Project 4

For this project we are going to be working with arrays, pointers and structs. Additionally, we'll be working on the some basics of cybersecurity. We will be working with electronic communication and making sure the messages we receive are authentic and haven't been tampered with.

Details

For this project we will once again have a command file. The command file will have a series of commands that instruct our program what to do. The commands will be described below. The commands will deal with messages and we will be using an mbox format. Now there are at least 3-5 different mbox formats and so we'll be using a simplified version of this and I'll describe it below. The heart of this project will be a struct that you create and then 2 arrays used to store the messages. One of the arrays will store valid or good messages, and the other array will store hacked or invalid messages. Both of these arrays will need to "grow" and so we'll be using pointers and dynamic arrays to accomplish this.

Input

For this project the mailbox program will respond to a series of commands:

1. **load** this command will expect a filename to appear after it. The file will be a valid mbox file and the messages will not be checked as they are loaded. (Even though some of them may be hacked, load will put them all in the inbox.)
2. **add** this command will simulate a new message coming in. On the next line will be a single message. The message will need to be checked for validity. If it is good, then it is added to the inbox. Otherwise it is added to the hacked message inbox or list.
3. **show** this command will be followed by which inbox to show, i.e inbox or hacked, and then optionally a number. If the command is just show inbox or show hacked then the entire request list of messages will be shown. If the command is ended with a number then, if the number is good, that message is shown. IF the message number is invalid, then an error message is shown.
4. **remove** this command will be followed by which inbox you wish to remove a message from and then it will end with a message number. This would look something like this: remove inbox 3 or remove hacked 2 If the message number is valid, then the message is removed from the list and an output would indicate it's removal. If the message number is invalid then an error output is displayed.
5. **save** this command will be followed by which inbox to save and a filename where to save the inbox. For example: save inbox mboxsave or save hacked mboxhackedsave The format of the saved file is such that it could be used to load. So to test, you could add some messages, save your inbox, then load that same saved file back in.

mbox file

mbox files in general come in many flavors and varieties. I've decided to simplify the format for our purposes but it is similar to what is actually used. The messages will all be stored in the following format

From <email address>

Date <date>

To <email address>

Subject <subject>

Message

<message body>

ID <id number>

<blank line>

The words that are not in the < > will be actual words in the file. You may use those to identify the parts of the message. The message will always come in that order. The parts in the < > will change and what is used to generate the ID. The word Message will always be on a line by itself and the following message will be a single line of text, i.e. you may use getline to read it. The ID is a signed hash. This will be explained in greater detail below, but it is what we will be using to make sure that any incoming messages haven't been tampered with. If the ID in the message doesn't match the ID that we compute when we receive the message, then the assumption is the message was tampered with in transmission and should not be trusted.

Output

Each command will have a related set of output. After you read each command, echo the command and the results of the command.

1. **load**: The load command is given an mbox file to load, so the results would be the command and the file name, on the next line you'll indicate how many messages are loaded.
2. **add**: The add message is followed by a message to add. The output will indicate if the message was okay and added to the inbox, or not okay and added to the hacked list.
3. **show**: If just the list name is given then all the messages are shown for that list. If a number is also given then just the one message is shown. If the message number is invalid then an error message is given.
4. **remove**: If the message number is valid and the message is removed, then positive message is output. If the message number is invalid then a message is output about not removing the message.
5. **save**: After the mail box is saved a message is output indicating how many messages were saved.

See the attached file below for more examples.

Samples

This is a sample command file

load mbox

add

From dmcphers@vt.edu

Date 2017/03/21

To dmcphers@vt.edu

Subject Get Started on P4

Message

Hey don't put off project 4 any more. You'll thank me when you are done.

ID 3630

show inbox

remove inbox -3

save inbox mbox2

remove inbox 1

add

From dmcphers@vt.edu

Date 2017/03/21

To dmcphers@vt.edu

Subject Get Started on P4

Message

Hey don't put off project 4 any more. You'll thank me when you are done.

ID 3631

show hacked

remove hacked 0

#any line with # is a comment

#add remove good number

#add show inbox good number

#add hacked message

#add show hacked list

#add show hacked with a number

This would be the results from running the command file

Command: load mbox

Messages: 1

Command: add

Message added

Command: show inbox

From HIWTDIPPG@vt.edu

Date 2014/4/11

To RLCVAZDSA@vt.edu

Subject Let's do lunch

Message

At first she would call to him as she did so with words that she probably considered friendly, such as "come on then, you old dung-beetle!", or "look at the old dung-beetle there!" Gregor never responded to being spoken to in that way, but just remained where he was forced to impose on them. "Anna! Anna!" his father called into the kitchen through the entrance hall, clapping his hands, "get a locksmith here, now!" And the two girls, their skirts swishing, immediately ran out through the hall, wrenching open the front door of the bedroom opened and Mr. "Grete, come with us in here for a little while then as well, and Gregor's sister would leave her work to help her mother, but nothing would have any effect on him. Now Gregor's sister also had to help his mother with the music stand, his mother with the music stand, his mother with the music stand, his mother with the cooking; although that was not about him in some other food that was more suitable? If she didn't do it herself he would rather go hungry than draw her attention to it, although he did not want to think about too much, so he started to move toward her, he was slow and infirm, but it was only now out of politeness that they allowed their peace to be disturbed. Smears of dirt were left on the walls, here and there were now many such things as one of the worst of these places when his sister no longer had any particular reason to be upset, and if he finally managed to do what he wanted with that leg, all the others seemed to be set free and would move about painfully. Please don't make things any harder for me than they are already, and don't take sides against me at the office. And the injury in Gregor's back began to hurt as much as he would.

ID 4226

From dmcphers@vt.edu

Date 2017/03/21

To dmcphers@vt.edu

Subject Get Started on P4

Message

Hey don't put off project 4 any more. You'll thank me when you are done.

ID 3630

Command: remove inbox -3

Message not removed

Message -3 not a valid message number

Command: save inbox mbox2

2 Messages saved

Command: remove inbox 1

Message removed

Command: add

Message was not authentic. Sent to hacked list.

Command: show hacked

From dmcphers@vt.edu

Date 2017/03/21

To dmcphers@vt.edu

Subject Get Started on P4

Message

Hey don't put off project 4 any more. You'll thank me when you are done.

ID 3631

Command: remove hacked 0

Message removed

For the shown commands this is the mbox that was loaded. It only has a single message:

From HIWTDIPPG@vt.edu

Date 2014/4/11

To RLCVAZDSA@vt.edu

Subject Let's do lunch

Message

At first she would call to him as she did so with words that she probably considered friendly, such as "come on then, you old dung-beetle!", or "look at the old dung-beetle there!" Gregor never responded to being spoken to in that way, but just remained where he was forced to impose on them. "Anna! Anna!" his father called into the kitchen through the entrance hall, clapping his hands, "get a locksmith here, now!" And the two girls, their skirts swishing, immediately ran out through the hall, wrenching open the front door of the bedroom opened and Mr. "Grete, come with us in here for a little while then as well, and Gregor's sister would leave her work to help her mother, but nothing would have any effect on him. Now Gregor's sister also had to help his mother with the music stand, his mother with the music stand, his mother with the music stand, his mother with the cooking; although that was not about him in some other food that was more suitable? If she didn't do it herself he would rather go hungry than draw her attention to it, although he did not want to think about too much, so he started to move toward her, he was slow and infirm, but it was only now out of politeness that they allowed their peace to be disturbed. Smears of dirt were left on the walls, here and there were now many such things as one of the worst of these places when his sister no longer had any particular reason to be upset, and if he finally managed to do what he wanted with that leg, all the others seemed to be set free and would move about painfully. Please don't make things any harder for me than they are already, and don't take sides against me at the office. And the injury in Gregor's back began to hurt as much as he would.

id 4226

Signing the Messages

To sign the messages we are going to use SHA256. Linked below is a file that will do that for you. The "message" we are hashing is the combination of the from, date, to, subject and message in that order. You can combine them into a single string using +. For example:

string messageToHash = from + date + to + subject + message;

That assumes you have variables with the names from, date, to, subject, and message. Your variable names may differ. Then to sign the message you'll use the sign function that is in the same cpp file. You will use the from address as the key. The sign message will return an unsigned int. You can sign the messageToHash like this:

unsigned int signID = sign( messageToHash, from );

These functions are provided for you in the sha256.cpp and they are declared in the sha256.h file. Simply download them and put them in your directories. Then #include "sha256.h" and when you compile you'll need to compile the sha256.cpp along with your code.

When you receive a message in the add command, there is a supplied ID. That id is really the signed hash of the message. The message may have been modified and so you'll check the id by hashing the message and then signing it. If the id that you calculate doesn't match the one in the command, then the message is assumed to be invalid. That message will not be added to the inbox, but instead to the hacked messages list.

Requirements

1. You must use a struct to create the message. It must have fields for at least the from, date, to, subject, message and id. It may have other fields, but it must contain at least the ones listed here.
2. You may not use classes.
3. You must use a pointer to an array of your message struct. This array must be dynamically allocated. The initial size of the array is 10. You must double the size when you grow.
4. You may not use a vector, list or other container class from the STL.
5. You may not use global variables.
6. You must comment your code following the code style guidelines found in Canvas.
7. You must implement the function void mailbox( string input, string output ) and that function must be declared in a header called mailbox.h