

# CEN5515 Project No. 1: A Mail Reader

## PURPOSE

- Exercising the design and implementation of a network application.
- Experience with an application layer protocol.

## DESCRIPTION

In this project, you will implement (in C or C++) a simple mail reader, which realizes a subset of the functions in the UNIX 'mail' utility. Using the mail reader, one can read, forward, save, load, and compose/send emails. We will call the mail reader `mymail`. The command line to invoke `mymail` is:

```
mymail mail_folder mail_server_address,
```

where `mail_folder` is the mail folder file name and `mail_server_address` is the IP address of the mail server. `Mymail` supports the following commands:

```
?          print this help message
#          display message number #
-          print previous
+          print next
<CR>      print next
l folder  load the folder
h a       display all headers
h [#]     display headers around # (default current message)
f user    forward current message to user
m user    compose mail to user
s folder  save current message to folder
x         exit
```

When displaying an email message (`#`, `+`, `-`, and `<CR>` commands), `mymail` should display a selected set of header lines in addition to the message body. These header lines include **From:**, **To:**, **Subject:**, **Message-ID:**, **Date:**, and **Delivered-To:**. When displaying the headers (`h a` and `h [#]` commands), `mymail` should display in a concise manner the information about message number, size, date, sender, and subject. When forwarding a mail (`f user`), `mymail` only forwards the selected header lines and the message body. String `>` is prepended to each line in the forwarded message. The subject should be `original_subject (fwd)`. The **To:** and **From:** header lines should also be generated by `mymail`. Note that the sender information must be the current user login name at the domain of the machine running `mymail`. When composing a message, `mymail` prompts the user to input the subject line and the message body. `Mymail` should automatically add header lines for **From:** and **To:** and the message (with headers) to be transmitted in SMTP should have the following format:

```
From: xxx@xxx.xxx.xxx
To: xxx@xxx.xxx.xxx
Subject: yyyy
```

```
zzzzzzzz
zzzzzzzz
```

Finally, `mymail` allows the user to load a new mail folder and store messages in other folders. *The behavior of all commands in your implementation should be similar to that of the sample `mymail` program or the UNIX mail utility.*

The mail folder file contains one or more mails. The mails are separated by a line “From <email\_address> <Date\_information>”. See /user/spool/mail/your\_login\_name on diablo for examples. Hence, the folder format is as follows.

```
From <email_address> <Date_information>
..... (header lines)
To:.....
.....

.....
..... (body, there is an empty line between body and header)
.....
From <email_address> <Date_information>
..... (header lines)
To:.....
.....

.....
..... (body, there is an empty line between body and header)
.....
From <email_address> <Date_information>
.....
```

The mail reader needs to parse the folder file, separate the emails, compute the size for each mail, and extract header information from header lines.

## GRADING POLICY

- The program compiles without no errors (10 points)
- ?, x (10 points)
- # (10 points)
- -, +, <CR> (10 points)
- h a, h [#] (10 points)
- m user (10 points)
- f user (10 points)
- l folder, s folder (10 points)
- demonstration (20 points) (you have to design the demo to show what you have done).
- extra 10 points for programs that distinguish read/unread mails AND automatically detect new mail arrivals.
- extra 50 points for programs that can read mail from a machine that does not have access to the mail folder. You have to use Pop3s over the secure socket interface, which is not covered in the class. The program would be much longer since the secure socket interface is much more complex than the socket interface.

## **DEADLINES AND MATERIALS TO BE HANDED IN**

You will demo your project on **Sept. 29**. You must send your project directory as a tar file to xyuan@cs.fsu.edu and hand in a hard copy of the program at your demo time.

You must design the demo to show all features within the time limit. Failing to demo a feature results in losing demo points and the points for the feature.

## **MISCELLANEOUS**

All programs handed in will be checked by MOSS, a software plagiarism detection system. The cheating policy will be strictly enforced.

The project is fairly large. Start today!!!!

You can use `getenv("LOGNAME")` to get the login name and `getenv("DOMAINNAME")` to get the domain name.

You will need to use many string manipulation functions such as `strstr`, `strncmp`, `sprintf`, and `sscanf`.

The details about the email header format can be found in RFC 822.