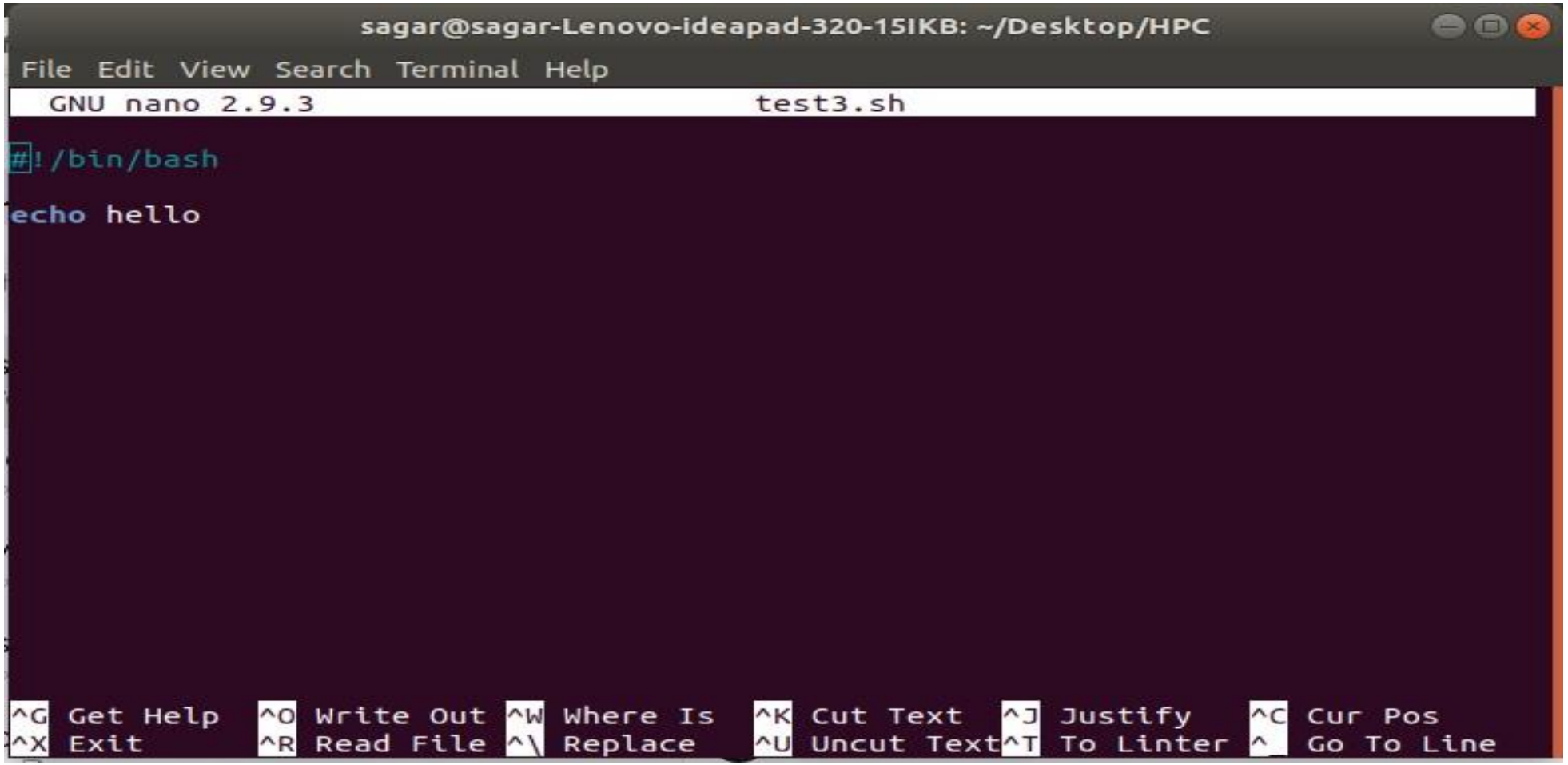
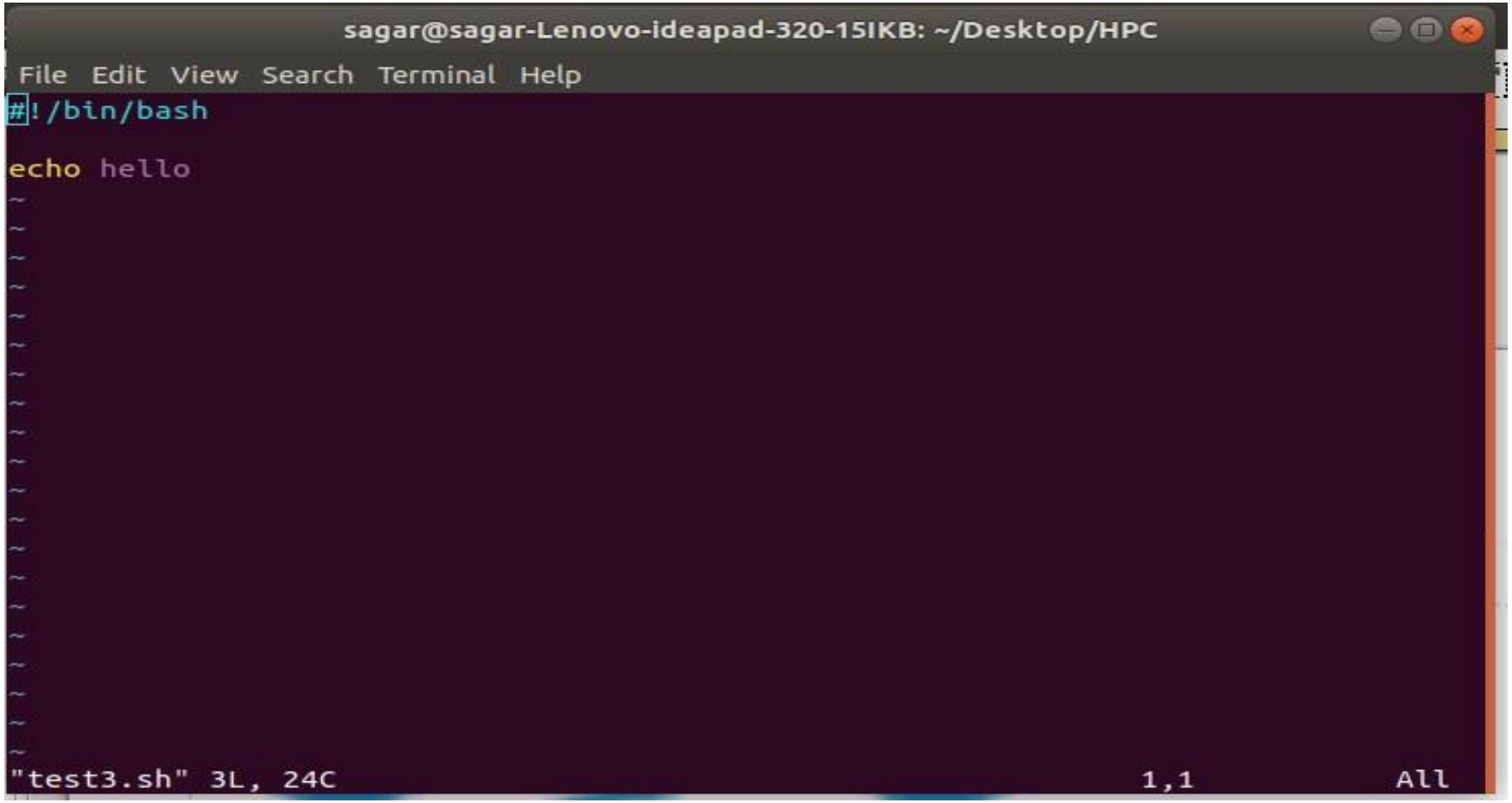
Some other useful commands and editors

Different editors and their role

- Vi
- Vim
- Nano
- Gedit
- Sublime


sagar@sagar-Lenovo-ideapad-320-15IKB: ~/Desktop/HPC
File Edit View Search Terminal Help
GNU nano 2.9.3 test3.sh
#!/bin/bash
echo hello
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Linter ^_ Go To Line


sagar@sagar-Lenovo-ideapad-320-15IKB: ~/Desktop/HPC
File Edit View Search Terminal Help
#!/bin/bash
echo hello
"test3.sh" 3L, 24C
1,1
All


~/Desktop/sh1.sh - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
sh1.sh
1 #!/bin/bash
2
3 echo hello
Line 3, Column 11
Tab Size: 4
Bourne Again Shell (bash)

Getting information about users

- `who`
  Lists all the users logged on the system.
- `whoami`
  Tells what user I am logged as.
- `groups`
  Tells which groups I belong to.
- `groups <user>`
  Tells which groups `<user>` belongs to.
- `finger <user>`
  Tells more details (real name, etc) about `<user>`
  Disabled in some systems (security reasons).

Process

- **Process**
  A process is compiled source code that is currently running on the system.
- **PID**
  All processes have a process id or PID.
- **init**
  The init process always has process ID 1. The init process is started by the kernel itself so technically it does not have a parent process. init serves as a foster parent for orphaned processes.
- **kill**

  When a process stops running, the process dies, when you want a process to die, you kill it.

Process

- **ps**-To find the processes
- **ps fx** -to find processes with relations
- **top**-order processes according to cpu usage or other properties

You can also kill processes from within top.

Press h inside top for help.

- **kill pid**

try:

sleep 1000 &
pidof sleep
kill pid

Disk devices commands

- Random access hard disk devices have an abstraction layer called block device to enable formatting in fixed-size (usually 512 bytes) blocks.
- Blocks can be accessed independent of access to other blocks.

- **lsblk** (information about all or specified block devices)
- **ls -l /dev/sd*** (all the device list starting with 'sd'.)

- discovering disk devices
- **sudo fdisk -l** (list of fixed disks)
- **sudo fdisk -l | grep Disk** (list of fixed disks with pattern disk)
- **sudo fdisk -l | grep 'Disk /dev/sd'** (list of fixed disks with pattern 'Disk /dev/sd')
- **sudo lshw** (Total h/w information)
- **lscpu** (cpu architecture information)

Vodafone
India
Foundation

_VOIS

Connecting Dreams

Shell Scripting

Contents

- What is a Shell Scripting?
- Shell scripting
- Small example of shell script automation

What is difference between command and shell scripting?

command is (computing) a directive to a computer program acting as an interpreter of some kind, in order to perform a specific task while script is (computing) a file containing a list of user commands, allowing them to be invoked once to execute in sequence.[1]

shell scripting

Shell scripting is the most useful and powerful feature in Linux

Minimizes typing of repetitive command
Can schedule jobs to run in the system
Can initiate back up activities for system administration
Similar to batch files in DOS, but more powerful than Batch files

Open a file with extension .sh using nano editor, we can type any number of commands that we use to type at command prompt

Save the file

Execute the file

sh file.sh
./file.sh (if the file has execution permission)
#!/bin/bash - Specify interpreter

How to make a script executable?

```
-rwxr-xr-x 1 sagar sagar    19 Jul 22 23:22 test2.sh.save
-rw-r--r-- 1 sagar sagar    24 Jul 23 10:17 test3.sh
drwxr-xr-x 2 sagar sagar  4096 Jul 13 15:49 testdir
drwxr-xr-x 3 sagar sagar  4096 Jul 13 16:22 testdir2
-rw-r--r-- 1 sagar sagar     0 Jul 22 12:21 testfile.txt
-rwxr-xr-x 1 sagar sagar    84 Jul 22 21:59 test.sh
-rw-r--r-- 1 sagar sagar 10240 Jul 21 23:34 test.tar
-rw-r--r-- 1 sagar sagar 20480 Jul 21 23:30 text0.tar
sagar@sagar-Lenovo-ideapad-320-15IKB:~/Desktop/HPC$ sudo chmod 755 test3.sh
[sudo] password for sagar:
Sorry, try again.
[sudo] password for sagar:
sagar@sagar-Lenovo-ideapad-320-15IKB:~/Desktop/HPC$ ls -l
total 136
drwxr-xr-x 3 sagar sagar  4096 Jul 21 21:09 hpc
drwxr-xr-x 3 sagar sagar  4096 Jul 21 21:10 lmg
drwxr-xr-x 3 sagar sagar  4096 Jul 22 12:27 sup
-rwxrwxrwx 1 sagar sagar    12 Jul 22 12:59 t11.txt
-rw-r--r-- 1 sagar sagar     8 Jul 22 12:24 t13.txt
-rw-r--r-- 1 sagar sagar     8 Jul 22 13:01 t15.txt
---xrwxrwx 1 sagar sagar     7 Jul 21 21:14 te2.txt
-rw-rw-rw- 1 sagar sagar     6 Jul 14 16:46 te3.txt
-rw-r--r-- 1 sagar sagar    53 Jul 21 23:57 te4.txt
-rw-r--r-- 1 sagar sagar 10240 Jul 21 23:29 te5.txt
-rw-r--r-- 1 sagar sagar    16 Jul 21 21:07 te6.txt
-rw-r--r-- 1 sagar sagar    12 Jul 21 22:35 te7.txt
-rw-r--r-- 1 sagar sagar    12 Jul 21 22:36 te8.txt
-rw-r--r-- 1 sagar sagar 20480 Jul 22 13:22 test
-rwxr-xr-x 1 sagar sagar    94 Jul 23 01:58 test2.sh
-rwxr-xr-x 1 sagar sagar    19 Jul 22 23:22 test2.sh.save
-rwxr-xr-x 1 sagar sagar    24 Jul 23 10:17 test3.sh
drwxr-xr-x 2 sagar sagar  4096 Jul 13 15:49 testdir
drwxr-xr-x 3 sagar sagar  4096 Jul 13 16:22 testdir2
-rw-r--r-- 1 sagar sagar     0 Jul 22 12:21 testfile.txt
-rwxr-xr-x 1 sagar sagar    84 Jul 22 21:59 test.sh
-rw-r--r-- 1 sagar sagar 10240 Jul 21 23:34 test.tar
-rw-r--r-- 1 sagar sagar 20480 Jul 21 23:30 text0.tar
sagar@sagar-Lenovo-ideapad-320-15IKB:~/Desktop/HPC$ ./test3.sh
hello
sagar@sagar-Lenovo-ideapad-320-15IKB:~/Desktop/HPC$
```

Comparison

For integer comparison we have the following

-eq : equal to
-ne : not equal to
-lt : less than
-gt : greater than
-le : less than or equal to
-ge : greater than or equal to

For string comparison we have

= : equal to
~= : not equal to

For logical operators

-a : AND
-o : OR

Shell scripting

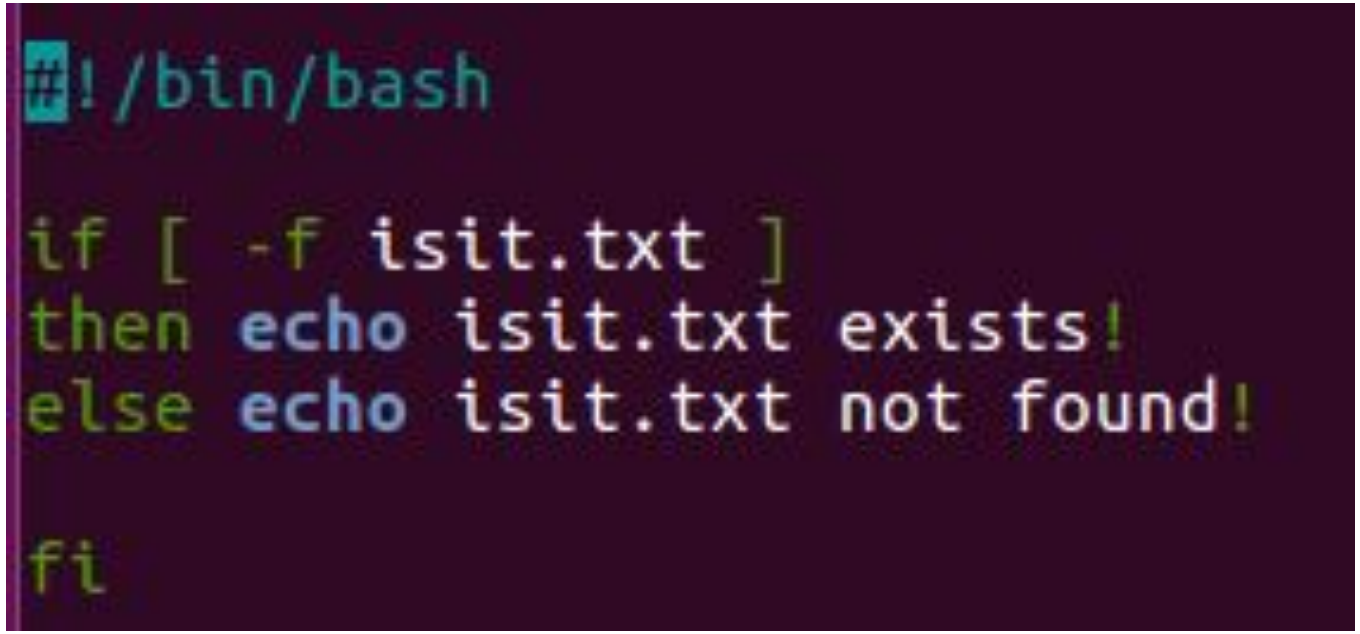
- write a shell script to use if then else

#!/bin/bash

if [-f isit.txt]

then echo isit.txt exists!

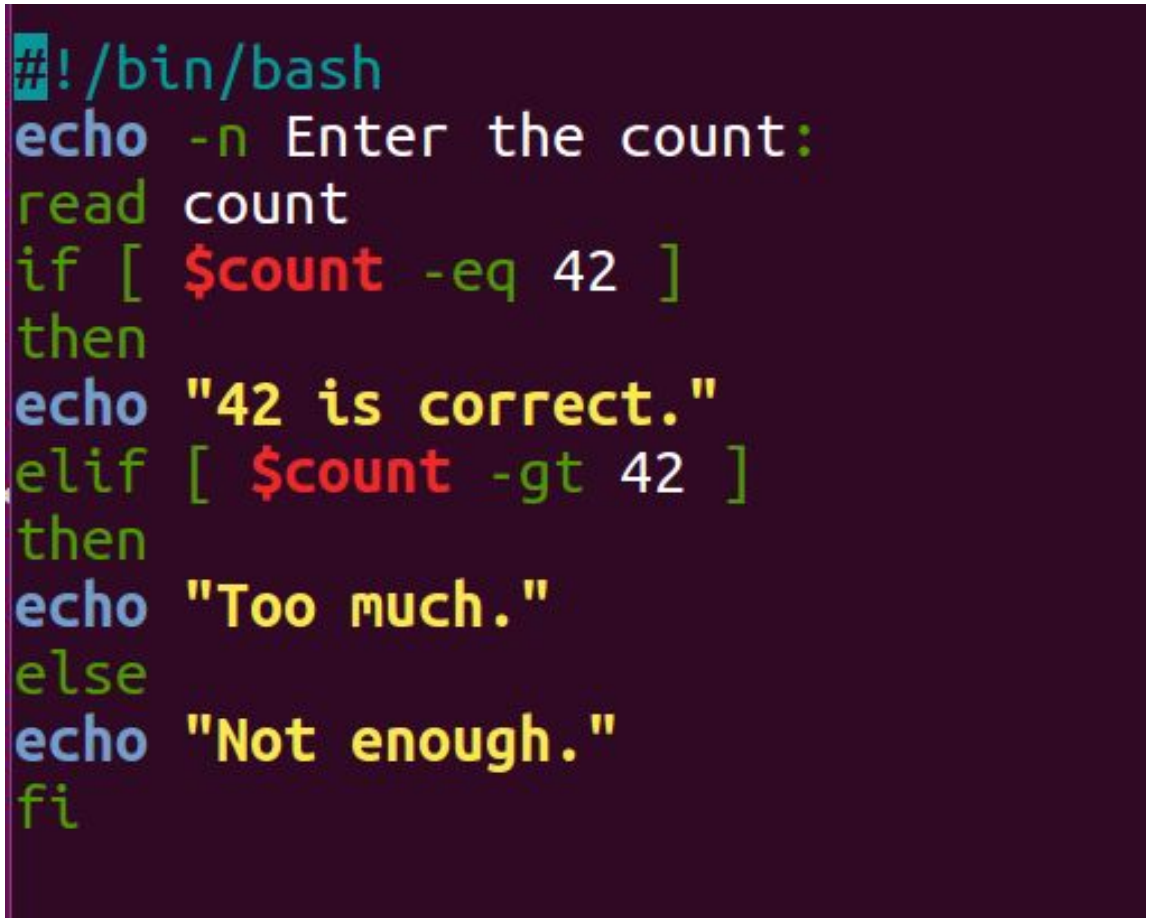
else echo isit.txt not found!
fi



if then elif

- You can nest a new if inside an else with elif.

```
#!/bin/bash
echo -n Enter the count:
read count
if [ $count -eq 42 ]
then
echo "42 is correct."
elif [ $count -gt 42 ]
then
echo "Too much."
else
echo "Not enough."
fi
```



For loop

```
#!/bin/bash
for counter in `seq 1 20`
do
echo counting from 1 to 20, now at $counter
sleep 1
done
```

```
for counter in `seq 1 20`
do
echo counting from 1 to 20, now at $counter
sleep 1
done
```

- another way:

```
#!/bin/bash
for i in 1 2 4
do
echo $i
done
```

The while loop

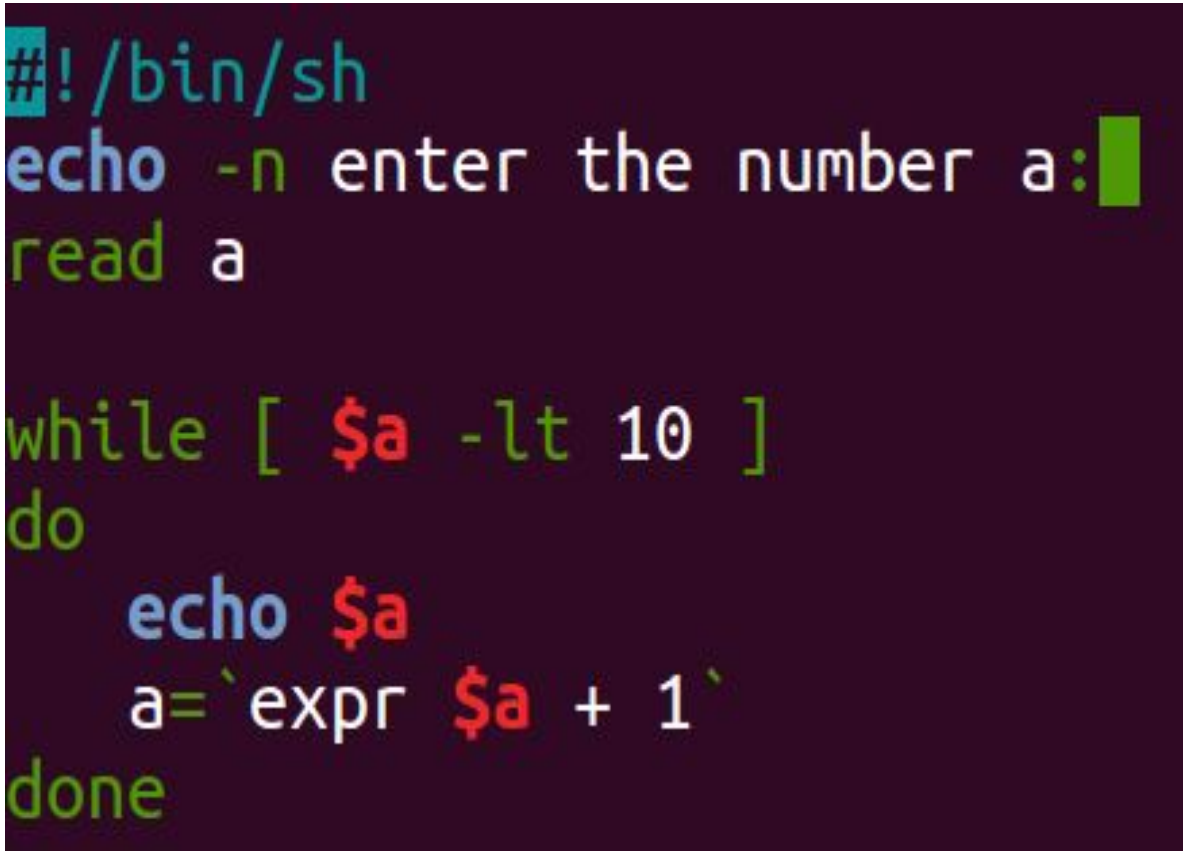
while command

do

Statement(s) to be executed if command is true

done

```
#!/bin/sh
a=0
while [ $a -lt 10 ]
do
   echo $a
   a=`expr $a + 1`
done
```



script parameters

- A bash shell script can have parameters.
- you can pass the arguments while executing

```
#!/bin/bash
echo The first argument is $1
echo The second argument is $2
echo The third argument is $3
echo \$ $$ PID of the script
echo \# $# count arguments
echo \? $? last return code
echo \* $* all the arguments
```

```
(base) sagar@sagar-Lenovo-ideapad-320-15IKB:~/Desktop/HPC$ ./test2.sh hello hi h
ru
The first argument is hello
The second argument is hi
The third argument is hru
$ 21861 PID of the script
# 3 No of the arguments
? 0 last return code
* hello hi hru all the arguments
```

String comparison

Write a shell script that asks for a valid password as input. If given password matches to secret password stored internally, then only it allows the access

```
#!/bin/sh
# This is some secure program that uses security.
VALID_PASSWORD="secret"
echo "Please enter the password:"
read PASSWORD
if [ "$PASSWORD" = "$VALID_PASSWORD" ]
then
echo "You have access!"
else
echo "ACCESS DENIED!"
fi
```

Sending an email from bash automated script

Install mailutils package first

sudo apt-get install mailutils

Prompted for postfix configuration choose options

general type of mail configuration : internet site

System mail name : testmail.com (basically domain name)

After this installation will take place.

Then go to postfix directories main.cf file

sudo nano /etc/postfix/main.cf

This will open up that file for editing , make sure you have write permissions otherwise make it by chmod command.

Find the field relayhost and insert gmail smtp details or according to your mail account.

Relayhost = [smtp.gmail.com]:587

After that in the last portion of file please put authentication as below

#Enable SASL authentication

smtp_sasl_auth_enable = yes

#Disallow methods that allow anonymous authentication

smtp_sasl_security_options = noanonymous

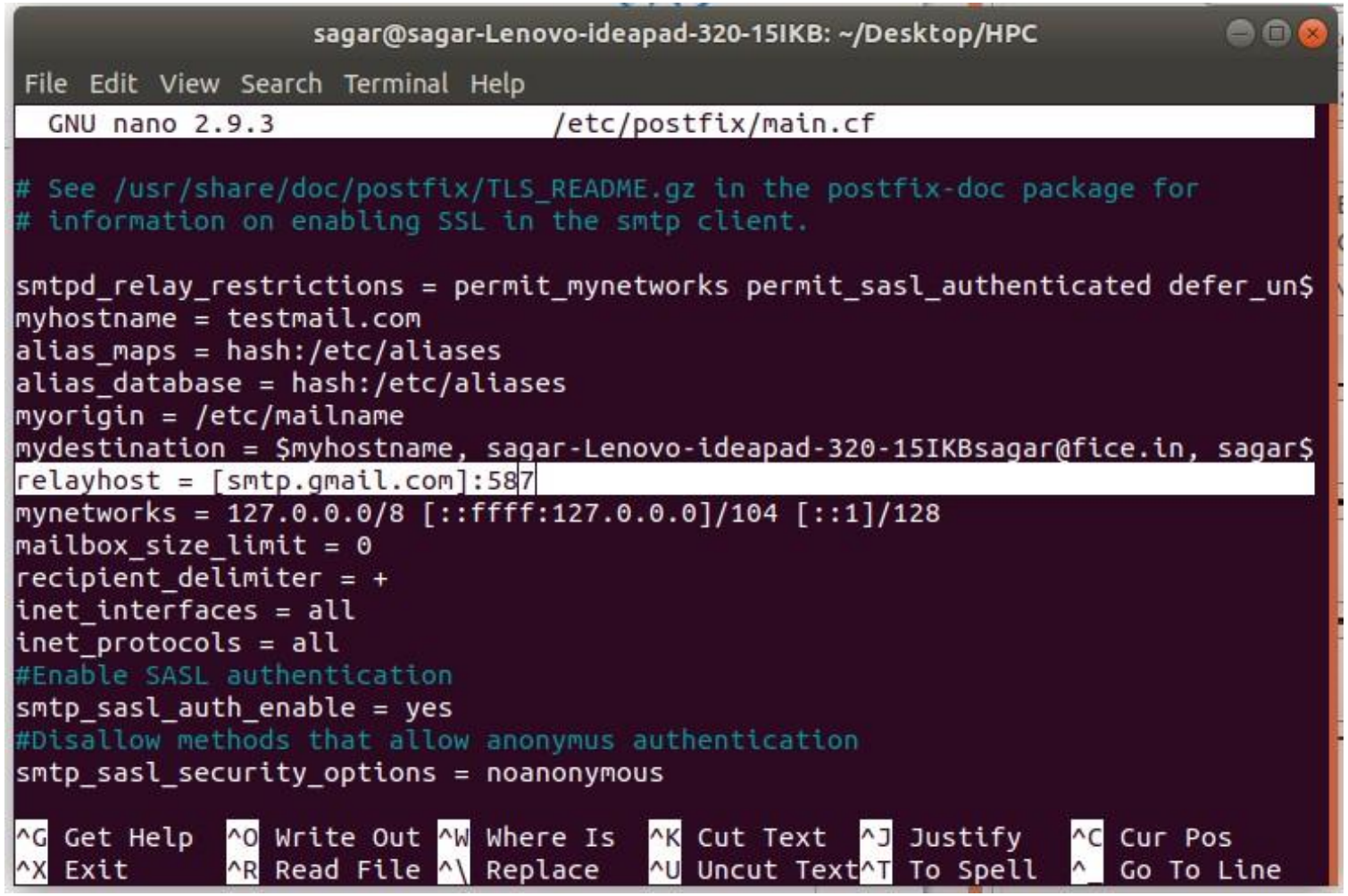
#location of sasl_passwd

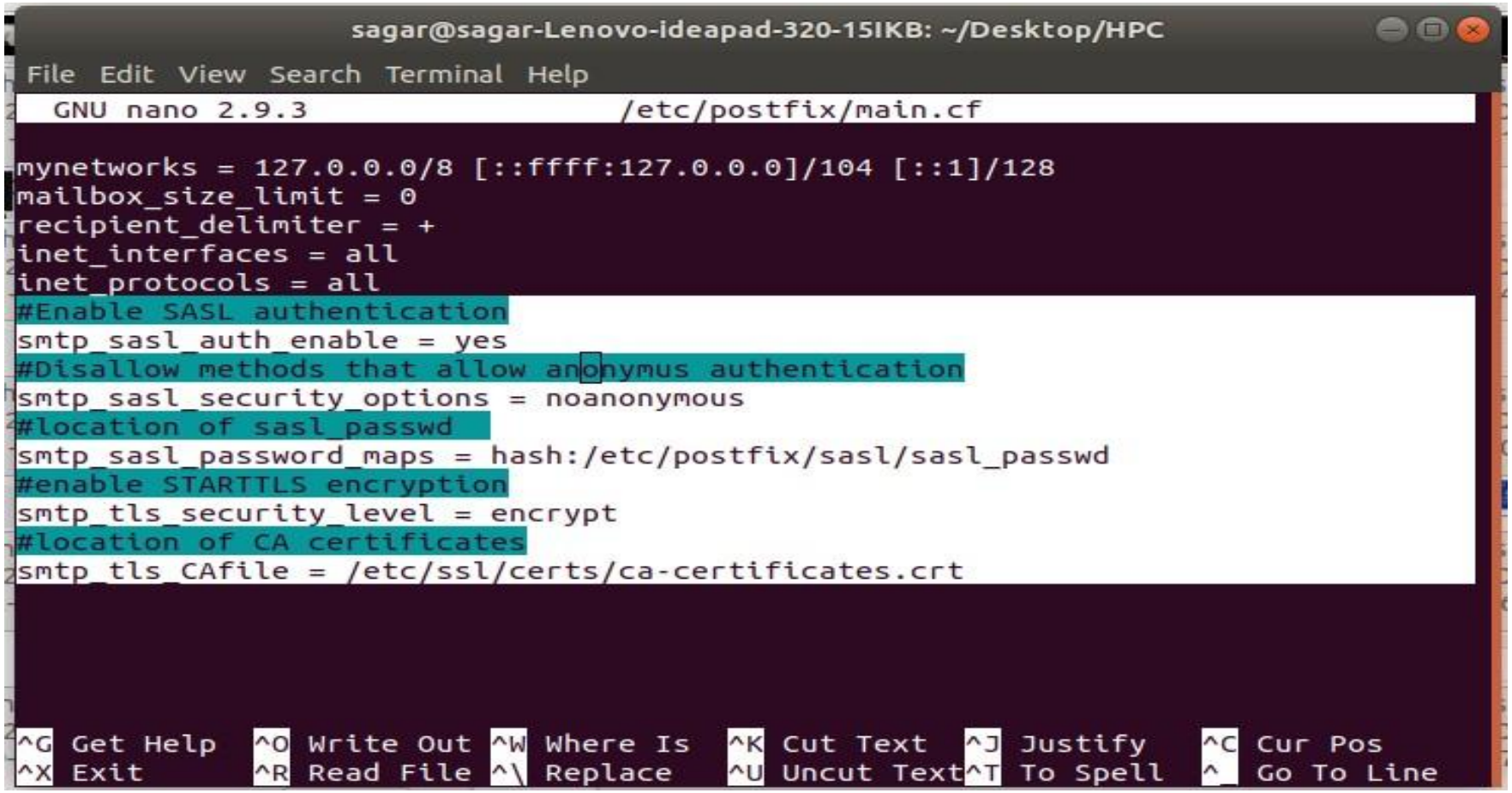
smtp_sasl_password_maps = hash:/etc/postfix/sasl/sasl_passwd

#enable STARTTLS encryption

smtp_tls_security_level = encrypt

#location of CA certificates
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt


sagar@sagar-Lenovo-ideapad-320-15IKB: ~/Desktop/HPC
File Edit View Search Terminal Help
GNU nano 2.9.3 /etc/postfix/main.cf
# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
# information on enabling SSL in the smtp client.
smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_un$
myhostname = testmail.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = $myhostname, sagar-Lenovo-ideapad-320-15IKBsagar@fice.in, sagar$
relayhost = [smtp.gmail.com]:587
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
#Enable SASL authentication
smtp_sasl_auth_enable = yes
#Disallow methods that allow anonymus authentication
smtp_sasl_security_options = noanonymous
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line


sagar@sagar-Lenovo-ideapad-320-15IKB: ~/Desktop/HPC
File Edit View Search Terminal Help
GNU nano 2.9.3 /etc/postfix/main.cf
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
#Enable SASL authentication
smtp_sasl_auth_enable = yes
#Disallow methods that allow anonymus authentication
smtp_sasl_security_options = noanonymous
#location of sasl_passwd
smtp_sasl_password_maps = hash:/etc/postfix/sasl/sasl_passwd
#enable STARTTLS encryption
smtp_tls_security_level = encrypt
#location of CA certificates
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

Then reload the postfix by

sudo postfix reload

Then go to sasl_passwd file

sudo nano /*etc/*postfix/sasl/sasl_passwd

Put following smtp details with login details into the file (gmail id and password)

[smtp.gmail.com]:587 xyx@gmail.com:password

Save and exit that file. And then map the password with following command

sudo postmap *etc/*postfix/sasl/sasl_passwd

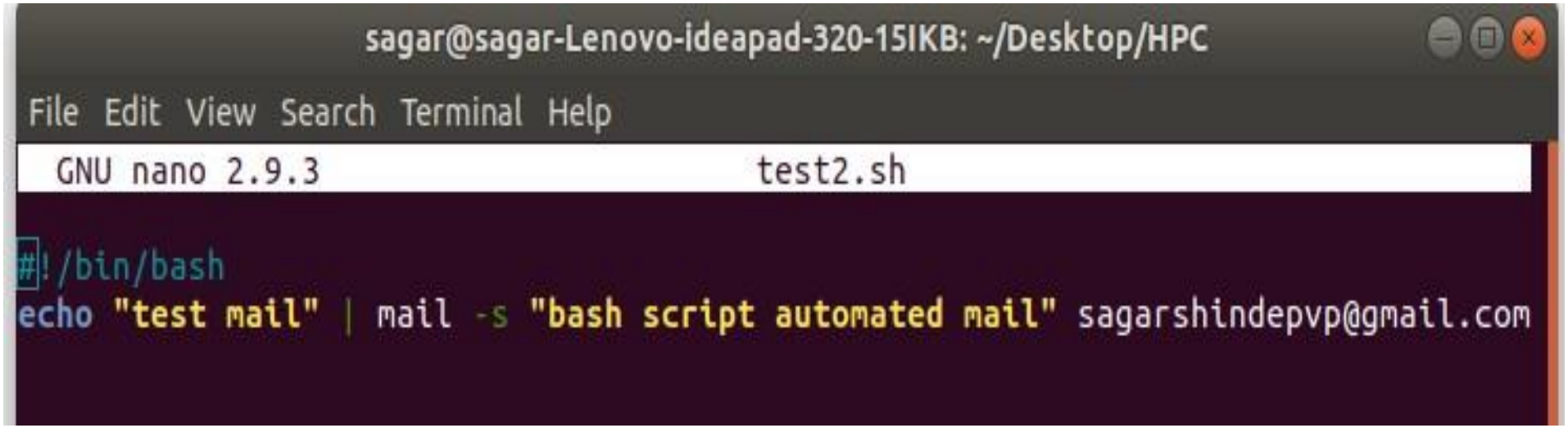
After this go to a shell script file and do the script as follows

#!/bin/bash

echo "test mail" | mail -s "bash script automated mail" **xyz@gmail.com**

Here body of mail is “test mail” , subject is “bash script automated mail” and recipient will be xyz@gmail.com

Run your bash script.



You should receive mail. But in case you don't please check whether your gmail settings are enables for less secure apps because it might block the mail service from linux to gmail. Your two step authentication needs to be disabled for this enabling of less secure apps access.

Once you are done with this you will receive a mail which we executed through our bash script.

References -

1. https://wikidiff.com/command/script#:~:text=is%20that%20command%20is%20(computing,once%20to%20execute%20in%20sequence