**4 Word Problems**

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

Adding group access rights would give the user 2 check phases. The first would be the group they are a part of and the second the user themselves. If the user is part of a group the system checks if that group they are a part of has permission to act on the file the way the user is trying too. If not the system goes onto the user themselves to see if their default rights give them the required access.

Implementing groups this way gives the user group rights but does not change the default permissions the user had before. If you were to change the user rights to strictly that of the group, you would skip the second check mentioned above.

Groups and roles share the responsibility of passing its permissions to the user(s) who are in the group or are assigned to a role. However, there are some differences as to how the two operate. A group will have a set of permissions that can be changed and pass those permissions onto a collection of users who inhabit the group. A role on the other hand will have permissions set to that role. Those permissions get passed to the individual who inhabits that role at a given time. Typically group permissions can change where a role is set permissions.

**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.**

One constraint could be that users can only be assigned one role. Many can occupy sales (or other roles) but no one user could occupy multiple roles. In our implementation there would be a check function that would check to see if the user already has a role, if so they cannot be assigned another role. Another constraint is that the user must check with the manager before switching their role if desired. The manager would have to give them permission to drop current role and then they can choose new role.

**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 ‘sessions’ is whenever the user logs into the system and the session can have many roles for the user but only 1 role can be activated at a time for each session. For this lab a session can be useful for when a user has more than one role, for example a user has Sales and Technical staff. They can only have the access rights of one role in a session so if a user is a using their Sales role in the session they’d have to deactivate their Sales role and then activate their Technical staff role if they want to use their access rights for technical staff.

A session is useful for limiting the number of users in a system at a time for a given group/role. It can also allow someone of an admin role to view who currently has a live session to see who is currently accessing the system. You can also implement a system to record sessions if you ever need to go back and view what someone did for security reasons. As a session is the mapping of a user to a role the relationship between the user, a role, and the given sessions would be deciding what role the user is taking on during that session. When a user joins the system and starts a new session, that session needs to be mapped to a specific role, and no other, for the duration of that session.