

CSE 303: Quiz #4

Due October 23, 2019 at 1:35 PM

Please use the remainder of this page to provide your answer. To submit your answer, create a pdf whose name is exactly your user Id and the “pdf” extension (e.g., abc123.pdf) and email it to spear@lehigh.edu before the deadline.

Imagine that in some i-node filesystem, only 128 bytes were needed for metadata. It would then be possible to use some space *in the inode* to store file contents. Let the i-node have 12 direct pointers, one indirect pointer, one double indirect pointer, and one triple indirect pointer. Be sure to show your work as you answer the following questions:

What is the biggest file that can be stored with just one i-node?

Since every block (including i-nodes) in the i-node structure for a filesystem is 4KB, we can fit a sufficiently small file within the i-node itself given that the metadata is small enough to allow it. Pointers are 4 bytes each, so $15 * 4 = 60$ bytes total. The metadata of the i-node was given to only contain 128 bytes. Therefore, given a max size of 4096 bytes, you can fit $4096 - 188$ bytes or 3908 bytes for a file to be stored into one i-node.

What is the biggest file that can be stored by the file system?

The biggest file that can be stored in the file system would be equal to all of the data blocks that a single i-node can provide for. This would mean $4KB * (12[\text{direct pointers}] + 1024[\text{indirect pointers}] + 1024^2[\text{double indirect}] + 1024^3[\text{triple indirect}])$ is the biggest file that can be stored into the file system.

Consider the file `/folder1/folder2/file`. Suppose the root folder has 64 files in it, with each filename being no more than 256 characters. Also suppose that `folder1` and `folder2` each only have one entry (`folder2` and `file`, respectively). Starting at the root, how many i-nodes will be accessed in order to read the first 1KB of data from the file? Justify your answer.

Since each i-node is meant to represent a single file or directory, I think that you would need one i-node per directory in the path plus the one for the file itself. This means that there would be 1 i-node for root + 1 i-node for `folder1` + 1 i-node for `folder2` and then + 1 i-node for storing the file itself. This totals to 4 i-nodes for accessing the file.