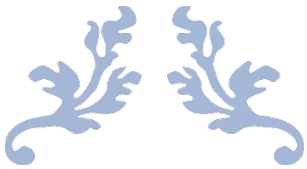

BLG 413E
Project 2 Report

Group Members

ID	Name	Surname
150120138	Emre	Özdil
150140115	Arzugül Merve	Ecevit

1. INTRODUCTION

For this project, we implemented a device driver which acts as a simple message box between the users on a system.

In this project, main aim is to modify read and write function in order to run “echo” and “cat” commands properly. We wrote “get_username” function for getting the current username in this terminal section.

We created 3 structs for this project and used linked list as data structures. Struct definitions are shown in below:

```
struct message {
    char *text;
    int isRead;
    char *fromUsername;
    int messageSize;
    struct message* next;
};

struct account {
    char *username;
    struct message* receivedMessages;
    int unreadCount;
    int totalMessage;
    struct account* next;
};

struct messagebox_dev {
    struct account* users;
    int isInclude;
    struct semaphore sem;
    struct cdev cdev;
};
```

- Message struct

Message struct consists of 5 information which are the text of message, a boolean for is the message is read, to whom the message should be sent, size of the message and next pointer for the message linked list.

- Account struct

Account struct consists of 5 information which are the username of account who received the message, list for received messages, user’s unread message count, user’s total message and next pointer for the account linked list.

- Messagebox device struct

Messagebox device struct consists of 4 information which are the list of users, a boolean for is the read mode is include or exclude, a semaphore for operations and a device.

2. FUNCTIONS FOR DEVICE DRIVER

a. Messagebox_init_module

In init function, the initialization of device driver is done. The major number of device driver is given. The parameters are set to 0.

b. Messagebox_open

In open function, the device will be opened and the data will be moved to device driver.

c. Messagebox_read

In read function, if user enters cat command, this function will start. Message and user will be taken from linked lists which are messagebox_device->users and messagebox_device->users->receivedmessages. In order to find the username of user, getUsername function will be called.

d. Messagebox_write

In write function, if user enters echo command, this function will start. Message and user will be added to linked lists which are messagebox_device->users and messagebox_device->users->receivedmessages. In order to get the username of user, getUsername function will be called.

e. getUsername

In getUsername function, from /etc/passwd file, the username of user will be found using the id of user,

f. Messagebox_ioctl

In ioctl function, using switch case, ioctl commands will be implemented. The commands are explained in below.

g. Messagebox_trim

In trim function, from messagebox_device->users, first user's received messages will be deleted in order. Then, user will be deleted. This will got to until the end of messagebox_device->users.

h. Messagebox_cleanup_module

In clean up function, the memory which are taken in messagebox_init function will be given back and messagebox_trim function will be called to free users and messages linked list. Module will be closed.

3. IOCTL COMMANDS

a. Message Box Exclude Mode

In exclude mode, we changed the boolean of program "isInclude" variable to 0, By this way, user will see just the unread messages. This is the default mode of program. To change the mode, user should have the permission.

b. Message Box Include Mode

In include mode, we changed the boolean of program "isInclude" variable to 1, By this way, user will see the all messages. To change the mode, user should have the permission.

c. Delete All Messages

In this mode, we took the username of user via terminal. Then in users linked list, the username will be found and received messages will be deleted. However, we have a crash in this part, after deleting if we want to add a message it is crashing. The problem can be about the pointers which we should not totally delete. To delete all messages, user should have the permission.

d. Change Maximum Unread Message Count

In this mode, we took count of maximum unread message for user via terminal, then, the driver changed the number of unread messages allowed. To change maximum unread message count, user should have the permission.