



**NORTHWESTERN POLYTECHNIC**  
UNIVERSITY

# Project: Remote Calculator + Expect Cron job + Email notice

**Prepared For:**

Mr. Henry Chang  
Software Quality Assurance and Test Automation CS522  
Fall 2021  
Northwestern Polytechnic University

**Prepared By:**

Ms. Nagalla, Santhi Sree ID:19568

## Table of Contents

- *SSH Setup in Ubuntu*
- *Interactive Calculator(myscript) : Server side*
- *Client side (your local machine: Activate.sh,exp)*
- *Test the Calculator Program*
- *Set up a cron job*
- *Configure cron job to support email*
- *Installing SSMTP on Ubuntu*
- *Cron job Email Notice*
- *Removing the cron jobs*
- *References*

# SSH Setup in Ubuntu

Steps to follow to connect SSH.

1. Create the key in Local Machine using Command:

- **ssh-keygen -t rsa -f .ssh/id\_rsa**

Just press 'enter key' for “Enter passphrase”

2. Change directory to .ssh using Command:  
**cd .ssh**

```
sans@sans-vb:~$ ssh-keygen -t rsa -f .ssh/id_rsa
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in .ssh/id_rsa
Your public key has been saved in .ssh/id_rsa.pub
The key fingerprint is:
SHA256:tj04ZZQXL6olsqfUYVP7Lw14PxEdWJH/hTsfAtG5DRM sans@sans-vb
The key's randomart image is:
+---[RSA 3072]---+
|                 +E+. |
|                 =.B. |
|                 +.=0=. |
|                 ..==+0... |
|                 S=0*000.. |
|                 .0*+ .0*= |
|                 .+00  ..B+ |
|                 .. .  ..* |
|                 .o |
+---[SHA256]-----+
sans@sans-vb:~$
```

3. Below command will Create .ssh directory in remote site if it does not exist and copy the public key you created on your Machine to the remote in **authorized\_keys** File.

```
cat id_rsa.pub | ssh  
username@server "mkdir ~/.ssh; cat  
>> ~/.ssh/authorized_keys"
```

4. Login into Remote Machine using below Command.

- **ssh username@server**

```
sans@sans-vb:~/.ssh$ cat id_rsa.pub | ssh 19568@35.167.127.201 "mkdir ~/.ssh; ca  
t >> ~/.ssh/authorized_keys"  
19568@35.167.127.201's password:  
sans@sans-vb:~/.ssh$ ssh 19568@35.167.127.201  
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.4.0-1058-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Tue Nov  9 11:35:33 PST 2021  
  
System load:  0.07               Processes:            102  
Usage of /:   15.5% of 58.10GB   Users logged in:     1  
Memory usage: 28%               IP address for eth0: 172.26.3.73  
Swap usage:   0%  
  
* Ubuntu Pro delivers the most comprehensive open source security and  
compliance features.  
  
https://ubuntu.com/aws/pro  
  
Get cloud support with Ubuntu Advantage Cloud Guest:  
http://www.ubuntu.com/business/services/cloud  
  
* Canonical Livepatch is available for installation.  
- Reduce system reboots and improve kernel security. Activate at:  
https://ubuntu.com/livepatch  
  
83 packages can be updated.  
8 updates are security updates.  
  
New release '20.04.3 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Tue Nov  9 11:29:11 2021 from 98.45.168.103  
19568@CS522:~$
```

Now you are in the remote system , and change the permissions of directories or files using below Command.

- **chmod 700 ~/.ssh**

You should be able to access your server account through SSH without a password in your Local Machine.

**ssh username@server**

- **ssh username@server ls -la**

```
19568@CS522:~$ ls -la
total 44
drwxr-xr-x  8 19568 13176 4096 Nov  9 11:35 .
drwxr-xr-x 12 root   root   4096 Nov  1 16:46 ..
-rw-----  1 19568 13176 3300 Nov  9 11:36 .bash_history
drwx-----  2 19568 13176 4096 Nov  2 13:01 .cache
drwx-----  3 19568 13176 4096 Nov  2 13:01 .gnupg
drwxr-xr-x  3 19568 13176 4096 Nov  6 13:48 .local
drwxr-xr-x  2 19568 13176 4096 Nov  9 11:35 .ssh
-rw-----  1 19568 13176 2780 Nov  6 22:22 .viminfo
drwxr-xr-x  2 19568 13176 4096 Nov  2 13:31 1.ssh
drwxr-xr-x  2 19568 13176 4096 Nov  2 21:17 calculator
-rwxr-xr-x  1 19568 13176  609 Nov  6 22:22 myscript
```

```
19568@CS522:~$ chmod 700 .ssh
19568@CS522:~$ ls -la
total 44
drwxr-xr-x  8 19568 13176 4096 Nov  9 11:35 .
drwxr-xr-x 12 root   root   4096 Nov  1 16:46 ..
-rw-----  1 19568 13176 3300 Nov  9 11:36 .bash_history
drwx-----  2 19568 13176 4096 Nov  2 13:01 .cache
drwx-----  3 19568 13176 4096 Nov  2 13:01 .gnupg
drwxr-xr-x  3 19568 13176 4096 Nov  6 13:48 .local
drwx-----  2 19568 13176 4096 Nov  9 11:35 .ssh
-rw-----  1 19568 13176 2780 Nov  6 22:22 .viminfo
drwxr-xr-x  2 19568 13176 4096 Nov  2 13:31 1.ssh
drwxr-xr-x  2 19568 13176 4096 Nov  2 21:17 calculator
-rwxr-xr-x  1 19568 13176  609 Nov  6 22:22 myscript
19568@CS522:~$
```

```
sans@sans-vb:~$ ssh 19568@35.167.127.201 ls -la
total 44
drwxr-xr-x  8 19568 13176 4096 Nov  9 11:35 .
drwxr-xr-x 12 root   root   4096 Nov  1 16:46 ..
-rw-----  1 19568 13176 3300 Nov  9 11:36 .bash_history
drwx-----  2 19568 13176 4096 Nov  2 13:01 .cache
drwx-----  3 19568 13176 4096 Nov  2 13:01 .gnupg
drwxr-xr-x  3 19568 13176 4096 Nov  6 13:48 .local
drwx-----  2 19568 13176 4096 Nov  9 11:35 .ssh
-rw-----  1 19568 13176 2780 Nov  6 22:22 .viminfo
drwxr-xr-x  2 19568 13176 4096 Nov  2 13:31 1.ssh
drwxr-xr-x  2 19568 13176 4096 Nov  2 21:17 calculator
-rwxr-xr-x  1 19568 13176  609 Nov  6 22:22 myscript
sans@sans-vb:~$
```



## Interactive Calculator: Server side

- Create an **Interactive Calculator (myscript)** Program to collect the user's inputs, instead of coming from command line.
- Run the Program(**myscript**) to test in server side using below command.

■ **./myscript**

```
19568@CS522:~$ ./myscript
N1:2
N2:4
Operation(+, -, *, /):+
6
19568@CS522:~$ ./myscript
N1:9
N2:67
Operation(+, -, *, /):-
-58
19568@CS522:~$
```

```
19568@CS522:~$ cat myscript
echo -n "N1:"
read N1

echo -n "N2:"
read N2

echo -n "Operation(+, -, *, /):"
read Operation

if [ "$Operation" = "+" ]
then
    (( result = N1 + N2 ))
elif [ "$Operation" = "-" ]
then
    if [ "$N1" -ge "$N2" ]
    then
        (( result = N1 - N2 ))
    else
        (( result = N2 - N1 ))
    result="-${result}"
    fi
elif [ "$Operation" = "*" ]
then
    (( result = N1 * N2 ))
elif [ "$Operation" = "/" ]
then
    (( result1 = N1 / N2 ))
    result="${result1}"
    # (( result2 = N1 * 100 / N2 ))
    # result="${result1}.${result2}"
else
    echo "Error: wrong operation $Operation"
    exit 1
fi

echo "$result"

19568@CS522:~$
```

## Client side (your local machine)

- Create shell script(**Activate.sh**) to automatically execute the server side's program and use Expect to automate the process(**exp**) entering data to the Interactive Calculator Program.
- Use the command "shuf" to automatically generate random number from a range for N1 and N2.

### Activate.sh

```
1 #!/bin/bash
2 N1=`shuf -i 1-100 -n 1`
3 N2=`shuf -i 1-100 -n 1`
4 ans=`shuf -i 1-4 -n 1`
5
6 if [ "$ans" -eq 1 ]
7 then
8     Operation="+"
9 elif [ "$ans" -eq 2 ]
10 then
11     Operation="\-"
12 elif [ "$ans" -eq 3 ]
13 then
14     Operation="*"
15 else
16     Operation="/"
17 fi
18
19 ./exp "$N1" "$N2" "$Operation"
```

# Test the Calculator Program

- Run shell script(**Activate.sh**) to automatically execute the server side's Interactive Calculator (**myscript**) program using following command.

■ **./Activate.sh**

## exp File

exp

```
1 #!/usr/bin/expect --
2 set timeout -1
3 set N1 [lindex $argv 0]
4 set N2 [lindex $argv 1]
5 set Operation [lindex $argv 2]
6 spawn ssh 19568@35.167.127.201 /home/19568/myscript
7 expect "N1:" { send "$N1\r" }
8 expect "N2:" { send "$N2\r" }
9 expect "Operation(+, -, *, /):" { send "$Operation\r" }
10 interact
```

```
2132
sans@sans-vb:~/Desktop/calculator$ ./Activate.sh
spawn ssh 19568@35.167.127.201 /home/19568/myscript
N1:97
N2:75
Operation(+, -, *, /):*
7275
sans@sans-vb:~/Desktop/calculator$ ./Activate.sh
spawn ssh 19568@35.167.127.201 /home/19568/myscript
N1:40
N2:48
Operation(+, -, *, /):*
1920
sans@sans-vb:~/Desktop/calculator$ ./Activate.sh
spawn ssh 19568@35.167.127.201 /home/19568/myscript
N1:90
N2:34
Operation(+, -, *, /):+
124
```

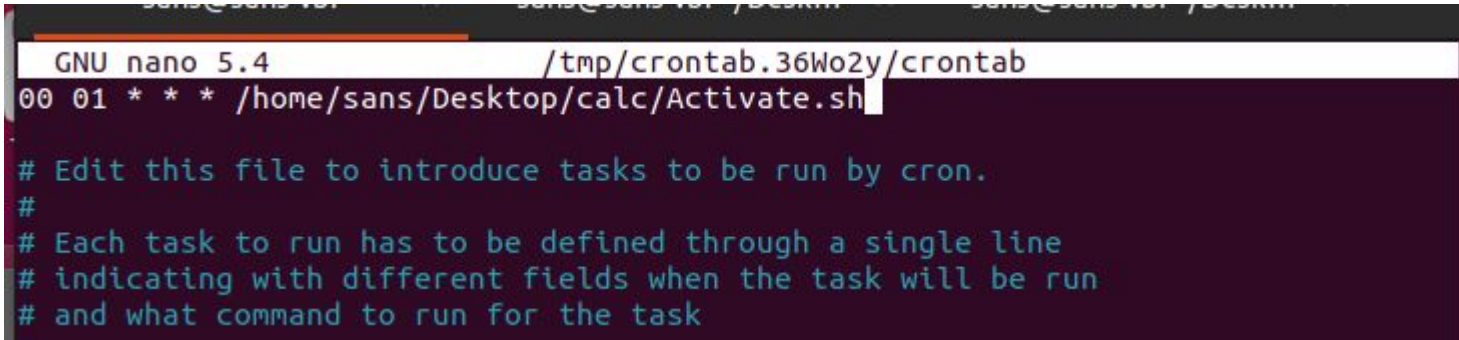


# Set up a cron job

- You can add or edit the cron jobs for the currently logged in user using below command.  
**Crontab -e**
- Schedule a cron job to run at “1.00 am” on everyday.

**00 01 \* \* \* /home/sans/Desktop/calc/Activate.sh**

Here “00 01” refers to 1.00 am



```
GNU nano 5.4 /tmp/crontab.36Wo2y/crontab
00 01 * * * /home/sans/Desktop/calc/Activate.sh

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
```

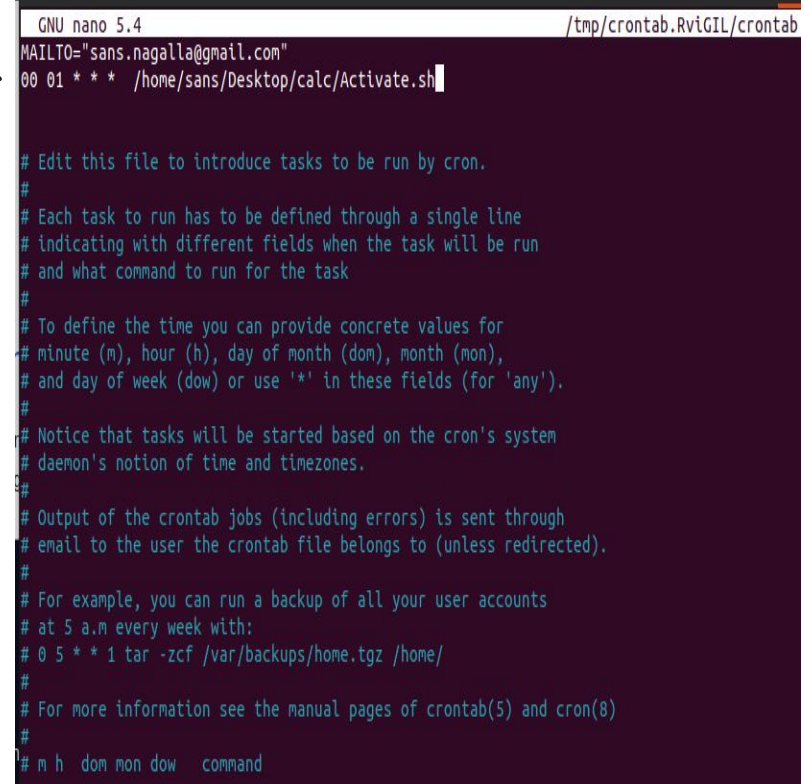
# Configure cron job to support email

- You can setup a mail address on which you want to receive the notifications of executed cron jobs using below Commands.

```
MAILTO="santhi@gmail.com"
```

```
00 01 * * * /home/sans/Desktop/calc/Activate.sh
```

- If you see “No MTA installed, discarding output” error in your syslog it means that there is output from your cron job but your local machine does not have a Message Transfer Agent installed to process the output into an email.

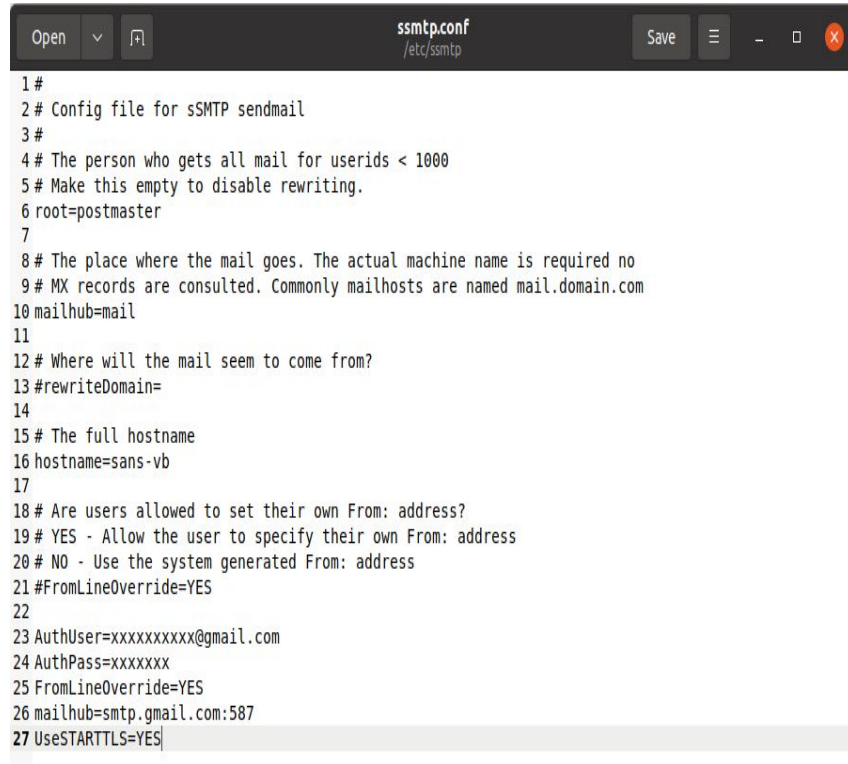


```
GNU nano 5.4 /tmp/crontab.RviGIL/crontab
MAILTO="sans.nagalla@gmail.com"
00 01 * * * /home/sans/Desktop/calc/Activate.sh

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
```

# Installing SSMTP on Ubuntu

- The installation process for this utility is pretty simple, use following command to install it.
  - **sudo apt-get install ssmtp**
- Let's edit “/etc/ssmtp/ssmtp.conf” file using any text editor application (Vi, Vim, Pico, Nano) , and specify the value for all parameters as shown.

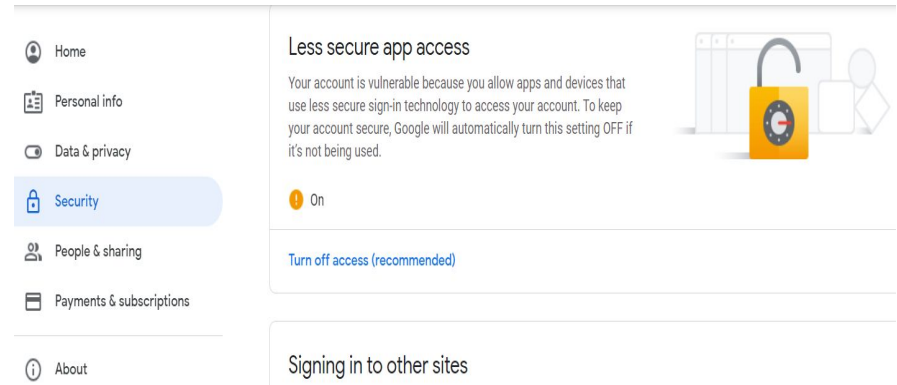
A screenshot of a text editor window with a dark theme. The title bar shows 'ssmtp.conf' and the file path '/etc/ssmtp'. The editor contains a configuration file with 27 lines of text, mostly in a comment format (starting with '#'). The parameters being configured include mailhub, mailhub=mail, where the mail will seem to come from, rewriteDomain, full hostname (sans-vb), whether users can set their own From: address (set to NO), AuthUser (xxxxxxx@gmail.com), AuthPass (xxxxxxx), FromLineOverride (YES), mailhub (smtp.gmail.com:587), and UseSTARTTLS (YES).

```
1 #
2 # Config file for sSMTP sendmail
3 #
4 # The person who gets all mail for userids < 1000
5 # Make this empty to disable rewriting.
6 root=postmaster
7
8 # The place where the mail goes. The actual machine name is required no
9 # MX records are consulted. Commonly mailhosts are named mail.domain.com
10 mailhub=mail
11
12 # Where will the mail seem to come from?
13 #rewriteDomain=
14
15 # The full hostname
16 hostname=sans-vb
17
18 # Are users allowed to set their own From: address?
19 # YES - Allow the user to specify their own From: address
20 # NO - Use the system generated From: address
21 #FromLineOverride=YES
22
23 AuthUser=xxxxxxx@gmail.com
24 AuthPass=xxxxxxx
25 FromLineOverride=YES
26 mailhub=smtp.gmail.com:587
27 UseSTARTTLS=YES
```

# Verify email

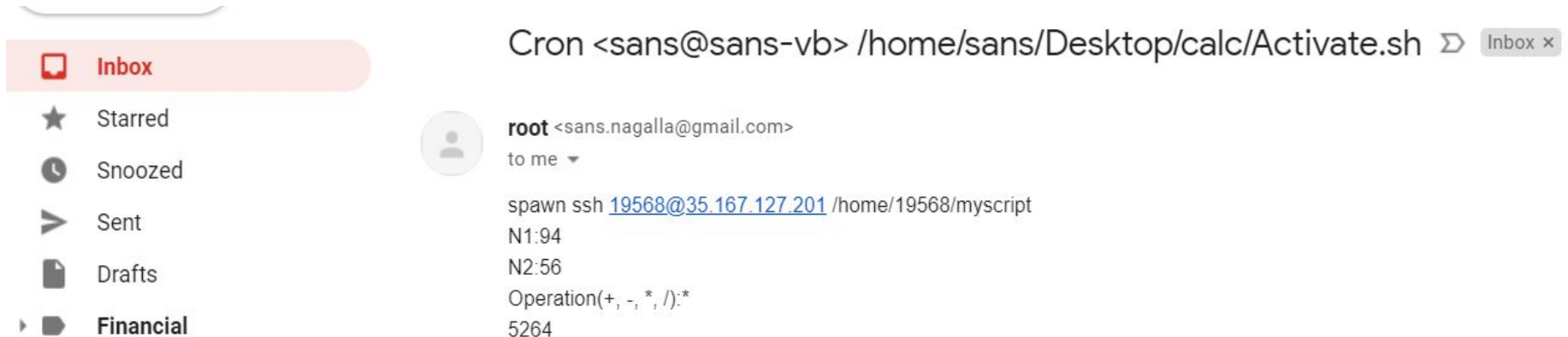
- Save the configuration file and you are good to go.
- If you are using Google account you should turn on the option in google account under security “Less secure app access”.
- Once SSMTTP has been configured, simply run following command on the terminal to verify that email is actually getting forwarded to the external mailhub.

```
echo "Testing" | mail -s "Test Email"  
santhi@gmail.com
```



# Cron job Email Notice

- Still if you see any issues to send out emails follow the link below.
  - <https://www.unixmen.com/install-ssmtp-to-send-emails-to-gmail-and-office3655/>
- After configuring the email, wait until the cron job triggers and mail is received.





# Removing the cron jobs

- The following command will list the scheduled cron jobs for the currently logged in user:

## Crontab -l

- If you want to remove the cron jobs of currently logged in user, execute the following command:

## Crontab -r

```
crontab: installing new crontab
sans@sans-vb:~$ crontab -l
MAILTO="sans.nagalla@gmail.com"
00 01 * * * /home/sans/Desktop/calc/Activate.sh

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
```

```
# M T W T F S S dom MON dow  COMMAND
sans@sans-vb:~/Desktop/calc$ crontab -r
sans@sans-vb:~/Desktop/calc$ crontab -l
no crontab for sans
sans@sans-vb:~/Desktop/calc$
```

# References

- GitHub URL - <https://github.com/santhinagalla/Software-Quality-Assurance-and-Test-Automation/tree/main/Shell%20Script/Project>
- [https://npu85.npu.edu/~henry/npu/classes/shell\\_script/backup/slide/expect.html#calculator](https://npu85.npu.edu/~henry/npu/classes/shell_script/backup/slide/expect.html#calculator)
- [https://npu85.npu.edu/~henry/npu/classes/qa/cron\\_job/slide/cron.html#email](https://npu85.npu.edu/~henry/npu/classes/qa/cron_job/slide/cron.html#email)
- <https://www.unixmen.com/install-ssmtp-to-send-emails-to-gmail-and-office3655/>
- <https://stackoverflow.com/questions/2556190/random-number-from-a-range-in-a-bash-script>