ECE566 Enterprise Storage Architecture
Individual Homework #2:
NAS, SAN, and Filesystems

Directions:

- This assignment will be completed in ***INDIVIDUALLY***. While you can discuss concepts with others both within and outside your group, actual steps and answers should not be shared!
- The solutions should be collected in a PDF file submitted via GradeScope. *Word documents will not be accepted*.

1 SAN and NAS [30 pts]

1.1 Comparing and contrasting

Consider this post by NetApp founder Dave Hitz. For a workload where you treat a big file as if it's a block device, how are NAS and SAN protocols similar? In what ways do they differ? What is the motivation for considering NAS protocols for workloads more traditionally supported by a SAN? [10]

1.2 Critical thinking about SAN and NAS

One common thing in the storage industry is for vendors and consultants to write misleading or uninformed whitepapers, either to directly promote their business or to present the appearance of expertise to drive consulting business. Thinking critically about author motivations and carefully assessing the claims presented is an important skill.

This archived article by cloud storage start-up Zadara contains several strange implications (it was later taken down, but the link goes to an archive mirror). For example, it says a NAS server is "a lot like ordinary file servers" (implying that it somehow *isn't* just a file server). Other than this, identify two false statements or implications in this article and explain why they are wrong. [10]

In this archived article by IT consultant CCB Technology, every one of the "cons" listed for NAS is wrong. Explain why. [10]

2 Filesystems [55]

2.1 Block allocation

Assume a filesystem uses simple sequential allocation of blocks as you answer the following questions. If you aren't sure, talk through it with the instructor.

a) Describe a sequence of file creation and deletion operations that would result in high *internal* fragmentation. [5]

b) Describe a sequence of file creation and deletion operations that would result in high *external* fragmentation. [5]

2.2 Indexed filesystems

Assume a filesystem that uses inodes similar to ext2, except with the following changes:

- Block size and inode size is 32 bytes (yes, this is absurd, but it makes the question work).
- Offsets are 64-bit (8 byte).
- Inodes have one direct, one single indirect, and one double indirect block index.

Assume that we want to store a 350-byte file in this filesystem.

a) For the file, how many bytes of internal fragmentation will there be for the data blocks alone? Show your math. [6]

b) For the file, draw a diagram of boxes and arrows that shows the inode, indirect blocks, and allocated data blocks. Label the inode "inode", label the indirect blocks "single indirect" or "double indirect", and label and number the data blocks "block 0", "block 1", etc. [6]

c) Based on your diagram, how many total disk blocks were allocated to store this file (including all metadata)? [6]

d) Based on the above, what is the metadata *overhead* for the file (metadata space divided by actual file content)? [6]

e) Name a change you could make to the filesystem design to lower this overhead. [6]

2.3 Raw flash file systems

Research the JFFS2 filesystem. What design features allow this filesystem to function directly on raw flash storage without a flash translation layer? [15]

3 File sizing and sparse files [15]

Research the concept of sparse files. This will be especially relevant in Lab 2.

a) What is a sparse file? [5]

b) What is the difference between *allocated size* and *apparent size*? [5]

c) How do you create a sparse file in Linux? How do you find allocated vs. apparent size for a file in Linux? [5]