System Programming – Project 2

Develop a device driver that will act as a simple message box between the users on a system. Placing a message into the message box will be achieved through a write operation on the device (named "/dev/messagebox"), starting with a constant prefix such as "@USERNAME" that will specify the recipient. When a user reads from the device, he/she will only see the messages addressed to him/her.

For example, if the user "alice" wants to send a message to the user "bob" to say "Hello", she will write to the device the text "@bob Hello". Later, when user "bob" reads from the device he will see the message as "alice: Hello". Although the message box is global for all users, user "bob" will not see any messages sent to other users.

When a user reads a message, the message should be marked as "read". In the default mode (EXCLUDE_READ), a read operation on the device will only return unread messages. If the message box is in "read all" mode (INCLUDE_READ), both unread and read messages should be returned. The mode should be changeable using an ioctl command. Only the superuser should be allowed to change the mode.

There is a limit on the number of unread messages that a user can receive. This is a global setting that applies to all users and should be a module parameter. If a user's message box is full, the system should indicate a proper error. Note that read messages are not counted toward this limit. Also implement an ioctl command for modifying this limit value. Only the superuser should be allowed to set this limit.

Finally, implement an ioctl command that will take a user name as parameter and delete all messages where this user is the recipient. Only the superuser should be allowed to do this operation.

What you need to implement:

- Write the device driver as described. The device should be usable through standard utilities like "cat" and "echo" and using redirection as in:

  ```
  echo "@joe hello" > /dev/messagebox
  cat /dev/messagebox
  ```
- Write test programs for demonstrating the ioctl commands.

Important notes!

- Pay attention to the general guidelines for projects.
- You are required to submit the source codes of all your files (including your test programs) through the Ninova system as a zip file.
- Make sure to return a meaningful error code when an error occurs.
- Don't blindly copy any code from other sources. If you base your code on the scull example, make sure that you use proper names in your code as opposed to leaving names from the scull example around.
- Each member of the team must make an individual submission, even though the submitted files are the same for all members.
- Team members will be graded individually based on their performance in the lab session and the submitted team project. Students who are not present during the lab session will not receive a grade for the project, even though they may have made a submission through the Ninova system.

Any form of cheating or plagiarism will not be tolerated. The submitted work should be the work of the team; collaboration or code sharing between different teams will be regarded as cheating. Cheating also includes actions such as, but not limited to, submitting the work of others as one's own (even if in part and even with modifications) and copy/pasting from other resources, including Internet resources, (even when attributed). Serious offenses will be reported to the administration for disciplinary measures.