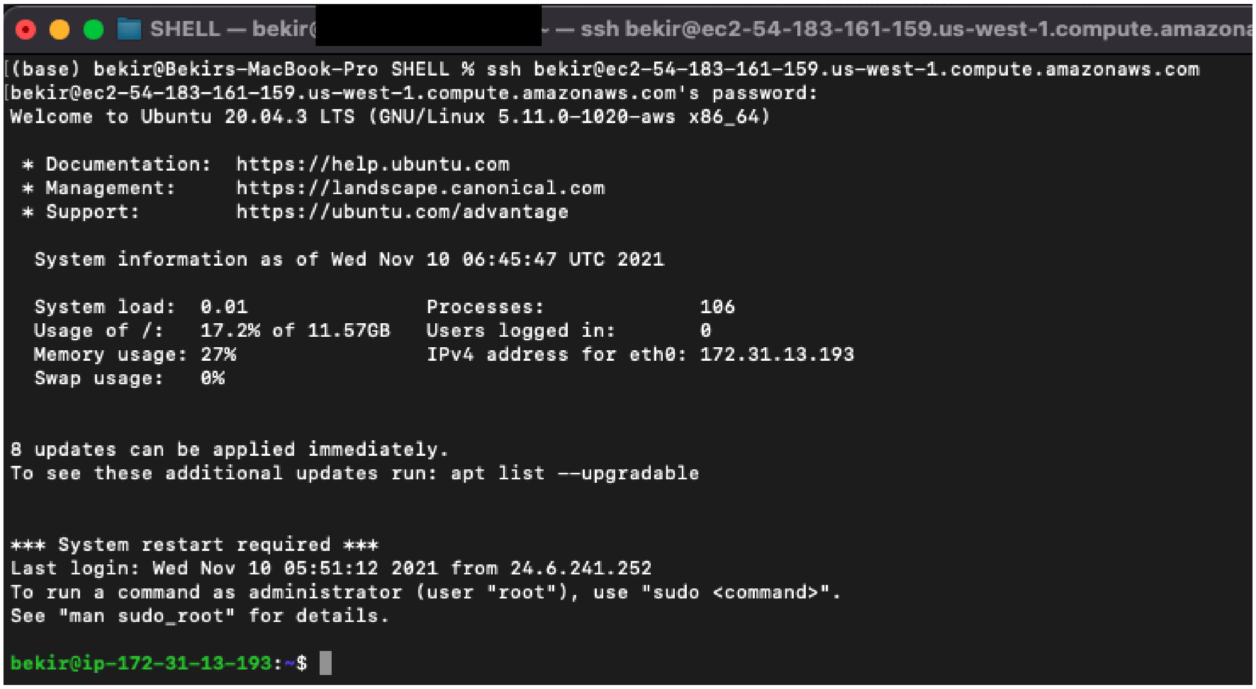
Running a shell script on a remote machine and interacting with it

This document shows how to run a script on a remote server.

Step-1 : Connecting to remote server with ssh

```
ssh bekir@ec2-54-183-161-159.us-west-1.compute.amazonaws.com
```



SHELL — bekir@[redacted]- — ssh bekir@ec2-54-183-161-159.us-west-1.compute.amazona

```
[(base) bekir@Bekirs-MacBook-Pro SHELL % ssh bekir@ec2-54-183-161-159.us-west-1.compute.amazonaws.com
[bekir@ec2-54-183-161-159.us-west-1.compute.amazonaws.com's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Nov 10 06:45:47 UTC 2021

  System load:  0.01               Processes:             106
  Usage of /:   17.2% of 11.57GB   Users logged in:       0
  Memory usage: 27%                IPv4 address for eth0: 172.31.13.193
  Swap usage:   0%


8 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


*** System restart required ***
Last login: Wed Nov 10 05:51:12 2021 from 24.6.241.252
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

bekir@ip-172-31-13-193:~$ 
```

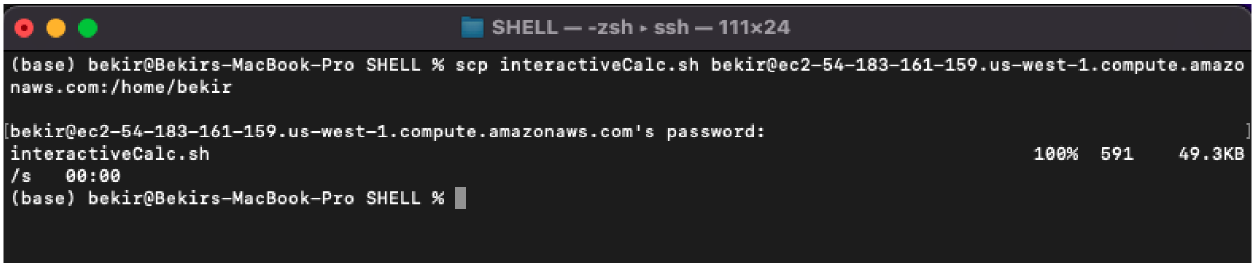
Step-2 : Testing the connection by copy a file to remote desktop

I copied a shell script by using scp to test the connection.

Command:

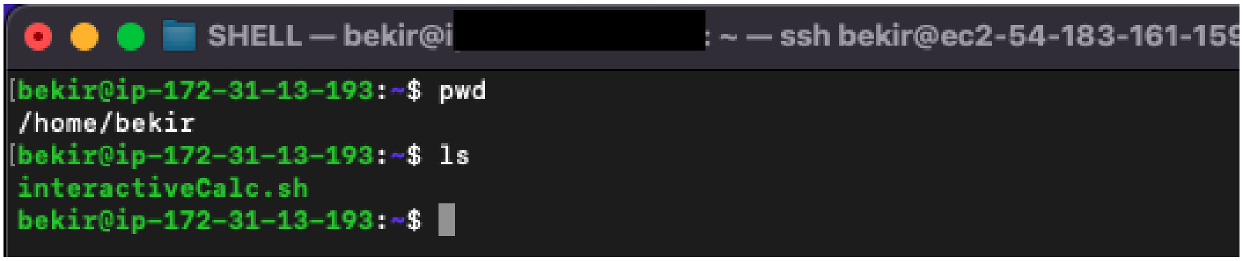
```
scp interactiveCalc.sh bekir@ec2-5-18-161-159.us-west-1.compute.amazonaws.com:/home/bekir
```

Local Machine:



```
(base) bekir@Bekirs-MacBook-Pro SHELL % scp interactiveCalc.sh bekir@ec2-54-183-161-159.us-west-1.compute.amazo
naws.com:/home/bekir

[bekir@ec2-54-183-161-159.us-west-1.compute.amazonaws.com's password:                                          ]
interactiveCalc.sh                                                                        100%  591     49.3KB
/s   00:00
(base) bekir@Bekirs-MacBook-Pro SHELL %
```

Remote Machine:



```
[bekir@ip-172-31-13-193:~$ pwd
/home/bekir
[bekir@ip-172-31-13-193:~$ ls
interactiveCalc.sh
bekir@ip-172-31-13-193:~$
```

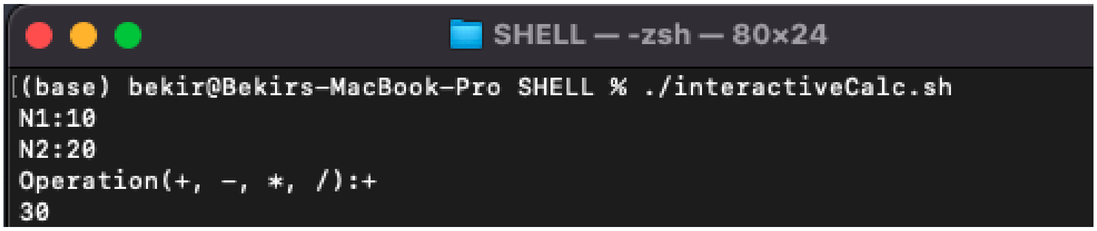
Step-3 : Calculator Script

InteractiveCalc.sh:

```
##!/bin/bash
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 [ "$Operation" = "/" ]
then
   (( result1 = N1 / N2 ))
   (( result2 = N1 * 100 / N2 ))
   result="${result1}.${result2}"
else
   echo "Error: wrong operation $Operation"
   exit 1
fi

echo $result
```

Testing the InteractiveCalc.sh on the local machine:



```
[(base) bekir@Bekirs-MacBook-Pro SHELL % ./interactiveCalc.sh
 N1:10
 N2:20
 Operation(+, -, *, /):+
 30
```

Step-4 : Making Calculator Script Interactive by using Expect Script

In this step, I created two files on the local machine. One file called ServerTriger.sh for generating random numbers and operations. Then called runScriptOnServer.exp that sends a command to the remote machine and provides credentials to ssh to the remote machine. Also interact with the calculator.

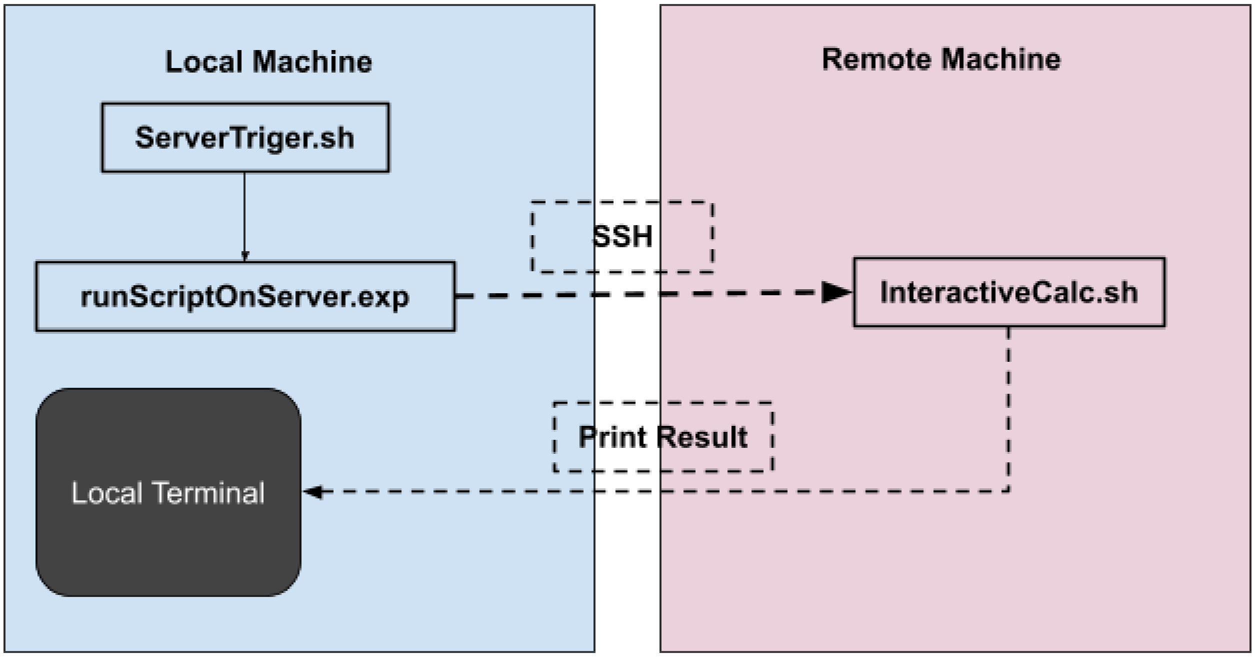
ServerTriger.sh

```
#!/bin/bash
N1=$(shuf -i 1-1000 -n 1)
N2=$(shuf -i 1-1000 -n 1)
ans=$(shuf -i 1-4 -n 1)

if [ "$ans" -eq 1 ]
then
  operation="+"
elif [ "$ans" -eq 2 ]
then
  operation="\-"
elif [ "$ans" -eq 3 ]
then
  operation="\*"
else
  operation="/"
fi
expect runScriptOnServer.exp $N1 $N2 "$operation"
```

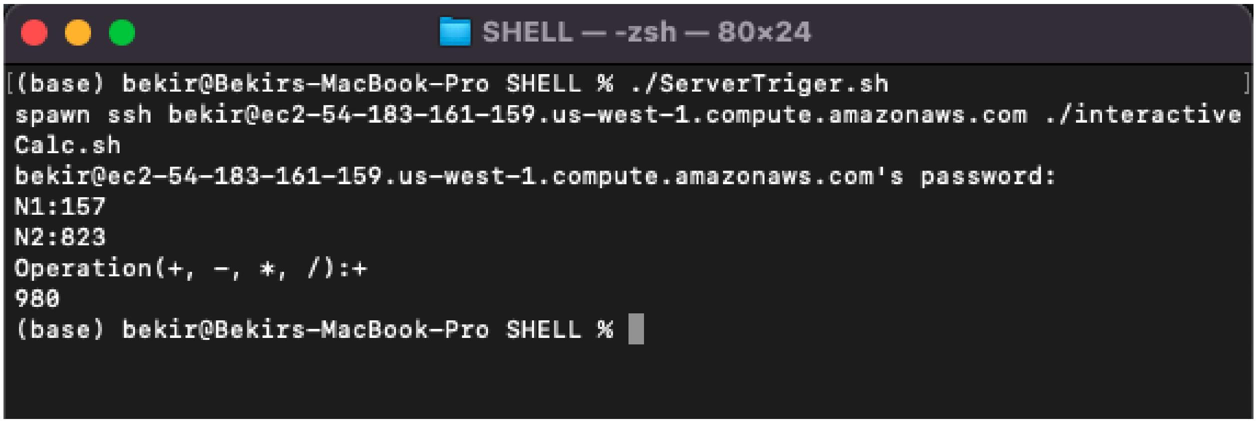
runScriptOnServer.exp

```
set timeout -1
set N1 [lindex $argv 0]
set N2 [lindex $argv 1]
set operation [lindex $argv 2]
spawn ssh bekir@ec2-5-13-161-159.us-west-1.compute.amazonaws.com ./interactiveCalc.sh
expect "bekir@ec2-5-13-161-159.us-west-1.compute.amazonaws.com's password:" { send "Password\r" }

expect "N1:" { send "$N1\r" }
expect "N2:" { send "$N2\r" }
expect "Operation(+, -, *, /):" { send "$operation\r" }
interact
```

System looks like the figure below:



Here is the result if we run ./ServerTriger.sh manually on local machine:



```
[(base) bekir@Bekirs-MacBook-Pro SHELL % ./ServerTriger.sh                      ]
 spawn ssh bekir@ec2-54-183-161-159.us-west-1.compute.amazonaws.com ./interactive
 Calc.sh
 bekir@ec2-54-183-161-159.us-west-1.compute.amazonaws.com's password:
 N1:157
 N2:823
 Operation(+, -, *, /):+
 980
 (base) bekir@Bekirs-MacBook-Pro SHELL % █
```

Step-5 : Setting up a cron job to automate the process

Cron job is needed to be created to automate the process at 1 am everyday and send an email notice.

Creating a cron job command:

```
crontab -e
```

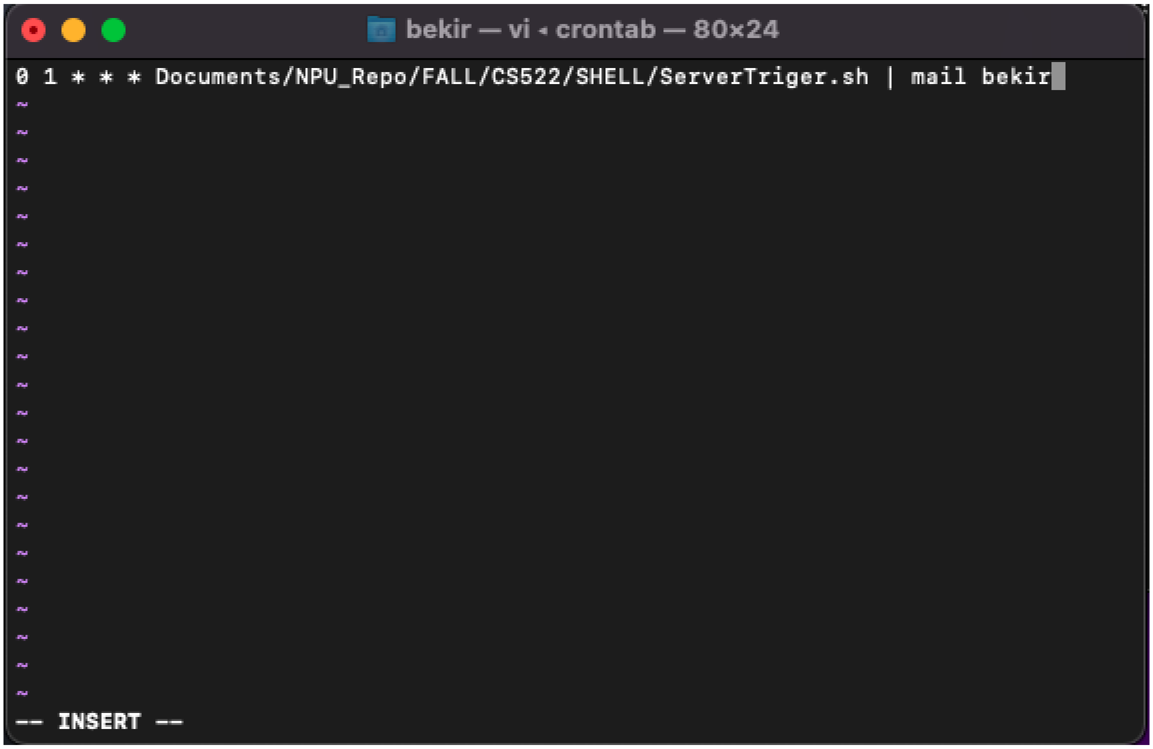
Listing the current cron jobs command:

```
crontab -l
```

Creating cron job at 1 am everyday and send mail to a local user

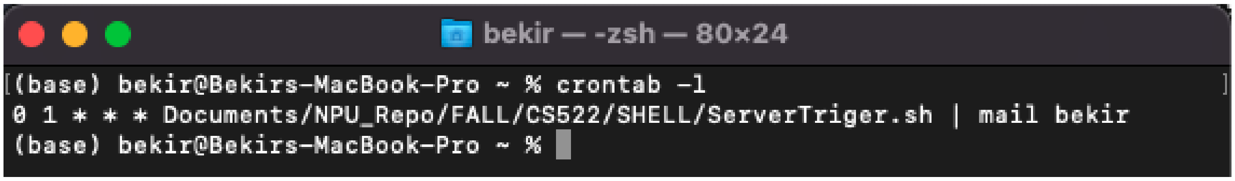
```
0 1 * * * Documents/NPU_Repo/FALL/CS522/SHELL/ServerTriger.sh
| mail bekir
```

Creating a cron job:



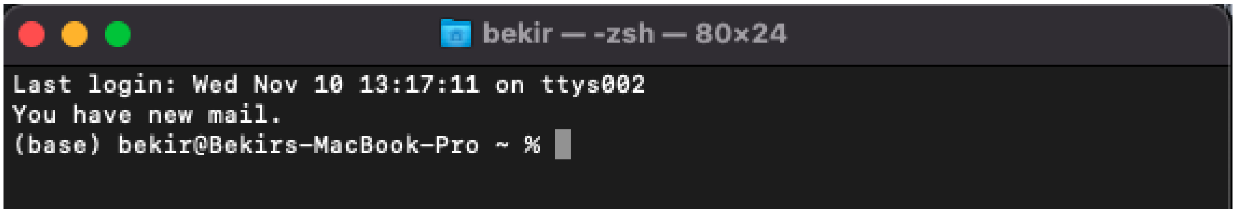
```
0 1 * * * Documents/NPU_Repo/FALL/CS522/SHELL/ServerTriger.sh | mail bekir
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

Listing cron jobs:



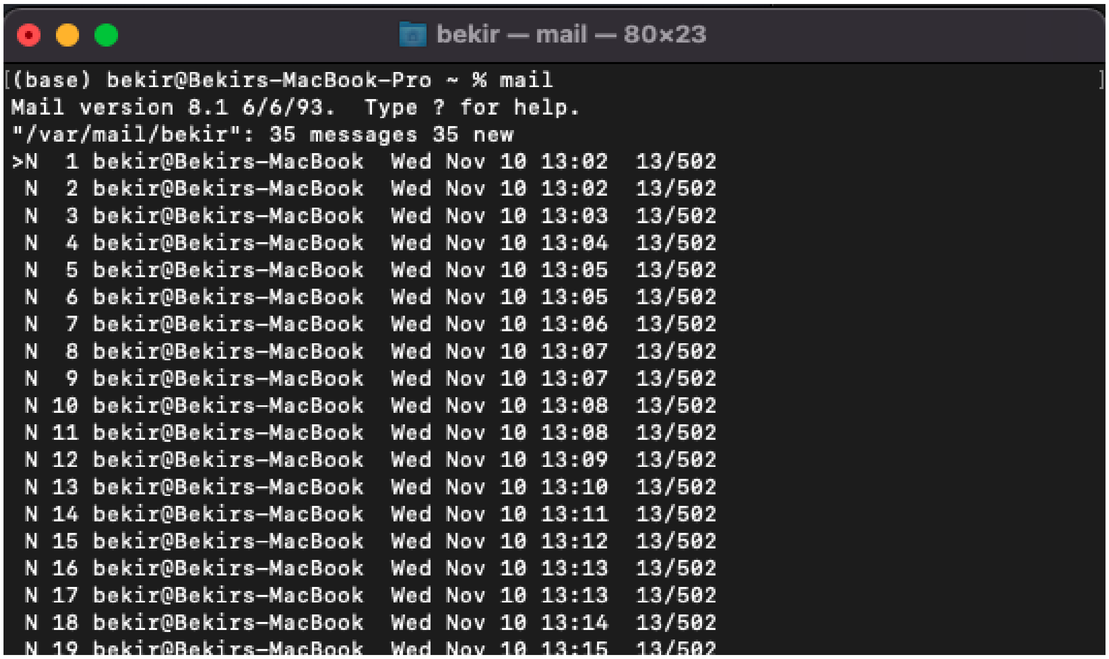
```
[(base) bekir@Bekirs-MacBook-Pro ~ % crontab -l
0 1 * * * Documents/NPU_Repo/FALL/CS522/SHELL/ServerTriger.sh | mail bekir
(base) bekir@Bekirs-MacBook-Pro ~ %
```

After the job ran. It will send an email to the user.

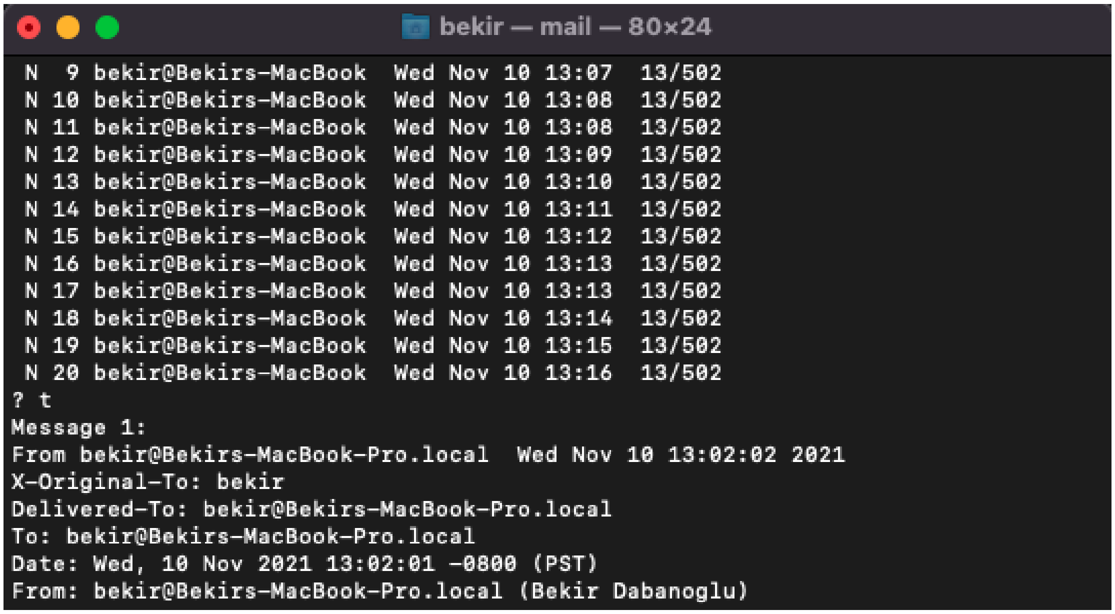


```
Last login: Wed Nov 10 13:17:11 on ttys002
You have new mail.
(base) bekir@Bekirs-MacBook-Pro ~ %
```

Command mail will bring all system mails. If we want to read the content of the mail we need to type -t after mail command:



```
[(base) bekir@Bekirs-MacBook-Pro ~ % mail
Mail version 8.1 6/6/93.  Type ? for help.
"/var/mail/bekir": 35 messages 35 new
>N  1 bekir@Bekirs-MacBook  Wed Nov 10 13:02  13/502
 N  2 bekir@Bekirs-MacBook  Wed Nov 10 13:02  13/502
 N  3 bekir@Bekirs-MacBook  Wed Nov 10 13:03  13/502
 N  4 bekir@Bekirs-MacBook  Wed Nov 10 13:04  13/502
 N  5 bekir@Bekirs-MacBook  Wed Nov 10 13:05  13/502
 N  6 bekir@Bekirs-MacBook  Wed Nov 10 13:05  13/502
 N  7 bekir@Bekirs-MacBook  Wed Nov 10 13:06  13/502
 N  8 bekir@Bekirs-MacBook  Wed Nov 10 13:07  13/502
 N  9 bekir@Bekirs-MacBook  Wed Nov 10 13:07  13/502
 N 10 bekir@Bekirs-MacBook  Wed Nov 10 13:08  13/502
 N 11 bekir@Bekirs-MacBook  Wed Nov 10 13:08  13/502
 N 12 bekir@Bekirs-MacBook  Wed Nov 10 13:09  13/502
 N 13 bekir@Bekirs-MacBook  Wed Nov 10 13:10  13/502
 N 14 bekir@Bekirs-MacBook  Wed Nov 10 13:11  13/502
 N 15 bekir@Bekirs-MacBook  Wed Nov 10 13:12  13/502
 N 16 bekir@Bekirs-MacBook  Wed Nov 10 13:13  13/502
 N 17 bekir@Bekirs-MacBook  Wed Nov 10 13:13  13/502
 N 18 bekir@Bekirs-MacBook  Wed Nov 10 13:14  13/502
 N 19 bekir@Bekirs-MacBook  Wed Nov 10 13:15  13/502
```



```
 N  9 bekir@Bekirs-MacBook  Wed Nov 10 13:07  13/502
 N 10 bekir@Bekirs-MacBook  Wed Nov 10 13:08  13/502
 N 11 bekir@Bekirs-MacBook  Wed Nov 10 13:08  13/502
 N 12 bekir@Bekirs-MacBook  Wed Nov 10 13:09  13/502
 N 13 bekir@Bekirs-MacBook  Wed Nov 10 13:10  13/502
 N 14 bekir@Bekirs-MacBook  Wed Nov 10 13:11  13/502
 N 15 bekir@Bekirs-MacBook  Wed Nov 10 13:12  13/502
 N 16 bekir@Bekirs-MacBook  Wed Nov 10 13:13  13/502
 N 17 bekir@Bekirs-MacBook  Wed Nov 10 13:13  13/502
 N 18 bekir@Bekirs-MacBook  Wed Nov 10 13:14  13/502
 N 19 bekir@Bekirs-MacBook  Wed Nov 10 13:15  13/502
 N 20 bekir@Bekirs-MacBook  Wed Nov 10 13:16  13/502
? t
Message 1:
From bekir@Bekirs-MacBook-Pro.local  Wed Nov 10 13:02:02 2021
X-Original-To: bekir
Delivered-To: bekir@Bekirs-MacBook-Pro.local
To: bekir@Bekirs-MacBook-Pro.local
Date: Wed, 10 Nov 2021 13:02:01 -0800 (PST)
From: bekir@Bekirs-MacBook-Pro.local (Bekir Dabanoglu)
```

Deleting specific message:

```
d [message number]
```