National University of Singapore
School of Computing

CS2106 **Tutorial 10** Semester 1 11/12

1. A file system uses 4KB disk blocks and 32 bits block addresses. i-nodes are used to keep track of blocks allocated to a file. Each i-node has 12 direct block addresses, one single indirect block address, one double indirect block address, and one triple indirect block address. What is the large possible file size on such file system?

   **Solution:** Each block can store $2^{12}/2^2 = 2^{10}$ addresses. Each i-node can refer to (directly and indirectly) a total of $12 + 2^{10} + 2^{20} + 2^{30} = 1{,}074{,}791{,}436$ blocks. Since each block is 4KB, the maximum file size is 4,299,165,744 KB (about 4 TB).

2. Would contiguous allocation be a good file allocation scheme for (i) flash storage in a digital camera? (ii) harddisk storage on a digital video recorder?

   **Solution:** (i) yes, files are typically created in order and copied to computer. File size is fixed (does not grow). (ii) no, shows are deleted after they are watched, creating holes.

3. In the design we discussed in class, the i-nodes are kept at the start of the disk. An alternative design is to allocate an i-node only when a file is created and place the i-node at the start of the first block of the file.

   Discuss the pros and cons of this alternative.

   **Solution:** Pros:

   - There is no limit on the number of i-nodes. We can have as many i-nodes as as many files.
   - i-nodes only occupy space if a file is used.
   - Reading is faster since an i-node is co-located with the first block of the file.

   Cons:

   - Looking up i-nodes is difficult. We can no longer derive where the i-node is stored based on the i-node number.
   - The whole block is used even when file is very small (e.g., empty file, or a device file that does not occupy disk space).

4. Consider a file system that uses i-nodes. Suppose that you are uploading a huge file to your friend.
   (a) While the file is still being uploaded, you decided to move the file into a different directory (on the same file system). Should this be allowed? Justify your answer.
   (b) What if you want to move the file to a different disk partition. Should this be allowed? Justify your answer.

**Solution:**

(a) Yes, we only need to move the directory entry. The file descriptor is pointing to the i-node, and the i-node does not change. The uploader can still keep tracks of the locations of next blocks to read from the same entry in i-node table.

(b) No, the i-node would no longer be valid after moving and the disk blocks on the original disk would be freed.