CS342 Operating Systems - Spring 2023
Exercise Homework #2

Exercise version of homework 2. Not to be graded. Solution is available.

Assigned: May 28, 2023.
Due date: NA Document version: 1.1

Q1. Consider a system consisting of 4 processes and a single resource type. The current state of the Maximum Demand (Claim) and Allocation matrices are as follows:

Max Demand	Allocation
P1 3	P1 1
P2 2	P2 1
P3 9	P3 3
P4 7	P4 2

What is the minimum number of resource instances that need to be available for this state to be safe?

Q2. Consider the following situation in a system where there are 5 processes and 3 resource types: A, B, C. No deadlock avoidance or prevention is applied. Initially, there exist 10 A, 6 B, and 4 C in the system (without allocations yet). Is there a deadlock at the moment? Prove your answer.

	Alloc			Request		
	A	B	C	A	B	C
P1	1	0	0	0	0	1
P2	3	0	1	0	4	0
P3	0	2	2	3	1	2
P4	1	0	1	2	3	0
P5	2	1	0	3	3	3

Q3. Assume we have system that is using single-level paging. Assume page table for a process is always in the memory. a) If a physical memory access takes 200 ns, what is the effective access time to memory (EAT) without TLB? b) Assume we have a TLB used. The TLB search takes 20 ns, no matter it is a hit or miss. If hit rate is 80%, what is the effective access time to memory? c) If two level paging would be used, what would be your answer for questions a) and b)?

Q4. Assume a system is using two-level paging, 32-bit virtual addresses and 40-bit physical addresses. A virtual address is divided into three pieces as follows: [10, 10, 12]. That means, the first 10 bits are index to the first-level table. Offset is 12 bits. Assume each page table entry (first-level or second-level) is 4 bytes.
a) How many bits in a page table entry are used to store a frame number?

b) How much memory is consumed by first and second level page tables for a process that has 256 KB of virtual memory used, starting at address `0x00000000`.
c) How much memory is consumed by first and second level page tables for a process that has a code segment of 48 KB at virtual address `0x01000000`, a data segment of 10 MB starting at virtual address `0x80000000` and a stack segment of 64 KB starting at virtual address `0xf0000000`. Assume for this question that stack segment also grows upward (towards higher addresses).

Q5. Consider the following page reference string of a process.

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

Assume the process has 3 frames that it can use, all empty initially.
a) Assume second chance (i.e., clock) algorithm is used as page replacement algorithm. Assume reference bits (R bits) are cleared after every 5 references (i.e., some time between every 5th and 6th reference). Show the memory state (the pages in memory and their R bit values) after each page reference. Also indicate which reference causes a page fault. Assume after a page fault, when the new page is loaded, its R bit is set to 1. b) Solve the same question for Optimal algorithm.

Q6. Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4,999. The drive is currently serving a request at cylinder (track) 2150, and the previous request was at cylinder 1805. The queue of pending requests, in FIFO order, is as follows:

2069 1212 2296 2800 544 1618 356 1523 4965 3681

Starting from the current head position, what is the total distance (in cylinders/tracks) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms? a) FCFS, b) SCAN, c) C-SCAN.

Q7. A disk has the following parameters:

Size: 1 TB
RPM: 15000
avg seek time: 3 ms
max transfer rate: 50 MB/s.

a) Assume block size 4 KB. What is the I/O time to read one random block from this disk? How many such transfers can we complete per second? What is the I/O data rate (i.e., throughput).

b) Assume we will read 1000 blocks, that are contiguous on the disk, sequentially. How many such transfers can we complete per second? What is the I/O data rate (i.e., throughput).

Q8. Consider a file system that uses inodes to represent files. Disk blocks are 8 KB in size, and a pointer to a disk block requires 4 bytes. Assume in an inode we have 10 direct disk block pointers, one single indirect pointer, one double indirect pointer, and one triple indirect pointer. That means, combined index allocation scheme is used to keep track of the blocks allocated to a file. a) What is the maximum size of a file that can be stored in this file system? b) How many second level index blocks are required for a file X of size 4 GB. c) If nothing, except the inodes, is cached in memory, how many disk block accesses are required to access a byte i) at offset 2^16, ii) at offset 2^21, iii) at offset 2^27 of the file X?