CS342 - Spring 2019
Homework #2

Assigned: **May 11, 2019.**
Due date: **May 16, 2019, 23:55.**

You will submit this homework through Moodle.

Q1) Assume there are N processes in a ready queue in decreasing order with respect to their lengths (cpu burst length). The length of process k is k time units (1 ¡= k ¡= N). Process N is head and process 1 is tail. Compute average waiting time (averaged over all processes) for each of the following scheduling algorithms: FCFS, SJF, RR(q=1 time unit).

Q2) Assume a cigarette requires three ingredients (items) to smoke: Tobacco (t), Paper (p) and a Match (m). Assume there are 3 smokers around a table, each of whom has an infinite supply of one of the three ingredients (items) one smoker has an infinite supply of tobacco, another has an infinite supply of paper, and the third has an infinite supply of matches. Assume there is also a non-smoking agent which has also an infinite supply of these items. The agent enables the smokers to make their cigarettes by randomly selecting and putting two items (out of three items) on the table. Then the smoker having the missing item will take the items from the table (in this way will make the table empty), will make his cigarette, and will be able to smoke for a while. When table becomes empty, agent again chooses two items in random and places them on the table. Another smoker can now smoke (or maybe the currently smoking smoker will take those items again and start smoking again after it has finished its current smoking). This process continues forever. Synchronize the agent and 3 smokers by use of semaphores to act in this way.

Q3) Assume we have the following address division scheme in a system where virtual addresses are 36 bits and 3 level paging is used: (10, 8, 8, 10). That means offset (displacement) part is 10 bits. Answer the following questions: a) What is the page size? b) Assume a program uses 200 MB at the beginning of the virtual memory (from address 0 upwards) and 140 MB at the end of the virtual memory (from max address downwards). How many second level page tables are used? How many third level page tables are used?

Q4) Consider the following page reference string of a process. Assume the process has 3 frames.

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

Find the number of page faults and also which references cause a page fault for the following page replacement algorithms: a) FIFO; b) LRU; c) OPT; d) R bit algorithm (assume in case of tie, smaller page number is removed, and R bits are cleared after every 4 references); e) Second chance (assume reference bits are cleared after every 6 references), f) LFU.

Q5) Assume we have a disk of size 32 GB. The block size is 4 KB. There are 200000 files in the disk and the average file size is 11000 bytes (i.e, on the average a file has 3 data blocks allocated). FCB size is 256 bytes. Directory entry size is fixed and is also 256 bytes. Assume a bitvector is used to keep track of the free blocks on the disk. You can assume that we have a single directory (root directory) in the system. Answer the followings questions based on this information.

a) How many blocks are there in the disk?
b) How many disk blocks are required to keep the bitvector (bitmap)?
c) How many disk blocks are required to keep the FCBs? Note that a disk block can keep several FCBs.
d) How many disk blocks are required to keep the directory information? Note that a disk block is large enough to keep several entries.
f) How much disk space (in blocks and also in bytes) is free approximately?
e) Consider the used part of the disk. What percent of it is meta-information?

Q6) Assume a disk that has 64 GB storage capacity. The filesystem on the disk uses 4 KB blocks and pointer size is 4 bytes. Answer the following questions.

a) Assume FAT is used. How many disk blocks is occupied by FAT for this disk assuming entry size is 4 bytes.
b) If indexed allocation scheme is used with two-level index structure, what is the maximum file size? How many index blocks are required for files A, B, C of sizes 1 MB, 10MB, 100 MB, respectively?
c) If indexed allocation scheme with 3 levels used, what is the maximum file size?
d) If indexed allocation scheme defined by Unix/Linux is used (mixed scheme), what is the maximum file size assuming an inode can store 10 direct pointers. For a file of size 1 GB, how many disk accesses would be required to access a byte at offset 0, offset 1000000, and offset 100000000 of the file. Find your answer for each offset separately. Assume nothing is cached.