Alan Achtenberg

CSCE 410

Homework 3

12/2/14

1)

a. An absolute path is the location of the file/folder from the root node of the directory, it contains all parent directories of the file/folder. A relative path is the location of the file/folder from the current working directory and it does not contain all parent directories of the file/folder.
b. The current directory is the directory that all commands will be executed from. For example, creating a new file(without specifying a file path) will be done in the current directory.

2)

a. Can keep backpointers to all links to a file in the file. That way when we delete the file we can delete all links to the file.
b. We can leave the links to a deleted file and only delete the link when it is accessed
c. We can keep a reference count of each file, meaning when we delete a link we only delete the file if the reference count is 0.

3)

a. Contiguous
    i. P (physical address)= Z+L
    The physical address consists of the block number and the offset which are X and Y added to the starting address
    Pblock= Z+X Poffset=Y

    ii. Just one block must be read, because we know it is contiguous we simply move the head back 6 Blocks to get to 4 from 10.
b. Linked
    i. Because the list is linked the physical address has little to do with the position in the list. To get the Pblock we must start with the first block and find the next block from the pointer at the last index a total of X times
    The offset will be from the last block which Y+1 because we want to account for the pointer

    ii. Four blocks must be read following this scheme, because we always start from the start of the list. Ie block 4 requires 4 reads and 10 requires 10 reads.
    iii. head back 6 Blocks to get to 4 from 10.
c. Indexed
    i. Read the index block into memory and then use the index X to find the physical block address. The offset into the physical block is simply Y

    ii. Two reads must occur, the first is the reading of the index block and the second is the actual desired data.

4)

The sum is simply calculated by the distance from each position to the next.

a.

FCFS	1	2	3	4	5	6	7	8	9	10
address	143	86	1470	913	1774	948	1509	1022	1750	130

The total seek distance is 7081

b.

SSTF	1	2	3	4	5	6	7	8	9	10
address	143	130	86	913	948	1022	1470	1509	1750	1774

The total seek distance is 1745

c.

SCAN	1	2	3	4	5	6	7	8	9	10	11
address	143	913	948	1022	1470	1509	1750	1774	4999	130	86

The total seek distance is 9769, the extra index is for the algorithm reading the last position in the disk.

d.

C-LOOK	1	2	3	4	5	6	7	8	9	10
address	143	913	948	1022	1470	1509	1750	1774	86	130

The total seek distance is 3363

5) A block can store 512/64=8 inodes meaning a inode takes up 1/8 of a block. Every file needs a inode. Since we know how much space an inode takes we can simply treat the inodes as small files, making the new percentages be as such, 50% inodes that take up 1/8, 40% files that take 1 block, 5% files that take 2 blocks, 2.5% files that take 3 blocks and 2.5% files that takes 10 blocks. Using these percentages to maximize.

The largest number of files is 5620. (assuming the percent of files is held to be exact)

The number of inodes required is the same 5620, the amount of space in blocks is 5620/8= 702.5 blocks

Number	%	size(Blocks)	Space
4496	0.4	1	4496
562	0.05	2	1124
281	0.025	3	843
281	0.025	10	2810
5620	0.5	0.1250	702.5
total space			9975.5

6)

Number of I/O operations			
	Continuous	Linked	Indexed
A	201	1	1
B	101	52	1
C	1	3	1
D	198	1	0
E	98	52	0
F	0	100	0