**Topics: Vi - Vim. Mail. Script. And lots of vi!**

*Our system is in good shape now, it has the most recent updates and upgrades, all our machines have a webserver installed, and PHP, and a database server (MySQL).*

*Today, you will learn to send an email. You need to install some services to make sure your email services running perfectly.*

Start your virtual machine and log on.

Any good server can send and receive mail. This is done with a so- called Message Transfer Agent (MTA). In this case, we use postfix as MTA. Installation of software is by now one of your easiest exercises:

**sudo apt-get install postfix**

For its configuration, it will inform you about its many ways to configure it, once you read them, use <tab> to move to 'OK'. Select 'Internet Site' (because this is what we are installing in this course: a full-fledged Internet server!). Advance to 'OK' with <tab>. Then, accept the 'System mail name' as it is. 'OK' gets you back.

postfix is now being installed … .

Also, we need an agent to communicate with the mail server. We use mailx for this purpose:

**mailx**

Ooops, something is wrong here:

You are informed that the utility is not installed. It offers you one package that contains this software. Choose mailutils.

Note the command to install this package   
Test if the installation was successful by typing again:

**mailx**

Now it looks better.

Next, check if your system has all utilities necessary for mail installed.

You can try this by sending a (local) mail to the (local) user that you created in Lab exercise 2. Which was the user name? Can you still remember? If not, extract it from the userbase, /etc/passwd:

**cat /etc/passwd**

This shows quite a number of user accounts. We can also use our knowledge of pipes here: The user must have a '**home**' directory, the *shell* for additional users is **/bin/sh**, (Look up what we said about the environment and useradd in the lecture: useradd -d shows the default user environment.) and we only need to see the first element in that line:

**cat /etc/passwd | grep home | grep /bin/sh | cut -d ':' -f1**

If this is empty, and it doesn't show you any user, maybe you didn't set up a second user? Then go back to lab sheet 2 and add a second user now. (S)he is needed for our upcoming exercises!

If everything went okay, this last command shows you the additional user(s) set up on your machine. Send the mail below now (of course, you replace the 'it098765' with the user name of that second user that you found out as above):

**echo 'Hello there!' | mailx -s "Mail from your sysadmin" it098765**

Did it work? Try with (replace the 'it098765' with that user name)

**ls -l /var/mail/it098765**

If there is a file, the email system is working (This is actually the user's Inbox).

Now you ask your user to check his/her mail.

Only if the user is not around, simply create another account for your current neighbour. Refer to Lab 2 for how to do that. Only if the user doesn't remember his / her password for that account, reset it for him / her. Refer to Lab 2 for how to do that.

Let your user log in through console 2, using **Alt+F2**, entering his / her username and password. If everything worked fine, (s)he will be greeted with a line 'you have mail'. If the user types **mail**, your mail shows up.

With 'Enter' it can be read, and with '**q**' the mail utility can be exited.

Ask the user to log out.

Go back to your own, first, console: **Alt+F1**

Another helpful utility is 'script'. It came with the base install, so you can use it already.

*Here is some explanation for the working of 'script':*

*Often, we need a 'hardcopy'. You also need a hardcopy for future labs and/or assignments. You will to prove what you did, and show what you did. We as system administrators sometimes need a log to show what we did, or to remember what we did. The utility script does this for us: Everything that you type, and everything that you see on the screen, is stored in a file.*

Go to your home directory. Start the scripting by using this command (replace the 'sn012345' with **your** user name on your machine):

**script sn012345.log**

It will say something like "Script started, file is sn012345.log". Everything you type and see, will be stored until you type 'exit'. Let's go:

**whoami date pwd**

**ls -l exit**

It will say something like "exit

Script done, file is sn012345.log"

You can now look at that file, using ls -l and cat / less /more. Look carefully: what you see is the content of the script. It is suggested you type

**clear**

**cat sn012345.log**

so that you can see only the content of that script-file. Then you mail this script-file to your lab tutor, as proof:

The backslash ('\') is used here for reasons of beauty: the line is sooo looong that we rather split it. This is being done by typing a backslash '**\**' at the end, followed by **'Enter'**. You may as well leave out the backslash and simply continue typing.

**echo sn012345 | mailx -a sn012345.log-s "Lab six" \ surizal@metalab.uniten.edu.my**

Opps. It’s not work. It is because mailx didn’t have any option for you to include the attachment in your email. Can you find the option for attachment whey you type:

**man mailx**

Can you find it??? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now you need to use other services.

sudo apt-get install mutt

Send the email back using this command

**echo sn012345 | mutt surizal@metalab.uniten.edu.my -a sn012345.log -s "Lab six"**

(student id sn012345 is echo-ed and *piped* to mutt, a mail processing system. ' -a' stands for **a**ttachment, '-s' for **s**ubject.

Mutt is a text-based mail client for Unix operating systems. It was originally designed as a Mail User Agent (MUA) and relied on locally accessible mailbox and sendmail infrastructure.

You can check if the message has been received properly by metalab, by checking the log file of your mail server:

**sudo tail -n 2 /var/log/mail.log**

If you can see some "… status=sent (250 2.0.0 Ok: queued as …" it has worked.

*This is an assignment. Please make sure that your email has been received by your lab tutor to get the marks.*

*.*