COSC 450 Operating System Test #2-1

11/06/2019

Name:_____.

1.

Since 1 block is 2KB, and 4 Byte per block address, it can save $2 \times 2^{10} / 4 = 2^9 = 512$ block information

Total = 512 + 10 = 522 block information.

Since a block size is 2KB, largest file will be 2KB × 522 =1044 KB

2.

a) Page 2
b) Page 0
c) Page 1
d) ? Page 0

3.

a.

$$Total\ Overhead(P) = Average\ page\ table\ size + the\ wasted\ memory\ in\ th\ last\ page\ of\ process$$
$$= \frac{S}{P} \times E + \frac{P}{2}$$

b.

$$Overhead'(P) = -\frac{SE}{P^2} + \frac{1}{2} = 0$$

$$P = \sqrt{2SE} : optimal \text{ page size}$$

4.

a)

Size of bit-map = $2 \times 2^{10} \times 2^{16}$ Byte = $2 \times 2^{10} \times 2^{16} \times 8$ Bit = 2^{30} bit.
There are 2^{30}blocks
Total disk size = $2^{30} \times 2 \times 2^{10} = 2^{41}$ Byte =2 TB

b)

# of block information per block = (block size / bit used for a block #) – 1
= $8 \times 2 \times 2^{10}$ / 32 bit = $2^9 - 1 = 512 - 1 = 511$
total # block number = 511×2^{28}
total disk size = $511 \times 2^{28} \times 2 \times 2^{10} = 511 \times 2^{39}$ Byte = about $2^8 \times 2^{40}$ Byte = 256 TB.

5.

a)

Process	Allocation			Need			Available		
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	7	4	3	2	3	0
P_1	3	0	2	0	2	0			
P_2	3	0	2	6	0	0			
P_3	2	1	1	0	1	1			
P_4	0	0	2	4	3	1			

A = (2, 3, 0) – P_1 – (5, 3, 2) – P_3 – (7, 4, 3) –P_0 –(7, 5, 3)- P_2 –(10, 5, 5) –P_4 –(10, 5, 7)

b)

Process	Allocation			Need			Available		
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	7	4	3	0	1	2
P_1	2	0	0	1	2	2			
P_2	3	0	2	6	0	0			
P_3	2	1	1	0	1	1			
P_4	3	2	2	1	1	1			

A = (0, 1, 2) – P_3 – (2, 2, 3) – P_1 – (4, 2, 3) –P_4 –(7, 4, 5)- P_0 –(7, 4, 5) –P_2 –(10, 5, 7)

c)

Process	Allocation			Need			Available		
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	7	2	3	0	0	2
P_1	2	0	0	1	2	2			
P_2	3	0	2	6	0	0			
P_3	2	1	1	0	1	1			
P_4	3	3	2	1	0	1			

non- of process can check with algorithm. Unsafe.

6.

a)

- Mutual exclusion
- Hold-and Wait
- No preemption
- Circular wait

b)

- Ignore
- Detection and recovery
- Avoidance with dynamic allocation
- By attacking one of necessary deadlock condition

c)

a segment is a logical entity.

- If the segments are large, to keep them in the physical memory might be wasting memory space.
- If a segment's virtual space is larger than physical space, it is not even possible to keep them in the physical memory.

7.

- Maintains an internal array M that keep track of the state of memory.
- M has as many as virtual memory pages *n*.
- Top *m* entries contain all the pages currently in the memory (page frames).
- Bottom *n* - *m* entries contains all the pages that have been referenced once but have been page out and are not currently in memory

8.

a.

1 block = $2 \times 2^{10} \times 8$ = 16384 bit, 16384 /32 = 512 -1 = 511 block numbers /block
128 GB = $(128 \times 2^{30})/(2 \times 2^{10}) = 2^{26}$ blocks (number of blocks in 128 GB)
Needs (2^{26}) /511 = 131328.5= 131229 blocks

b.

128 GB = $(128 \times 2^{30})/(2 \times 2^{10}) = 2^{26}$ blocks (number of blocks in 128 GB)
System need one bit per block, the bit map size is 2^{26} bits.
2^{26} bits = $(2^{26})/ 8 = 2^{23}$Byte
Since each block size is 2KB, need $(2^{23})/ (2 \times 2^{10})$= 4096 blocks to save free block information.

c.

- Since this system use 32bit disk block number, this system support 2^{32} blocks
- Maximum disk size = $2^{32} \times 2 \times 2^{10}$ Byte =8×2^{40} = 8 TB

9. (5 pt.) About Log-Structured File System

a) files are cached in the RAM when it is opened.

b)

- In LSF, each i-node is not at a fixed location; they are written to the log.
- LFS uses a data structure called an **i-node map** to maintain the current location of each i-node.
- Opening a file consists of using the map to locate the i-node for the file.

10.

a)

A=(2,1,0,0) -> (2,2,2,0) Deadlock

b)

P_1 P_3 P_2 P_4 P_5

A=(0,0,0) $\rightarrow$ (0,1,0) $\rightarrow$ (3,1,3) $\rightarrow$(5,1,3) $\rightarrow$ (7,2,4) $\rightarrow$ (7,2,6)

11.

Sol) since P_1 need 3 R_5 in total minimum Y should be >=2.

since R_1=0, R_2 =0, only P_4 can be selected based on A with X >=1

- with X=1, Y=2 A=(0 0 1 1 2)

after P_4, A = (0 0 1 1 2) + (1 1 1 1 0) = (1 1 2 2 2)

after P_1, A = (1 1 2 2 2) + (1 0 2 1 1) = (2 1 4 3 3)

after P_3, A = (2 1 4 3 3) + (1 1 0 1 0) = (3 2 4 4 3)

after P2, A = (3 2 4 4 3) + (2 0 1 1 0) = (5, 2 5 5 3)

12.

Seek time + rotation delay = 7 + 3 = 10 msec

Average file size = 4×2^{10} Byte = 2^{12} Byte,

Transfer rate = 8MB/sec = 8×2^{20} Byte/sec = 2^{23} Byte/sec

A file with average size can transfer 10 + (2^{12} Byte /2^{23} Byte/sec)$\times 10^3$ = 10.49 msec

Read + write takes 10.49 + 10.49 = 20.98 msec

A file (average size = 4KB) takes 20.98 msec (transfer time)

Half of a 32 GB = 16 GB

Number of files in 16GB = 16GB/ average size of file = (16×2^{30})/ (4×2^{10}) =4×2^{20}

16GB space take $20.98 \times 4 \times 2^{20}$ mec = 87996497.92 msec = 87996.49792 sec = 24.4 hour

13.

a) **Phase 1 :** begins at the starting directory and examines all the entries in it. For each modified file, its i-node is marked in the bitmap. Each directory is also marked and recursively inspected.
b) **Phase 2:** unmarking any directories that have no modified files or directories in them or under them.
c) **Phase 3:** all marked directory is dumped
d) **Phase 4:** all marked files is dumped

14.

Solution 1) Attacking hold and wait, starvation

Solution 2) Attacking circular wait, If a process need two resource at a same time, this solution have problem