Homework for Week 12
(Due Date: Check Canvas)

1. [1] Consider a RAID level 5 organization comprising of five disks, with the parity for sets of four blocks stored on the fifth disk. How many blocks are accessed in order to perform the following?

   (a) A WRITE of one block of data.

   (b) A WRITE of seven continuous blocks of data.

2. [1] Consider a file system that uses inodes to represent files. Disk blocks are 8KB in size, and a pointer to a disk block requires 4 bytes. This file system has 12 direct disk blocks, as well as single, double, and triple indirect disk blocks. What is the maximum size of a file that can be stored in this file system?

3. [4] Consider using a bitmap versus a linked list of free blocks. The disk contains a total of B blocks, F of which are free. A disk address requires d bits. The bitmap uses one bit for each block. The linked list is a data structure maintained in a dedicated portion of the disk. Each list element points to a single free block.

   (a) State the condition under which the two methods use the same amount of disk space, assuming that the linked-list method connects all blocks individually.

   (b) For $d = 16$ bits, determine the fraction of the disk that must be free for the above condition to hold.

   (c) Repeat the two problems above, assuming that the linked-list method connects groups of adjacent blocks, rather than individual blocks. That means, each list element points to the first of block of a group, and contains a two-byte number indicating how many blocks are in the group. The average size of a group is five blocks.

References

[1] A. Silberschatz, P. Galvin, and G. Gagne, *Applied Operating Systems Concepts*, John Wiley & Sons, Inc., New York, NY, 2000.

[2] Deitel, Deitel, and Choffnes, *Operating Systems*, Pearson / Prentice Hall, 2004.

[3] A. S. Tanenbaum, *Modern Operating Systems*, Pearson / Prentice Hall, 2008.

[4] L. F. Bic, A. C. Shaw, *Operating Systems Principles*, Prentice Hall 2003.

[5] C. Crowley, *Operating Systems, A Design-Oriented Approach*, Irwin 1997.

[6] M. Herlihy, N. Shavit, *The Art of Multiprocessor Programming*, Elsevier, 2008