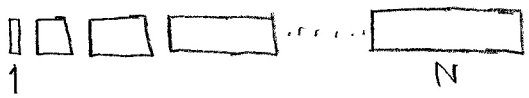
Bartu Atabek
21602229
Section 03

CS 342 Homework #2

16/05/2019

Q1) FCFS



$1 \leq k \leq N$

First wait → 0
Wait second process → N
Wait third process → N+N-1
Wait four process → N+N-1+N-2....+3+2

} N-1 processes

$(N-1)N+(N-2)(N-1)+(N-3)(N-2)+\ldots+2.1 =$

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

$$\text{Average} = \frac{N^2-1}{3} //$$

SJF:

For Process 1 → 0 wait
" " 2 → 1 wait
⋮

For $N-1^{th}$ process → 1+2+3+....N-2 wait
For N^{th} process → 1+2+3+....,N-2+N-1 wait

$$\sum_{k=1}^{N-1}(N-k).k = \frac{1}{6}.(N)(N-1)N$$

$$\text{Average} = \frac{N^2-1}{6} //$$

For RR (q=1 time unit)

Wait in the first round → $1+2+3+4+\cdots N-1 = \frac{N \cdot N-1}{2}$

" " " second " → $1+2+3+4+\cdots N-2 = \frac{(N-1)(N-2)}{2}$

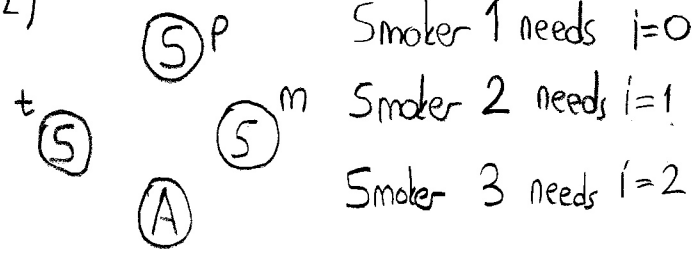
⋮

" " " $(N-1)^{th}$ round → 1

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

Average: $\frac{1}{6}N^2-1$ //

Q2)



Smoker 1 needs i=0
Smoker 2 needs i=1
Smoker 3 needs i=2

```
Semaphore item[3];
Semaphore isEmpty;

//smokers                          //agent
while(1) {                         while(1) {
    //wait items[i]                    wait(isEmpty);
    wait(items[i]);                    int k = rand()%3;
    signal(isEmpty);                   signal(items[k]);
    //smoke                        }
}
```

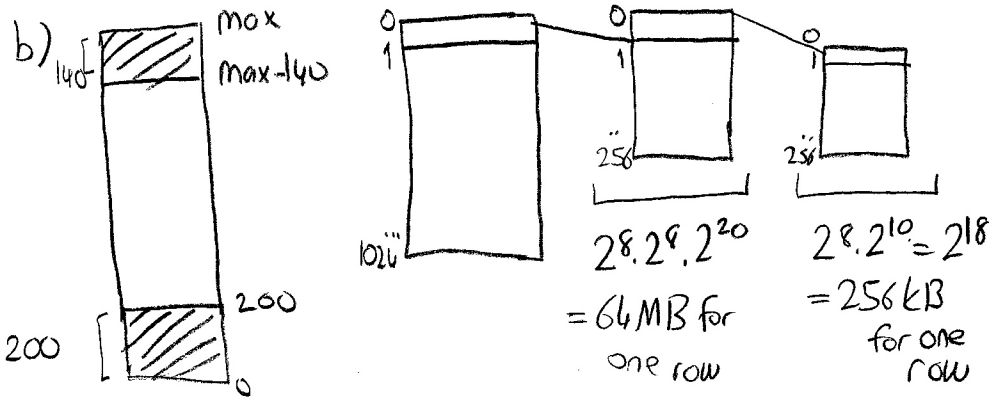
Q3)

a) 36 bits 3 level paging (10,8,8,10)

10	8	8	10

page size = 2^{10} = 1024 bytes = 1KB

b)



$2^8 \cdot 2^9 \cdot 2^{20}$ = 64MB for one row

$2^8 \cdot 2^{10} = 2^{18}$ = 256kB for one row

Q3) (cont'd)

* For the 200 MB space: $\frac{200}{3} = 4 \rightarrow$ second level pgs

$\Rightarrow 64 \cdot 3 + 8 = 200$ MB

$\Rightarrow 256 \cdot 3 + 32 = 800$ third level pgs

* For the 140 MB space: $\frac{140}{64} = 3$

$64 \cdot 2 + 12 = 140$ MB $\rightarrow$ 3 second level pgs

$\Rightarrow 256 \cdot 2 + 48 = 560$ third level pgs

Overall, 7 second lvl pgs.
1360 third lvl

Q4)

a) FIFO:

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

13 pg faults //

MMMHHMMMHMMMMHHMMHM

b) LRU:

3 3 3	(3) 2	2 2 2	(2)	3
5 5	5 5	(5) 4 4	4	4
(4)	6 (6)	3 (3) (5)	7	7

10 pg faults //

MMMHHMMHHMMHMHHMHM

c) OPT:

3 3 3	3 (3)	4	4	3
5 5	5 5	(5)	7	7
(4)	(6) 2	2	2	2

8 pg faults //

MMMHHMMHHHMHHHMHHM

Q4) (cont'd)

d) R-bit

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

MMMH, HMMH, HMMH, MHMM, HHM

2 → Ø X Ø X O
3 → Ø X Ø + Ø 1
4 → Ø X Ø r Ø r Ø 1
5 → Ø r Ø r Ø r O
6 → Ø X O
7 → Ø 1

11 pg faults //

e) Second Chance:

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

10 pg faults //

MMMHHM, MHHMMH, MHHMHH, M

→ 3 → 5 → 4
1 1

→ 4 → 3 → 5 ⇒ 3 → 5 → 6
0 0 0 0 0 1

→ 3 → 5 → 6 ⇒ 5 → 6 → 2 6 → 2 → 5 2 → 5 → 3 3 → 2 → 4
Ø 0 1 1 0 1 0 1 1 Ø 1
1 1

→ 3 → 2 → 4 2 → 4 → 5 4 → 5 → 7 4 → 5 → 7 5 → 7 → 3
0 0 0 1 1 1 Ø 0 1
1

f) LFU:

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

12 pg faults //

MMM H H MM H H H MMH M H MMMM

Q5)

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

b) 2^{20} bits for blocks $\frac{2^{20}}{2^{12}} = 2^{8}$ block for the bitmap //
2^{20} bytes for blocks

Q5) (cont'd)

c) $2\times10^5 \times 2^8 = 5\times10^4 \times 2^{10} = 5\times10^4$ KB (Block size = 4KB)

$\frac{5\times10^4}{4} = 12500$ disk blocks are required //

d) $\frac{2\times10^5 \times 2^{18}}{2^{22}} = 12500$ blocks //

f) 600.000 is approximately $\approx 6.2^{20}$

free blocks = $(8-6)\,2^{10} - 2^8$

$\Rightarrow 2.2^{20} - 2^8 - 12500$ blocks free. $\approx 2.2^{10}$ //

free bytes = $2.2^{20}.4.2^{20} = 8.2^{30} = 8$ GB //

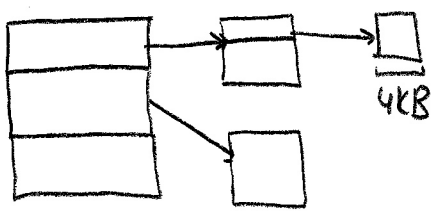
e) $\frac{2^8 + 12500}{6\times10^5 + 2^8 + 12500} \leq 0,02 \Rightarrow 2\%$ //

Q6)

a) $\frac{64\,GB}{4KB} = 11.2^{20}$ blocks

$16\times4\times2^{20} = 2^{26}$ bytes $\Rightarrow \frac{2^{26}}{2^{12}} = 2^{14}$ blocks //

b)



$\frac{4\,KB}{4} = 2^{10}$ entries per block

$2^{10}.2^{10}.2^{14} = 2^{34} = 4$GB max size //

* In order to index 1MB; 1 second and 1 first level tables are required therefore total of 2 index tables. //

Q6) (cont'd)

b) (cont'd)

* In order to index 10 MB 3 second and 1 first level tables therefore 4 total index blocks.

* In order to index 100 MB = 25 second level and 1 first level table therefore 26 total index blocks are required.

c) by using 2 levels $\Rightarrow 2^{32}$ bytes

" " 3 " $\Rightarrow 2^{32} \cdot 2^{10} = 2^{42}$ bytes = 4TB maximum file size. //

d) By using 10 direct & 3 indirect pointers the maximum filesize is:

$$\Rightarrow \underbrace{10 \times 4KB}_{\text{direct}} + \underbrace{2^{20} \times 4KB}_{\text{1st layer}} + \underbrace{2^{20} \times 4KB}_{\text{2nd layer}} + \underbrace{2^{30} \times 4KB}_{\text{3rd layer}}$$

$$= 4KB + 4MB + 4GB + 4TB = 4 + 4096 + 4,194,304 + 4,294,967,296$$

$$= 4,299,165,700 \text{ KB} \approx 4004 \text{ TB} //$$

* Offset = 0 requires 2 disk accesses (1 for inode 1 for data) //

* Offset = 1000000 $\rightarrow 2^{22} > 1,000,000 \Rightarrow$ in first layer 1 inode + 2 data access = 3 access //

* Offset = 100000000 $\rightarrow 2^{32} > 100000000 \Rightarrow$ inside 2nd layer 1 inode + 3 data

= 4 disk accesses //