

Chapter 4 Assignment & other questions

Q1)

A UNIX file system has 1-KB blocks and 4-byte disk addresses. What is the maximum file size if i-nodes contain 10 direct entries, and one single, double, and triple indirect entry each?

Q2)

What would happen if the bitmap or free list containing the information about free disk blocks was completely lost due to a crash? Is there any way to recover from this disaster, or is it bye-bye disk? Discuss your answers for UNIX and the FAT file system separately.

Q3)

Suppose storage: 16 GB USB drive, Block size: 4KB

Shall we use FAT16 or FAT32? and

How many blocks do we need on the disk for the FAT?

Q4)

Consider the following segment mapping table (SMT)

<i>Segment</i>	<i>Base</i>	<i>Length (limit)</i>
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

What are the physical addresses of the following logical addresses?

- a) <0, 430>
- b) <1, 10>
- c) <2, 500>
- d) <3, 500>
- e) <4, 122>
- f) <6, 1952>

Q5)

Given a logical address field with the following format:

2 bits	16 bits	8 bits
Seg #	Page #	Page Offset

Answer the following questions:

- a) What is the number of segments?
- b) What is the maximum size of the segment?
- c) What is the size of a page?
- d) What is the maximum number of pages per segment?

Q6)

Assume that the main memory has the following 5 fixed partitions with the following sizes: 100KB, 500KB, 200KB, 300KB and 600KB (in order)

- How would each of the First-fit, Best-fit and Worst-fit algorithms place processes of 212KB, 417KB, 112KB and 426KB (in order)?
- Compute the total memory size that is not used for each algorithm.
- Which algorithm makes the efficient use of the memory?

Q7)

Assume that the MMU is using the multiple variable partition schemes. The memory size is 2560KB, from which the OS Kernel uses 400KB. You have the following processes:

<i>Process</i>	<i>Size</i>	<i>CPU burst</i>
P1	600K	10
P2	1000K	15
P3	300K	20
P4	700K	8
P5	500K	15

Using the First-fit allocation technique and the multiprogramming system (FCFS), do the following (without using compaction)

- Load and execute the processes until P5 is loaded (do not terminate P5).
- Every time a program is loaded, compute the **external fragmentation**.