CSS 551: Operating System Design and Implementation

Project 3

Protection for the Mailbox IPC

Sunny Changediya (A20353568)

Parth Desai (A20351426)

Chang Liu (A20365835)

Chengxi Sun (A20366976)

Contents

2.1 Secure System Call Manual:

System Call	Call_nr	Function	Argument
MINIT	79	Initialize mailbox & owner List	NA
MSENDER	108	Register User to mailbox	(UserID, mailbox_id)
USRSEND	69	User process send system call	(sendID,msg)
USRRECIVE	70	User process receive system call	(recvID,msg)
MSEARCH	103	Search ownerID, usedID, mailbox_id, mailbox_name, and map to specific ID	(userID,regnum,mailbox_id) (name,regnum,mailbox_id)
REMUSER	105	To remove sender/recv from list	(userID, mailbox_id)
MGARBAGE	56	To garbage mailbox messages	NA
SNAP	44	To display messages from mailbox	NA
MCREATE	58	Create a new Mailbox	(ownerID,name,type)
CHANGEP	97	Update permission of user	(userID,perm)
DENYACCESS	110	To block access to user	(userID,new perm)

2.2 Design of System Calls & manual Page:

MINIT	int do_minit()
Library wrapper	int mailbox_init(void);
Description	This call initializes the owner list and mailbox list
Arguments	NA
Return value	0

MSENDER	int do_msender()
Library wrapper	int add_user_list(int sid, int regnum, int mailbox_id)
Description	Register new user based on mailbox_id
Argument	Userid, regnum=search/register, mailbox_id is automatically retrieved from back of system
Return value	1 on user registered

USRSEND	int do_usrsend()
Library wrapper	int mail_insert(char data[], int no_of_recv, int destrec[], int mailbox_id)
Description	It sends message to designated mailbox_id with receiver list
Argument	message, receiver/msg count, receiver-list, mailbox_id
Return value	1 on message deposite

USRRECIVE	int do_usrrecive()
Library wrapper	int mail_receive(char *data[], int recid, int mailbox_id)
Description	It retrieves message from designated mailbox_id by checking receiver id in receiver_list with each message receiver list
Argument	message, receiver_id, mailbox_id
Return value	Msg display & 1 on success

MSEARCH	int do_msearch()
Library wrapper	int search_list(char name[10], int id, int regnum, int mailbox_id, int flag)
Description	Search call to search userid, mailbox_name, and unique mailbox_id. Flag is used for return value purpose. Based on flag function returns user-id,mailbox-id or type of mailbox.
Argument	Name, userid, mailbox_id, flag value to return
Return value	Flag is returned based on value set in function

REMUSER	int do_remuser()
Library wrapper	int remove_user(int user_id, int mailbox_id, int user_flag)
Description	Remove user of that mailbox based on mailbox-id. Flag is used to identify the correct user list like owner-list,user-list or public-users list
Argument	userid, mailbox_id, flag
Return value	0 on user removed

MGARBAGE	int do_mgarbage()
Library wrapper	NA
Description	Deadly system call which removes mailbox, owners, users and everything. Should be used only by superuser.
Argument	Mailbox=-1, owner_list=1, user_list=2
Return value	0 (always success)

SNAP	int do_snap()
Library wrapper	NA
Description	Display mailbox status, user/owner available
Argument	Owner_list=1, user_list=0
Return value	0 (always success)

MCREATE	int do_mcreate()
Library wrapper	int mail_init(int ownID, char name[], int type)
Description	Creates a new mailbox and rights are given only to the owner
Argument	Ownerid, mailbox-name, type=public/secure
Return value	0 (always success)

CHANGEP	int do_changep()
Library wrapper	int change_perm(int ID, int mailbox_id, char *perm)
Description	Update the permissions of user. Only owner & superuser have this rights
Argument	Userid, mailbox-id, permission
Return value	1 on permissions updated

DENYACCESS	int do_denyaccess()
Library wrapper	int deny_access(int userid, int mailbox_id, int bit, int flag)
Description	It is used to deny send/receive access of any public user. Used with authentication of owner as only owner and superuser have rights. Flag is used to identify whether to make update privilege or just search for that privileges
Argument	Userid, mailbox-id, SEND/RECV bit, flag
Return value	1 on permissions updated or found

2.3 Exception Handling:

1. Exception:

- To make PUBLIC mailbox accessible, we are depositing message from all users even if that user is not registered to system. It will be automatically registered and message is deposited. Malicious message may hamper mailbox and halt the system.
- Superuser is given ultimate power to do everything and can specify as many owners as he wants. Maximum number of owners into system may lead to numerous exceptions and mailbox handling may be hard.
- Due to lack of message format structure, critical part of authentication and protection is based on mailbox name and garbage or occurrence of null character may induce errors and make unable to access mailbox access. This can be handled by introducing new message structures into MINIX.

2.4 Test Cases:

NO.	Test case description	Expected results	Test results	status
1	Register owner1	owner registered	Success	PASS
2	Register owner1	owner register fail	Fail	PASS
3	Register sender/receiver to mailbox M1	User registered	Success	PASS
4	Register sender/receiver to unauthorized mailbox M2	Not registered	Fail	PASS
5	Receiver retrieve message from authenticated mailbox	Message received	SUCCESS	PASS
6	Register Same users to multiple mailboxes	Users registered	Success	PASS
7	user deposit/retrieve message from unauthorized mailbox	Not allowed	fail	PASS
8	Change privileges of user by owner	Privileges changed	Success	PASS
9	Change privileges of owner by superuser	Owner privileges changed	success	PASS
10	PUBLIC mailbox creation by registered owner	Public mailbox registered	Success	PASS
11	Allow all users to send and receive from public mailbox	Access allowed and user registered to public user list if not avail already	success	PASS
12	Deny send/receive for users in public mailbox	Send/recv access removed	success	PASS
13	Remove public user from list	Public user removed by authorized owner	success	PASS

14	Public user send/recv from mailbox except public	Access denied	Fail	PASS
15	User access registered mailbox	Access allowed	Success	PASS
16	Create new users for specific mailbox	Only authorized owner allowed to create	Success	PASS
17	Empty the mailbox by owner	Allowed to that owner	success	PASS

2.5 Criticism:

1. The propped security constraint is vague in some cases. Having owner privileges and then again removing those privileges doesn't really make sense. Because once removed, owner is no longer perceived as owner but eventually becomes a normal user.
2. Giving sender/receiver permission to send/receive does not make sense. Sender and receiver have existence into system only for that purpose and to remove their any of the one privilege makes them inactive and may lead to system halt. Eg. Sender1 does not have read access. And mailbox may have old messages to be read by sender1. Since sender1 is already into system we cannot garbage the message for sender1 and also message can't be read by sender1. So making system contained with obsolete messages and may halt after some time.
3. Mailboxes are created by only owner and having many public mailboxes is invalid to system. It will confuse users to which public mailbox they send/receive message. If specific public mailbox is specified then it is as good as having secure mailbox. Systems generally have only single public entity.