COSC 450 Midterm Practice

Note 14:

1. A computer system generates a 32-bit virtual address for a process. This system has 8 GB RAM and page size is 4KB.

   a) If each entry in the page table needs 64 bits per entry, calculate the possible size of the page table by bytes.

   b) Page frame number information for each page must be saved in the page table. How many bits does it need to save page frame number information?

2. A Computer with a 64 bit virtual address use two-level page table. Virtual addresses are split into a 25 bit top-level page table field, a 25 bit second-level page table field and an offset. (need check)

   - What is the size of each page?
   - How many possible page tables are there?
   - How many possible pages are there?
   - If the system supports 8 GB Ram, how many bits need to be reserved for saving page frame number in the each of page table entry?

3. Page table structure

   a) What is the main motivation of multilevel page table?

   b) What is the main motivation of hased page table?

   c) How many entries are there in inverted page table?

Note 15:

4. How many page faults occur for following reference string with three page frame for each of following page replacement algorithm? Let's assume page frames are initially empty.
   Reference string: 5, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 5, 0, 1
   a) LRU:

   b) Optimal:

5. (5 pt.) A system use the paging for managing virtual memory. The system has four page frames. The time of loading, time of last access, and the reference bit **R**, and modified bit **M** for each page are as shown below (the times are in clock ticks):

Page	Loaded	Last referenced	R	M
0	220	265	0	0
1	245	255	0	1
2	115	270	1	0
3	126	280	1	1

a) Which page will FIFO (First in First Out) replace?

b) Which page will NRU (Not Recently Used) replace?

c) Which page will LRU (Least Recently Used) replace?

d) Which page will Second chance replace?

Note 16:

6. What is Belady's Anomaly?
    1.
    2.
    3.
    4.

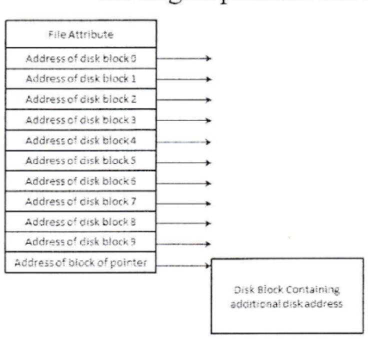
7. (10 pt.) Page size is one of most important design issue in the operating system. We can mathematically analyze page size based on following assumptions:
    - S: average size of process (byte)
    - P: the size of page (byte)
    - E: Each page entry requires (byte)
    - 50% of memory in the last page of the process is wasted due to internal fragmentation

    a. Define the total overhead function based on page size P.

    b. Find the optimal page size formula based on the total overhead (by minimize the total overhead)

Note 17:

8. (2 pt.) LINUX like system use i-node to maintain the file system. Attributes and block addresses are saved in i-node. One problem with i-nodes is that if each one has room for a fixed number of disk addresses, what happens when a file grows beyond this limit? One solution is to reserve the last disk address not for a data block, but instead for the address of block containing more disk-block addresses as shown following picture.
Picture shows that i-node contains 10 direct addresses and these were 4 bytes each. A block size is 2 KB. If a file use i-node and one extra block to save block information, what world the largest possible file size could be?



9. Lets assume that a LINUX system use bitmap for maintain free disk block information. Let assume the bitmap was completely lost due to the crash. Is it possible to recover bitmap? If possible, discuss your algorithm to recover the bitmap. If not, discuss why.

10. In the file system, two methods are widely used to keep track of free blocks: a linked list and a bitmap. Let's say a block size is 8-KB and 64-bit disk block number in a file system.
    a. How many maximum blocks are needed for keep track 128-GB disk with linked list?

    b. How many blocks are needed for keep track of 128-GB disk with bitmap?

    c. What is the maximum disk size supported by the operating system?

Note 18:

11. About Log-Structured File System
    a. Log-Structured File system can be applied based on the assumption. What is this assumption?

b. Linux use i-node for saving blocks information for a file. To open a file, operating system checks the directory for the file to get i-node number. Since i-nodes are located in special location, operating system does not need search for i-node. In LSF (Log-Structured File) system, i-node is not located in specific location. Briefly discuss how LSF operating system could access a file.

12. User B and C are sharing a file by saving shared file information in each directory. Discuss the problems with this and solutions to it.

13. (2 pt.) One way to use contiguous allocation of the disk and not suffer from holes is to compact the disk every time a file is removed. Since all files are contiguous, copying a file requires a seek and rotational delay to read the file, followed by the transfer at full speed. Writing the file back requires the same work. Assuming a seek time of 5 msec, a rotation delay of 4 msec, a transfer rate of 8MB/sec, and an average file size of 8 KB, how long does it take to read a file into main memory and then write it back to disk at a new location? Using these numbers, how long would it take to compact half of a 16 GB disk.

14. A system uses bitmap to keep track of free blocks. Let's say that a block size is 2KB. The system uses 2^{12} blocks for bitmap. What is the total disk size?

Note 19:

15. (2 pt.) Free disk space can be kept track of using a free list or bitmap. Disk addresses requires D bits. For a disk with B blocks, F of which are free, state the condition under which the free list uses less space than the bitmap. For D having the value 16 bits, express your answer as a percentage of the disk space that must be free.

16. Briefly discuss physical and logical dumps.

Note 20-22:

17. (10 pt.) Consider a system with 5 processes (P_0 ... P_4) and 3 resources types (A, B, C) with E = {10, 5, 7}. Resource-allocation state at time t_0 shows in table.

Process	Allocated			Max Need			R		
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	7	5	3	7	4	3
P_1	2	0	0	3	2	2	1	2	2
P_2	3	0	2	9	0	2	6	0	0
P_3	2	1	1	2	2	2	0	1	1
P_4	0	0	2	4	3	3	4	3	1

A = (3, 3, 2)

a) Will a request of (1, 0, 2) by P_1 be granted? (it is not yes/no problem)

b) Will a request of (3, 2, 0) by P_4 be granted? (it is not yes/no problem)

c) Will a request of (3, 3, 0) by P_4 be granted? (it is not yes/no problem)

18. Discuss each of followings.

a. What are four necessary conditions for a deadlock
   1.
   2.
   3.
   4.

b. Four strategies for detailing with a deadlock
   1.
   2.
   3.
   4.

c. Why are segmentation and paging sometimes combined into one scheme for memory management?

19. (5 pt.) A system need maintain four matrix for deadlock detection: E, A, C, R

a) A system has three processes and four kinds of resources. Following shows snapshot of matrix A, C, R and E at time T_0

$$E = [4\ \ 2\ \ 3\ \ 1],\ C = \begin{bmatrix} 0010 \\ 2001 \\ 0120 \end{bmatrix},\ R = \begin{bmatrix} 2001 \\ 2111 \\ 2100 \end{bmatrix},\ A = [2\ \ 1\ \ 0\ \ 0]$$

Show whether deadlock situation or not based on deadlock detection algorithm (its not "yes", "no" question)

b) A system has five processes and three kinds of allocatable resources. At a certain time, matrix A, C, and R at a certain time are

A = (0, 0, 0)

$$E = [7\ \ 2\ \ 6],\ C = \begin{bmatrix} 010 \\ 200 \\ 303 \\ 211 \\ 002 \end{bmatrix}, R = \begin{bmatrix} 000 \\ 202 \\ 000 \\ 100 \\ 002 \end{bmatrix}$$

Show whether deadlock situation or not based on deadlock detection algorithm (its not "yes", "no" question)

11. (10 pt.) A system has four processes and five allocatable resources. The current allocation and request recourse for each process are as follows:

	Allocated					Need More					Available				
	R_1	R_2	R_3	R_4	R_5	R_1	R_2	R_3	R_4	R_5	R_1	R_2	R_3	R_4	R_5
P_1	1	0	2	1	1	0	1	0	0	2	0	0	X	1	Y
P_2	2	0	1	1	0	0	2	1	0	0					
P_3	1	1	0	1	0	1	0	3	0	0					
P_4	1	1	1	1	0	0	0	1	1	1					

What is the minimum value of X and Y for which this is safe state?
You should describe how logically select value of X and Y .