**Part 4 - Word Problems**

**1. Suppose that in the DAC implementation in section 2, we introduced ‘groups’ of users. How would you modify your implementation to include access rights? Are there any similarities or differences between ‘groups’ and ‘roles’ in the two implementation of the authorization module?**

We would modify our implementation to include access rights by only permitting groups as entries.

DAC systems assigns access rights based on rules specified by users. Subjects can determine who has access to their objects and change the permissions of files they own, making it discretionary.

RBAC system Users are assigned to Roles based on their responsibilities. Roles are similar to groups in Unix file system DAC, with two distinctions. Firstly, a group is a set of users, whereas a role is a set of rights. Secondly, in some implementations, a user is always a member of a group, whereas a subject may activate or deactivate rights associated with any subject’s roles. Subjects may login with most of their roles deactivated, and activate a role only when the rights associated with the role are necessary.

**2. What other ‘constraints’ can be imposed on the RBAC implementation in section 3. Give examples and discuss briefly how you would implement them in your code.**

A constraint is a defined relationship among roles or a condition related to roles. The following constraints can be imposed on the RBAC implementation in section 3:

Mutually exclusive roles are used when we want a user to be assigned to only one role in the set either during a session or statically, and that any permission can be granted to only one role in the set. Therefore, a set of mutually exclusive roles have non-overlapping permissions. For example, if two users are assigned different roles in the set, then the users have non-overlapping permissions while assuming those roles. The purpose of this is to increase the difficulty of collusion among individuals of different skills or divergent job functions.

Cardinality is another constraint that is used to set a maximum number of users that can be assigned to a given role. For example, in section 3 only one user can be assigned the role ‘Manager’. System can also impose a constraint on the number of roles a user can activate for a single session or set a maximum number of roles that can be granted a particular permission.

Prerequisite constraint can be used to dictate that a user can only be assigned to a particular role if it is already assigned to another specified role. In this case, the manager automatically inherits all the rights of other users to the files.

**3. Another ability of RBAC-based system is to allow for ‘sessions’. Give an example of how a session can be useful for the set up in the lab, and how your code would represent the relationship between users, roles, and the given session.**

A session is a mapping between a user and an activated subset of the set of roles to which the user is assigned. It is used to define a temporary one-to-many relationship between a user and one or more of the roles to which the user has been assigned. The user establishes a session with only the roles needed for a particular task which establishes which permissions are available to the user during the time of the session.

For example, this can be useful for dynamic separation of duty constraints, where two roles can be assigned to the same user, but they can’t be used together. Therefore if the user wants to use one of the roles, he or she must deactivate one before activating the other.