**Project report**

Design and implementation of SMTP Server and POP3 Client

Name: Chenchen GAO

**I. Overview**

In this lab, the goal of this project is:

Receive Email from smtp client (thunderbird, evolution …) and forward it to destination.

Receive Email from pop3 server (pop3.sina.com, pop3.buptis.cn …) and store it as \*.eml file.

Friendly User Interface and good interactive User Experience.

**II. Requirements Analysis Development Environment**

OS is Linux ubuntu.

Programming language is C.

**Functional requirements**

***Basic functions***

● SMTP server

1. Create TCP socket listening on port 25

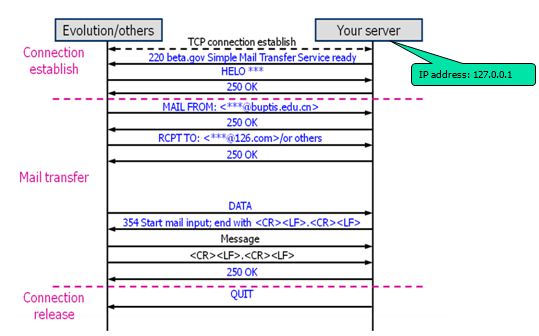
Port 25 is a well-known port for email application.

Build establish a TCP connection to port 25, the sending machine (here use Thunderbird), operating as a client, and SMTP server waits for the sending request.

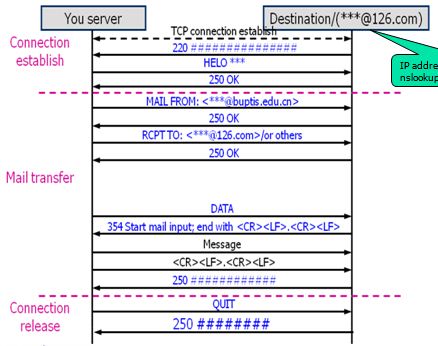
2. Receive email from Evolution, Thunderbird, Kmail or others

At this part, SMTP server is working as a server and it will receive emails which sent by a sending machine.

In this lab, we uses Thunderbird work as client and user can use to write an email, email include ‘Mail From’, ‘Send TO’, ’Subject’, ‘Data’, ’Message’, etc .When the server is ready and listen on port 25, Thunderbird can send an email through the SMTP server to the destination.



3. Forward email to the IP address of destination SMTP server



At this part, SMTP server is working as a client and it will send the received email to the destination mail server, such as @163.com, @qq.com.

When SMTP server received the email from client, it will build another connection to mail server of the destination. And the IP address that is gonging to send to coming from the ‘Send TO’ email address. After send() and recv() operation we can finally forward email to the IP address of destination SMTP server.

● POP3 client

1. Connect POP3 server on port 110

2. Guide user to login and show information about their mails

3. Display prompt characters such as “mypop >”

4. Can display content of mails in terminal

***Advanced functions***

● SMTP server

1. Authentication with username and password by BASE64 encoding

When set the Thunderbird using password model, it will send a auth login message, including user name and password. And to be sure that we have the authentication to send email, you should check that whether the name and password are match to each other.

In order to implement this function we use a function named base64(), and though it we can encode our username and password, then compare with the username and password sent form Thunderbird. After the authentication check, if it is successful, SMTP server will send the email to the destination mail server and finish its function.

2. Automatic resolve the MX record of destination domain

When we write a email, we will set ‘Send TO’. It is an email address that we will send the email to. And it can be any email server, for example, @gmail.com, @qq.com, @163.com, etc.

When we finished the basic function, the destination domain is achieved by using the command in the terminal as follow:

nslookup –query=MX 126.com

nslookup 126mx…..netease.com

And then put the result of the domain name directly into the MailserverAddr structure, so this limited SMTP server’s ability, it can only send email to the specific email server.

Automatic resolve the MX record of destination domain is that it can automatically distract the domain name from the ‘Send TO’ message, and then find the IP address of the destination email server. We use dns\_resolve() function to get the response and find\_mx() function to solve the responses, after that using a init() function to clean the name, and get the final answer. But at this time, it is only a domain name, we use a gethostbyname() function to get the hostent structure and point to the IP address in it. And use a inet\_ntoa() to get a IP address in text form.

3. Support SSL

In this lab, we are not achieved this function.

● POP3 client

1. Login with implicit password (replace your password by \*\*\*\*)

2.Can download mails and save them only on local machine (remove from remote server)

3. Provide commands set with Linux’s style such as “ls, put, get”

4. Provide function “display by subject”

5. Provide function “search text in all mails”

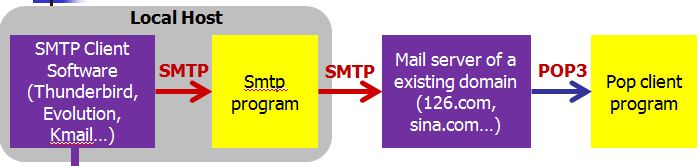
**III. Preliminary Design**

Preliminary design includes

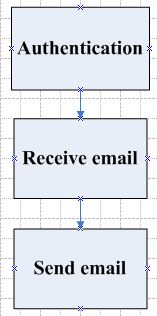
* + Decomposition of functional modules
  + Relationship and interface between the modules
  + Overall flow chart

**Decomposition of functional modules**

● SMTP



The whole SMTP server including three main part modules as shown as follow:



**Module Part I: Authentication**

This module functions as a test of authentication, if username and password are legal, then SMTP continues send email to the destination mail server.

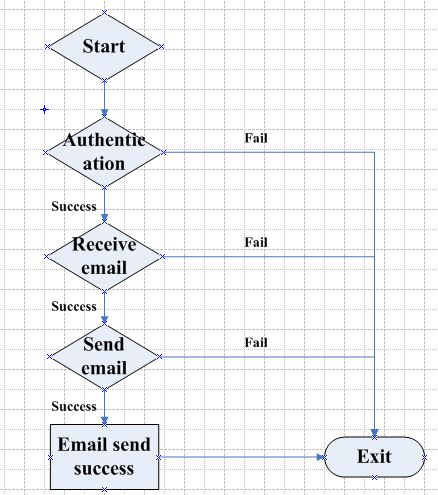
**Module Part II: Receive Email**

This module receives email from Thunderbird, including the automatic resolve the MX record of destination domain.

**Module Part III: Sending Email**

This module sends email to the destination mail server, after sending the email, it will close the connection.

**Overall flow chart**



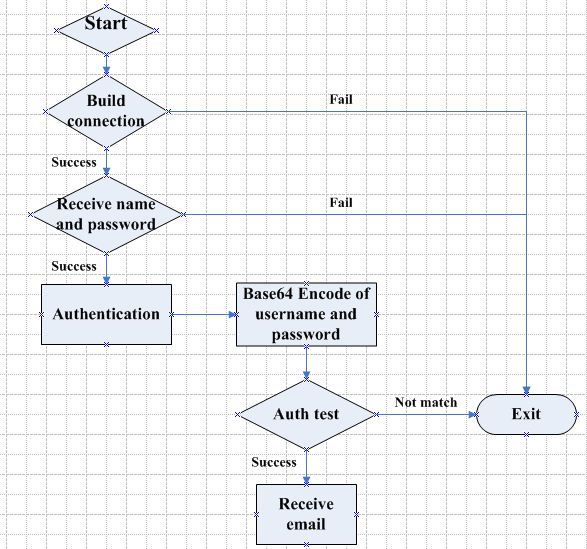
**IV. Detailed Design**

Detailed design includes

* + Design analysis of each module
  + Flow chart of each module

**1.SMTP**

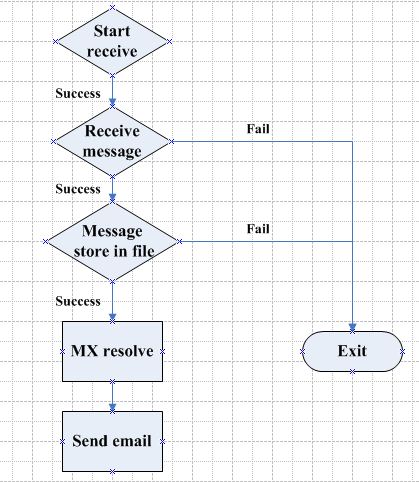
**Module Part I: Authentication**

****

When our SMTP server build a connection with Thunderbird, Thunderbird will send a auth login message which is encoded using base64, and we need to find out whether the client has the authentication to send email.

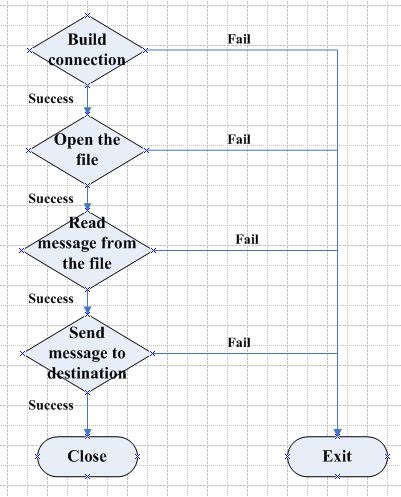
In order to implement this function we use a function named base64(), and though it we can encode our username and password, then compare with the username and password sent form Thunderbird. After the authentication check, if it is successful, SMTP server will continue its basic function, receive email and going to next part.

**Module Part II: Receive email**

****

This module receives email from Thunderbird, and store the message in a file. At the same time, we prepare the destination address for the next part. The destination address is store in a buff which receives from Tunderbird, we use a function automatic resolve the MX record of destination domain. And assign the destination address to a hostent structure, so we can use it in the third part.

**Module Part III: Send email**

****

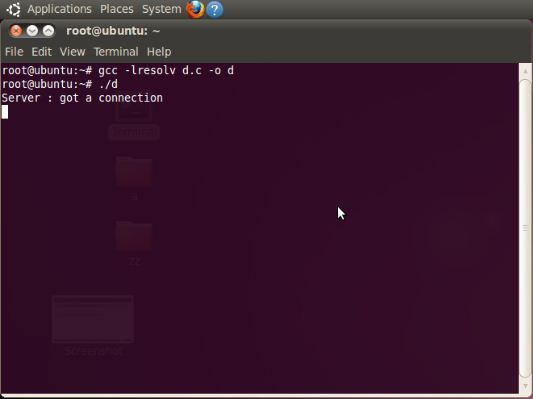
This module sends email to the destination mail server, after sending the email, it will close the connection.

First, build a connection with the destination mail server. Then we should open and read the file which contains the email message from the sender.

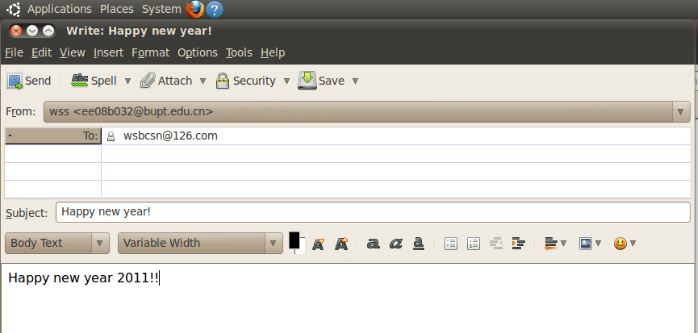
At last, send the message to the destination email server, and close the connection, and the whole SMTP process finished.

**V. Results**

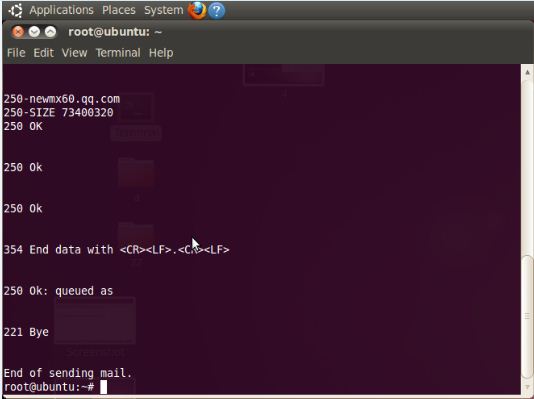
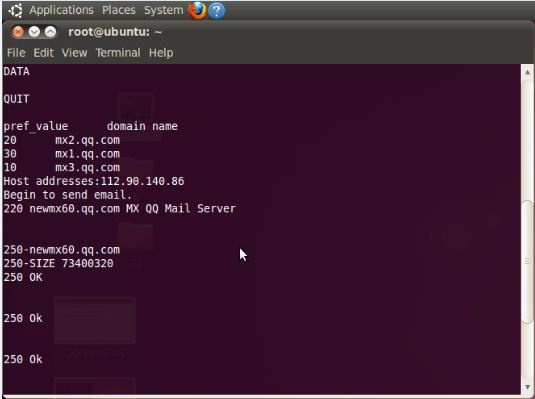
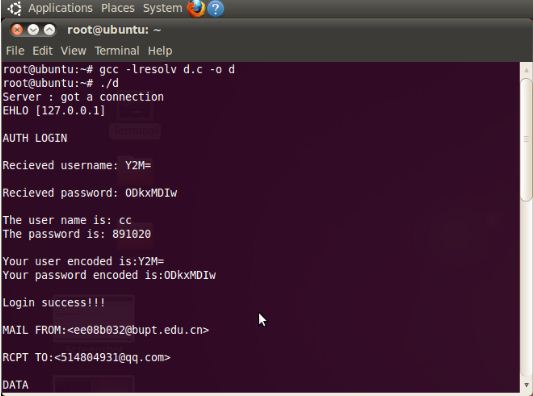
1. Compile the program, and run it, smtp server build a connection.



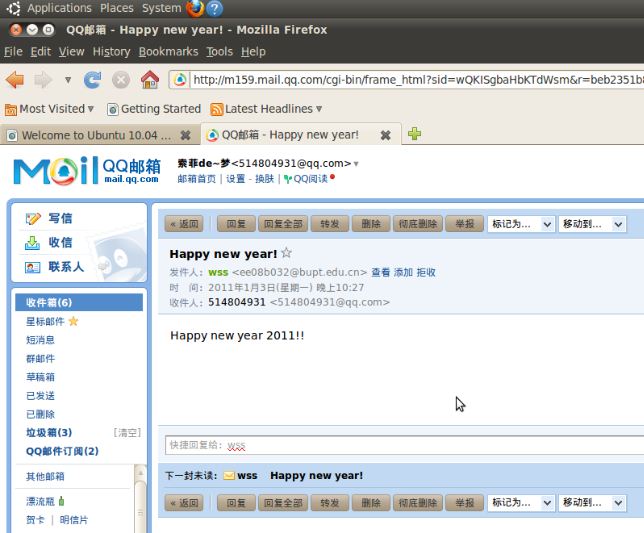
2. Use Thunderbird write an email



3. Start send the email and smtp server got some response message



4. Destination email box got the email



5. Wireshark capture the packets of the transmission

