CSC 262
Homework #4
Due at the start of class on Wednesday, February 25.

1. **Block Allocation**. This question focuses in detail on the specifics of file block allocation on a Unix system.

Recall that block allocation in Unix is via *inodes*, which point to the sequence of disk sectors holding the disk's data. The root inode of a file includes all of the file attributes maintained by the system, leaving only a little space for block pointers. The capacities of the various inode blocks on a particular system are as follows:

Level	**Number of Blocks**	**Number of Bytes**
Direct	12	12K
Single Indirect	256	256K
Double Indirect	256^2	65M
Triple Indirect	256^3	16G

For the three parts below, calculate how many 1K blocks would be occupied by the file described. Then draw the inode structure that would be required to store a file of that size. Your drawing should show the inode structure clearly, but need not include exacting detail. For example, you shouldn't draw all 256 lines in the single-indirect inode, but should use an ellipsis (…) instead. If you like, when portions of the drawing are repeated, you may draw them once with a label such as (A) and refer to them symbolically thereafter. You don't need to draw the data blocks, but should indicate where the last data block of the file is indexed.

a. A file of 10 bytes.

b. A file of 260 KB.

c. A file of 320 MB. (Make sure you plan your drawing to leave yourself enough room.)

2. **Protection and Security**. The designers of a new Unix-like operating system are contemplating a change to the standard protection mechanism. Instead of an access list (rwxrwxrwx) associated with each file, there will be a list of capabilities (rwxrwxrwx) associated with each user. The first three letters indicate the capabilities of the user with respect to files owned by that user, the middle three letters indicate the capabilities of the user with respect to files in a group to which the user belongs, and the last three letters indicate the capabilities of the user with respect to all other files.

a. Assuming there are more files than users, will this new mechanism require more storage or less than the standard Unix mechanism would require?

b. Characterize the flexibility that the new mechanism allows when selecting protection policies, as compared to the standard Unix mechanism. Does it offer the same flexibility, strictly more flexibility, strictly less, or more in some ways but less in others? Explain, using specific examples.

c. Now, forgetting the implementation specifics in this problem, characterize the flexibility allowed by capability lists in general when selecting protection policies, as compared to access control lists. Does they offer the same flexibility, strictly more flexibility, strictly less, or more in some ways but less in others? Explain, using specific examples if necessary.