CSCI 3753 Spring 2014 **Vince Coghlan**

Homework #4

Problem 1. [40 Points] You are asked to allocate a file according to either a File Allocation Table (FAT) or multi-level indexed allocation (UNIX inode - triply indirect). Assume that the file is 132 MB long, there are 2 KB per disk block, each pointer in the FAT occupies 4 bytes, and each index block entry takes 4 bytes.

(a) How many bytes are used to lay out the file when using a FAT file system?

132MB fits into 66000 Blocks. We need a pointer for every block as well as a EOF pointer and directory table entry, This means $66002 \cdot 4 = 264008$ bytes.

(b) How many bytes are used to lay out the file when using a UNIX-style file system?

We still need 66000 pointers, excluding the first 12 pointers, that is 129 blocks. That is easily covered by one singly indirect pointer. The total memory used is thus $15 + 2K + 129 \cdot 2K =$ 266255Bytes.

Now suppose that you wish to read the 66,000'th block of the file. Assume that each of the following counts as one search operation: moving from one element to the next in a linked list; indexing into an index block; moving from index block to the next.

(c) How many searches are needed to read block 66,000 when using the FAT file system?

66000

(d). How many searches are needed to read block 66,000 when using the UNIX-style file system?

$13 + 1 + 1 = 15$

Problem 2. [30 Points] Suppose you are given a flash memory consisting of 4 KB pages, and there are 1024 pages. Assume that 300 pages are currently allocated.

(a) How many bytes of memory would the OS need to keep track of free space if a bitmap is used?

$1024/8 = 128$ bytes

(b) How many bytes of memory would the OS need to keep track of free space if a linked list is used? Assume 2 bytes/pointer.

$(1024 - 300) * 2 = 1448$ bytes

(c) Under what conditions would a linked list be more memory-efficient than a bitmap?

if there are less than 64 free blocks then it will take less than 128 bytes. The bitmap will always take 128 bytes.

Problem 3. [30 Points] A disk has 1000 cylinders, 0 – 999. Disk access requests arrive in the following order of cylinder numbers: 30 80 200 3 702 80 96 89 100. Assume that the R/W head is on cylinder number 103 initially moving towards larger numbered cylinders. Calculate the total distance (Number of cylinders) travelled by the R/W head to service these requests if

(a) FCFS algorithm is used

$$|103 - 30| + |30 - 80| + |80 - 200| + |200 - 3| + |3 - 702| +$$

$$|80 - 702| + |96 - 80| + |89 - 96| + |89 - 100| = 1795$$

(b) SSTF algorithm is used

$$|103 - 100| + |100 - 96| + |96 - 89| + |89 - 80| + |80 - 80| +$$

$$|80 - 30| + |30 - 3| + |3 - 200| + |200 - 702| = 799$$

(c) SCAN algorithm is used

$$|103 - 702| + |702 - 3| = 1298$$