Assignment 3

By Eldor Bekpulatov (eb654)

Question 1

a) N_p $-$ $number\ of\ lottery\ tickets$
   p $-$ $process\ number$
   T $-$ $total\ number\ of\ processes$
   $Condition:\ 0 \leq p < T$
   P_p $=$ $Probability\ that\ process\ p\ is\ will\ next\ be\ selected\ to\ run\ by\ a\ scheduler$

$$P_p = \frac{N_p}{\sum_{k \in [0...T]} N_k} = \frac{N_p}{\sum_{k=0}^{T} N_k}$$

b) Minimum amount of time that a process might be delayed before it has an opportunity to execute is 0.01 seconds, because a process can be selected to be executed randomly.

   Maximum amount of time that a process might be delayed before it has an opportunity to execute is infinite, because although very unlikely, a process can never be selected to be executed next. It would have to be a very unlucky process, but it is possible. Since, I don't think this is an acceptable answer, I propose we treat the expected time to wait as the maximum amount of time a process might be delayed before it is executed.

   To calculate this, lets treat a simple case where we roll a fair die, and calculate the expected number of times we have to roll to get x where $1 \leq x \leq 6$. We can see that $E(X = x) = \frac{1}{p}$ where p is the probability of $X = x$ happenin and $X = x$ simply means rolling an x. Given this, we can say that the estimated number of times of rolling a 1 is $E(X = 1) = 6$, because for a fair die p=⅙. Thus, we can say the expected time to wait for a process p to be scheduled is $\frac{1}{P_p}$ where P_p is given in part a.

$$E(X = p) = \frac{1}{P_p} = \frac{\sum_{k=0}^{T} N_k}{N_p}$$

c) If p_1 is waiting for p_2 where the priority of p_1 is higher than the priority of p_2, then the scheduler should allow for p_2 to inherit the priority of p_1. Here, p_1 would still be limited by the priority of p_2, but the priority of p_2= priority of p_1. Then, $p_1$would be limited by the priority of p_1, which is what we want.

Question 2

Given:

- 1 surface per disk
- 200 tracks per surface
- 500 blocks per track
- 512 bytes per block
- 3600 RPM
- 1 msec per track seek

1) $Capacity \ = \ \frac{1\ surface}{1\ disk} \times \frac{200\ tracks}{1\ surface} \times \frac{500\ blocks}{1\ track} \times \frac{512\ bytes}{1\ block} = \ 51,200,000\ \frac{bytes}{disk}$

2) $Max\ seek\ time\ = \ 199\ tracks\ \times \frac{1\ msec}{1\ track}\ = \ 199\ msec$

$Max\ Rotation\ Time\ = \ 499\ blocks\ \times \frac{1\ track}{500\ blocks} \times \frac{1\ rev}{1\ track} \times \frac{1\ min}{3600\ revs} \times \frac{60\ sec}{1\ min} \times \frac{1000\ msec}{1\ sec} = \ 16.$

$Max\ Latency\ = Max\ Seek\ Time\ +\ Max\ Rotation\ Time\ = 199\ msec\ +\ 16.6\ msec\ =\ 215.6$

3) $Max\ Transfer\ Rate\ = (\frac{3600\ revs}{1\ min} \times \frac{1\ min}{60\ sec} \times \frac{1\ track}{1\ rev}) \times (\frac{500\ blocks}{1\ track} \times \frac{512\ bytes}{1\ block})\ =\ 15,360,000$

Given:

- Capacity tolerance ¼ of initial
- Read and Write speed 2X

4) We can make the disk have ¼ of the capacity by decreasing the number of tracks per surface to 50 tracks. This will reduce the max latency of the read and writes because our max seek time will reduce by 4 times.

5) $Capacity_{new}\ = \ \frac{1\ surface}{1\ disk} \times \frac{50\ tracks}{1\ surface} \times \frac{500\ blocks}{1\ track} \times \frac{512\ bytes}{1\ block} = \ 12,800,000\ \frac{bytes}{disk}$

6) $(Max\ seek\ time)_{new} = \ 49\ tracks\ \times \frac{1\ msec}{1\ track}\ = \ 49\ msec$

$Max\ Rotation\ Time\ = \ 499\ blocks\ \times \frac{1\ track}{500\ blocks} \times \frac{1\ rev}{1\ track} \times \frac{1\ min}{3600\ revs} \times \frac{60\ sec}{1\ min} \times \frac{1000\ msec}{1\ sec} = \ 16.$

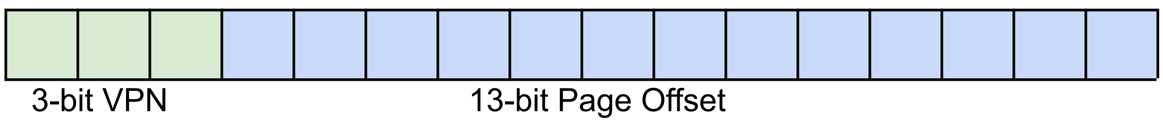
$(Max\ Latency)_{new}\ = (Max\ Seek\ Time)_{new}\ +\ Max\ Rotation\ Time\ =\ 65.6\ msec$

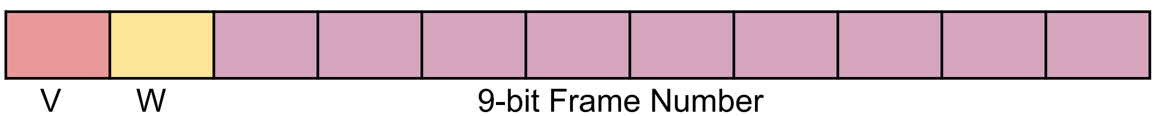
7) Both read and write requests will have their max latency reduced by 3.28 times compared to the previous design, because $\frac{215.6\ msec}{65.6\ msec} = 3.28$

8) Max rotation time stays the same, thus the max transfer rate will not be different from the one calculated in part a.

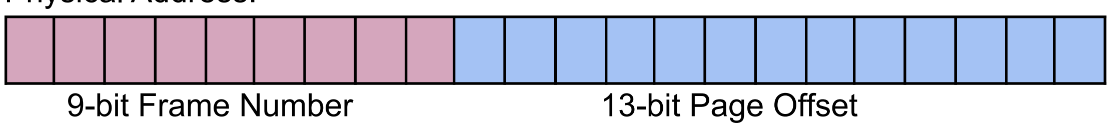
Question 3

Given:





a) Physical Address:



$$Physical\ Size = 2^{22}\ bytes = 4,194,304\ bytes = 4096\ Kbytes$$

b)

Virtual Address	Valid	Physical Address	Writable
`0x1234`	y	`0x07234`	n
`0x4321`	y	`0x10321`	n
`0x8888`	n	-	-

c)

Scheme	All page numbers of mapped pages (7)	# page faults (7)	# page faults (10)
FIFO	0 1 2	5	7

LRU	0 1 7	4	6
OPT	0 1 2	4	5