8/10

COSC 450 Operating System Test #2

11/21/2022

Name: Jung An

1. (15 pt.) 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 2-KB and 32-bit disk block number in a file system.
   a. How many maximum blocks are needed for keep track 128-GB disk with linked list?

   $2^{11} / (2^5/2^3) = 2^9 - 1 = 511$ block info

   $128GB/2KB = 2^{37}/2^{11} = 2^{26}$ blocks

   $2^{26}/511 = 131328.501$ blocks

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

   2^{26} blocks

   $2^{26}/2^{11} \cdot 2^3 = 2^{12}$ blocks

   c. What is the maximum disk size supported by the file system

   $2^{32}/2^3 \cdot 2^{11} = 2^{40} = 1TB$

   ? bit to byte

   X

-4

2. (5 pt.) About Log-Structured File System
   a) Log-Structured File system can be applied based on the assumption. What is this assumption?

   X It is based on assumption that the files are in sequence block of information ??

   I don't have a clue on this question

   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.

   First, it access i-node pertaining the file ???

   X it reads the reference to the file and access it by refering to the file reference

-3

3. (10 pt.) 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.

   2^{32} bit / $2^{12} \cdot 2^{3}$ bit = 2^{17} page

   $2^{17} \cdot 64$ bit = $2^{17} \cdot 2^{6} = 2^{23}$ bit = 2^{20} byte = 1MB

   X

   -4

   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?

   $8GB/4kB = 2^{3} \cdot 2^{30} / 2^{2} \cdot 2^{10} = 2^{21}$

   21 bits

4. (10 pt.) Let's 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.

   Yes it is possible. It would be more of creating new bit map with new bitmap, initially set all to be 0 (unused block) then loop through each block and mark 1 if the block has been used.

   -5

   X

5. (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)

New snap

$$C = \begin{pmatrix} 0 & 1 & 0 \\ 3 & 0 & 2 \\ 3 & 0 & 2 \\ 2 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix} \quad R = \begin{pmatrix} 7 & 4 & 3 \\ 0 & 2 & 0 \\ 6 & 0 & 0 \\ 0 & 1 & 1 \\ 4 & 3 & 1 \end{pmatrix} \quad A = (2,3,0)$$

Yes it will be granted

$$(2,3,0) \xRightarrow{P_1} (5,3,2) \xRightarrow{P_3} (7,4,3) \xRightarrow{P_4} (7,4,5) \xRightarrow{P_0} (7,5,5) \xRightarrow{P_2} (10,5,7)$$

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

New Snap

$$C \begin{pmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 3 & 0 & 2 \\ 2 & 1 & 1 \\ 3 & 2 & 2 \end{pmatrix} \quad R \begin{pmatrix} 7 & 4 & 3 \\ 1 & 2 & 2 \\ 6 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad A = (0,1,2)$$

Yes it will be granted

$$(0,1,2) \xRightarrow{P_3} (2,2,3) \xRightarrow{P_1} (4,2,3) \xRightarrow{P_4} (7,4,5) \xRightarrow{P_1} (7,5,5) \xRightarrow{P_2} (10,5,7)$$

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

New snap

$$C \begin{pmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 3 & 0 & 2 \\ 2 & 1 & 1 \\ 3 & 3 & 2 \end{pmatrix} \quad R \begin{pmatrix} 7 & 4 & 3 \\ 1 & 2 & 2 \\ 6 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \quad A = (0,0,2)$$

No it will not be granted

(0, 0, 2) unsafe (deadlock)

6. (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?

Page 1 ✗ (last refered the oldest)

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

Page 1

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

Page 0

d) Which page will Second chance replace?

Page 3

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.

$$\text{overhead} = \frac{S \cdot E}{P} + \frac{1}{2}$$

✓

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

$$P = \frac{S \cdot E}{P} + \frac{1}{2}$$

$$\cancel{P = \frac{2S \cdot E + P}{P}},$$

$$1 = \frac{2S \cdot E + P}{P2}$$

$$P^2 = 2S \cdot E + P$$

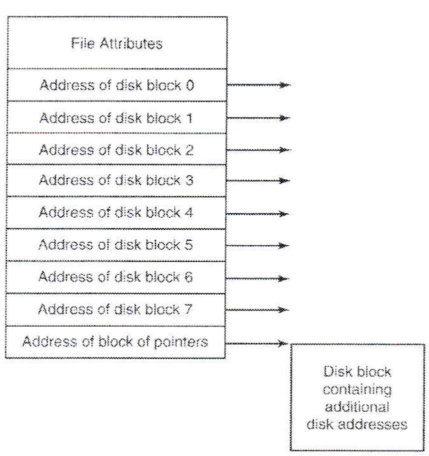
$$P = \sqrt{2S \cdot E + P} \quad ???$$

I know the answer is $\sqrt{2SE}$ but can't figure out the calculation

8. (5 pt.) Discuss each of followings.
   a) What are four necessary conditions for a deadlock
      1. Mutral exclusion
      2. hold and wait
      3. No premptive
      4. Circular wait
   b) Four strategies for dealing with a deadlock
      1. ignore
      2. deetect and recovery
      3. premptive with dynamic allocation
      4. Attack one of deadlock condition
   c) Why are segmentation and paging sometimes combined into one scheme for memory management?

      Due to amount of space physical memory can table. If big segmentation it will waste memory space and if the virtual paging is too big, it does not fit in physical memory

9. (10 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 8 direct addresses and these were 16 bytes each. A block size is 2 KB. If a file use i-node and one extra block to save block information, what world be the maximum file size?



$2KB / 16 bytes = 2^{11} / 2^4 = 2^7 = 128$

$128 + 8 = 136$

$136 \cdot 2KB = 272 KB$

10. (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)

$A = (2,1,0,0) \overset{\substack{(2,1,0,0) \\ P_2}}{\Rightarrow} (2,2,2,0)$ deadlock

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)

$A = (0,0,0) \overset{\substack{(0,0,0) \\ P_0}}{\Rightarrow} (0,1,0) \overset{\substack{(0,0,0) \\ P_2}}{\Rightarrow} (3,1,3) \overset{\substack{(2,0,2) \\ P_1}}{\Rightarrow} (5,1,3) \overset{\substack{(1,0,0) \\ P_3}}{\Rightarrow} (7,2,4) \overset{\substack{(0,0,2) \\ P_4}}{\Rightarrow} (7,2,6)$

it is not a deadlock

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 .

$$C = \begin{pmatrix} 1 & 0 & 2 & 1 & 1 \\ 2 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \quad R = \begin{pmatrix} 0 & 1 & 0 & 0 & 2 \\ 0 & 2 & 1 & 0 & 0 \\ 1 & 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \quad A = (0\ 0\ X\ 1\ Y)$$

First select lowest allowed case in this case is P_3

assume $A = (0, 0, 1, 1, 1)$ (same as R for P_3) after completion

$A = (1, 1, 2, 2, 1)$ however this doest fit next lowest P_0 So we increase Y by 1 $(1, 1, 2, 2, 2) \overset{P_0}{\Rightarrow} (2, 1, 4, 3, 3) \Rightarrow$ which then fits all

$X = 1$, $Y = 2$

12. (5 pt.) To solve deadlock problem, we can attack one of necessary deadlock condition. Mr. Computer provide following solutions. What necessary deadlock condition he attack and what are problem with his solution?

Solution 1:

- Allows a process to request resources only when the process has none
- To get a new resource, first, release all the resources currently holds and request all at time same time

Attack hold and wait

Hold and wait solution has a disadvantage of Starvation

Solution 2:

- A process can hold only one resource.
- If a process need second resource, the process need release the first one.

Attack Circular wait.

This will become problem if process request two resource at once