CIS 415 – Operating Systems

Homework Assignment 3
Fall 2016 – Prof. Sventek

Due at 5:00pm on Tuesday, 22 November 2016

All questions must be answered by you without outside assistance. **Submission is via Canvas.** You may submit either a plain text (.txt) or a PDF (.pdf) file. Succinct, concise answers to the questions are preferable to long, rambling ones.

Textbook Questions (50 points)

1. OSC 10.11: [18 points]
Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 2150, and the previous request was at cylinder 1805. The queue of pending requests, in FIFO order, is:

2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681

Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms?

a. FCFS

current	next	cylinders
2150	2069	81
2069	1212	857
1212	2296	1084
2296	2800	504
2800	544	2256
544	1618	1074
1618	356	1262
356	1523	1167
1523	4965	3442
4965	3681	1284
	TOTAL	13,011

b. SSTF

current	next	cylinders
2150	2069	81
2069	2296	227
2296	2800	504

2800	3681	881
3681	4965	1284
4965	1618	3347
1618	1523	95
1523	1212	311
1212	544	668
544	356	188
TOTAL		7,586

c. SCAN

current	next	cylinders
2150	2296	146
2296	2800	504
2800	3681	881
3681	4965	1284
4965	4999	34
4999	2069	2930
2069	1618	451
1618	1523	95
1523	1212	311
1212	544	668
544	356	188
TOTAL		7,492

shorthand is (4999-2150) + (4999-356) = 2849 + 4643 = 7,492

d. LOOK

current	next	cylinders
2150	2296	146
2296	2800	504
2800	3681	881
3681	4965	1284
4965	2069	2896
2069	1618	451
1618	1523	95
1523	1212	311
1212	544	668
544	356	188
TOTAL		7,424

shorthand is (4965-2150) + (4965-356) = 2815 + 4609 = 7,424

e. C-SCAN

current	next	cylinders
2150	2296	146
2296	2800	504
2800	3681	881
3681	4965	1284
4965	4999	34
4999	0	5000
0	356	356
356	544	188
544	1212	668
1212	1523	311
1523	1618	95
1618	2069	451
	TOTAL	9,918

shorthand is (4999-2150) + 5000 + (2069-0) = 2849 + 5000 + 2069 = 9,918

f. C-LOOK

current	next	cylinders
2150	2296	146
2296	2800	504
2800	3681	881
3681	4965	1284
4965	356	4609
356	544	188
544	1212	668
1212	1523	311
1523	1618	95
1618	2069	451
	TOTAL	9,137

shorthand is (4965-2150) + (4965-356) + (2069 - 356) = 2815 + 4609 + 1713 = 9,137

2. Consider a file system that uses inodes to represent files. Disk blocks are 1 kB in size, and a pointer to a disk block requires 4 bytes. Each inode has 8 direct block pointers, as well as one each of single, double, and triple indirect block pointers.

n	2	3	5	10	20	30	40
2^n	4	8	32	1,024	1,048,576	1,073,741,824	1,099,511,627,776

label				kB	MB	GB	TB

a. What is the maximum size of a disk (in bytes) for which one can use this file system? [3 points]
Disk block pointer is 4 bytes, means we can address 2^{32} blocks on a disk using this file system. 1 kB is 2^{10} bytes. Thus, the maximum size of a disk for which one can use this file system is

$$2^{10} \times 2^{32} = 2^{42} \text{ bytes} = 2^{32} \text{ kB} = 2^{22} \text{ MB} = 2^{12} \text{ GB} = 2^{2} \text{ TB} = 4 \text{ TB}$$

b. What is the maximum size of a file (in bytes) that can be stored in this file system? [9 points]
- Since each disk block pointer is 4 bytes, this means that each indirect block can point to $2^{10} / 2^2 = 2^8$ blocks.
- The direct block pointers address 8 blocks;
- the single indirect block pointer addresses 2^8 blocks;
- the double indirect block pointer addresses 2^8 indirect blocks, each of which addresses 2^8 blocks, thus a total of $2^8 \times 2^8 = 2^{16}$ blocks;
- the triple indirect block pointer addresses 2^8 double indirect blocks, each of which addresses 2^{16} blocks, thus a total of $2^8 \times 2^{16} = 2^{24}$ blocks;

The total is thus $2^3 + 2^8 + 2^{16} + 2^{24}$ blocks $= 2^3 \times (1 + 2^5 + 2^{13} + 2^{21})$ blocks
= 8 x (1 + 32 + 8192 + 2,097,152) blocks = 8 x 2,105,377 blocks
= 16,843,016 blocks

size in bytes = 1 kB x 16,843,016 blocks = 16,843,016 kB = 16.063 GB

All other contributions are dwarfed by triple indirect, so the size is $\approx 2^{24}$ kB = 16 GB

3. The processor for which you are designing your application as L1i and L1d virtual caches.

a. What type of data does each cache hold? [2 points]
L1i holds recently accessed instructions, L1d holds recently accessed data

b. Describe in detail the activities of the cache + memory system when executing the instruction [3 points]

LOAD virtual address, register

1. check L1i for the instruction; if not found, fetch from memory and insert into L1i cache
2. check L1d for the "virtual address"; if ! found, fetch from memory and insert into L1d cache
3. insertion into L1d cache may trigger writeback to RAM if line being evicted is dirty

c. Assume that the above instruction is executed many times in a loop, and that the instruction itself is in the cache. Also assume that memory access costs τ μs, and cache access costs $\tau/15$ μs. What cache hit rate ρ for "virtual address" is required for the memory system to run 5 times faster than with no caching at all? Show your work. [7 points]
1. without caches, each access is τ to fetch the instruction and τ to fetch the data, for a total of 2τ
2. with the caches, and assuming that the instruction is in the cache, then each access

is $\tau/15$ for the instruction, $\tau/15$ to check for "virtual address" in the data cache, and $(1-r)\tau$ to fetch the data from memory when not in the cache
3. thus, $2\tau/((2/15+1-r)\tau) = 5 \rightarrow 2/15 = 17/15 - r \rightarrow r = 17/15 - 6/15 = 11/15$

d. Suppose we have a memory system that has a main memory, a single-level cache, and demand paging virtual memory. The three levels of the memory system have the following access times:

Cache	2ns
Main memory	100ns
Paging disk	10ms

i. The cache has a 95% hit rate. What is the effective memory access time if we consider only the cache and main memory and ignore page faults and disk access times? [3 points]
$t_{eff} = t_{cache} + (1 - r) * t_{mem}$
$t_{eff} = 2ns + 0.05 * 100ns = 7ns$

ii. Now recalculate the effective memory access time assuming the same cache hit rate (95%) plus a page fault rate of 0.001% (i.e., 99.999% of the memory accesses succeed without producing a page fault). [5 points]
$t_{eff} = t_{cache} + (1 - r) * [(1 - pfrate) * t_{mem} + pfrate * t_{disk}]$
$teff = 2ns + 0.05 * [0.99999 * 100ns + 0.00001 * 10{,}000{,}000ns]$
$= 2ns + 0.05 * [99.999ns + 100ns]$
$= 2ns + 9.99995ns = 11.9995ns \approx 12ns$

Note: Like all assignments in this class **you are prohibited from copying any content from the Internet or sharing ideas, code, configuration, text or anything else or getting help from anyone in or outside of the class, except where noted**. Consulting online sources is acceptable, but under no circumstances should anything be copied. Failure to abide by this requirement will result in sanctions ranging from zero on the assignment to dismissal from the class.