2 5 3 3 4 2 | 5 4 2 7 4 7 | 3 => 10 page faults

x x x ✓ ✓ x | x ✓ ✓ x x ✓ | x ✓ ✓ x ✓ ✓ | x

1 1 1	1 1 1	1 1 1	0 0 0	0 0 1	0 0 0	0 0 1
3 5 4	5 4 3	4 3 5	4 3 5	3 5 6	3 5 6	5 6 2
0 1 1	0 1 1	1 1 1	0 0 0	0 0 1	0 1 1	0 0 0
6 2 5	6 5 2	5 2 3	5 2 3	2 3 4	3 4 2	3 4 2
0 0 1	0 1 1	1 1 1	0 0 0	0 0 1	0 1 1	0 1 1
4 2 5	2 5 4	5 4 2	5 4 2	4 2 7	2 7 4	2 4 7
0 0 1						
4 7 3						

f) LFU

3 5 4 3 5 6 2 5 2 3 4 2 5 4 2 7 4 7 3 => 13 page faults

x x x ✓ ✓ x x ✓ ✓ ✓ x x ✓ x x x x x x

Counts

2 : ||||

3 : ||||

4 : |||||

5 : |||||

6 : |

7 : ||

3	3	3	3	3
5	5	5	5	5
4	6	2	4	2

2	2	2	7	3
5	5	5	5	5
4	7	4	4	4

Q5)

a) $2^{35} / 2^{12} = 2^{23}$ blocks

b) 2^{23} bits are needed. $\frac{2^{23}}{2^{3}} = 2^{20}$ bytes are needed. $\Rightarrow$ $\frac{1MB}{4KB} = 256$ blocks is needed for bit vector.

c) $200000 \cdot 256 = \frac{51,200,000 \text{ Byte}}{4096 \text{ Byte}} = 12500$ disk blocks are required.

d) $200000 \cdot 256 = \frac{51,200,000 \text{ B}}{4096 \text{ B}} = 12,500$ disks blocks are req.

e) Meta info: $12500 + 12500 + 256 = 25256$ blocks.

25256 blocks $\cdot 4096 = 103,448,576$ bytes for total metadata.

$200000 \cdot 11000 = 2,200,000,000$. Then, $\frac{103,448,576}{2,200,000,000} \cdot 100 = \underline{\underline{4.7}}\ \%$

f) $2^{23} - 625256 = 7763352$ blocks are free

$2^{35} - ((200000 \cdot 11000) + (4096 \cdot 25256)) = 32056289792$ free bytes.

Q6) a) $2^{36}/2^{12} = 2^{24}$ blocks

$2^{24} \cdot 4 = 2^{26}$ bytes

$2^{26}/2^{12} = 2^{14}$ disk blocks are occupied.

b) 1 block can fit $2^{12}/2^{2} = 2^{10}$ pointers.

1 second level table can fit $2^{10} \cdot 2^{12} = 2^{22}$ bytes = 4MB

First level table can fit $2^{10} \cdot 2^{10} \cdot 2^{12} = 2^{32}$ bytes = 4 GB

- To index 1MB, one top and one second level table is sufficient.

- To index 10MB, three second and one top level table is enough.

- To index 100MB, 25 second level table and one top level table is enough.