

Network protocol – Mail Application

By: Noam Mansur 305596785, Alon Kollman 203565650

General Comments		
<ul style="list-style-type: none">- The server can address only one client at a time.- The connection between the server and the client is created at the start of the client program run, and disconnects only after the client did a proper sign out or if the client process was killed. Only after that happens, the server can address a different client.		
General Message Structure		
<p>Each Message consists of two 'size' bytes followed by the content of the message.</p> <p>The size in the 'size' bytes is the size in bytes of the content (meaning we don't take in account the two 'size' bytes in that number).</p> <p>Maximum length of message is SMALL_BUFFER_SIZE (see under constants values) unless stated otherwise.</p>		
Constant Values		
DEFAULT_PORT	6423	The default port.
SUCCESS_MSG	"Success"	Message sent on success of certain operations.
FAIL_MSG	"Failure"	Message sent on failure of certain operations.
SMALL_BUFFER_SIZE	100	Maximum length of most messages.
BIG_BUFFER_SIZE	5000	Maximum length of some messages.
MAX_USERNAME	50	Maximum length of username.
MAX_PASSWORD	50	Maximum length of password.
MAX_SUBJECT	100	Maximum length of email subject.
MAX_CONTENT	2000	Maximum length of email content.
TO_TOTAL	20	Maximum users that one can send a message to.
Connection and authentication		
<p>After establishing connection between the two parties, the server should send a 'greeting' message.</p> <p>After receiving the greeting, The client should send a message of the form: 'USER;PASS' where USER is the username, and password is PASS , both limited to MAX_USERNAME and MAX_PASSWORD respectively.</p> <p>The server then returns a SUCCESS_MSG or FAIL_MSG according to whether the state of the authentication.</p>		
Client Message:	'USER;PASS'	Server Response: SUCCESS_MSG or FAIL_MSG

Mail Operations			
Composing a new email			
<p>To compose a new email, the client should send two messages:</p> <p>First one is 'COMPOSE' and after it a message of the form: 'RECEIPIENTS;SUBJECT;CONTENT' where:</p> <ul style="list-style-type: none"> - RECEIPIENTS is a string of usernames separated by commas ('NAME,NAME,NAME...') where each username is limited to MAX_USERNAME and the number of users to be on the recipients list is limited to TO_TOTAL. - SUBJECT is the subject of the email, limited to length MAX_SUBJECT. - CONTENT is the content of the email, limited to MAX_CONTENT. <p>RECEIPIENTS, SUBJECT and CONTENT should not contain semicolons or new line characters. The sever responds with SUCCESS_MSG or FAIL_MSG.</p>			
Client Message:	'COMPOSE'	Server Response:	No Response
Client Message:	'RECEIPIENTS;SUBJECT;CONTENT'	Server Response:	SUCCESS_MSG or FAIL_MSG.
Checking inbox			
<p>To check one's inbox, the client sends the message: 'SHOW_INBOX'.</p> <p>The server then responds by sending for every email in the user's inbox a message of the form: 'MSG_ID. FROM "SUBJECT"' followed by a single message: 'END' to indicate there are no more messages at that user's inbox.</p> <p>MSG_ID is the unique message id in the user's inbox, FROM is the username that sent the message and SUBJECT is the message subject.</p>			
Client Message:	'SHOW_INBOX'	Server Response:	Multiple messages of the form: 'MSG_ID. FROM "SUBJECT"' followed by the message: 'END'.
Fetching an email			
<p>To get the content of a specific email, the client sends a message: 'GET_MAIL MSG_ID' where MSG_ID is the id of the message the user wants to read.</p> <ul style="list-style-type: none"> - If message is found, the server responds with a message of the form: 'FROM;TO;SUBJECT;CONTENT' where FROM is the username who sent the email, TO is the recipients of that email, SUBJECT is the email's subject and CONTENT is the email's content. Notice that this message is limited to BIG_BUFFER_SIZE. - If message is not found, the server responds with a FAIL_MSG. 			
Client Message:	'GET_MAIL MSG_ID'	Server Response:	'FROM;TO;SUBJECT;CONTENT' or FAIL_MSG
Deleting an email			
<p>To delete a specific email, the client sends a message: 'DELETE_MAIL MSG_ID' where MSG_ID is the id of the message the user wants to delete.</p> <ul style="list-style-type: none"> - If message is found, the server should delete and respond SUCCESS_MSG. - If message is not found, the server responds with a FAIL_MSG. 			
Client Message:	'DELETE_MAIL MSG_ID'	Server Response:	SUCCESS_MSG or FAIL_MSG
Signing out			
To sign out properly, the user should send the message: 'QUIT'.			
Client Message:	'QUIT'	Server Response:	No response