LINUX

Table of content

Click on the link to select a title

Process

compression

Shell Script

Perl

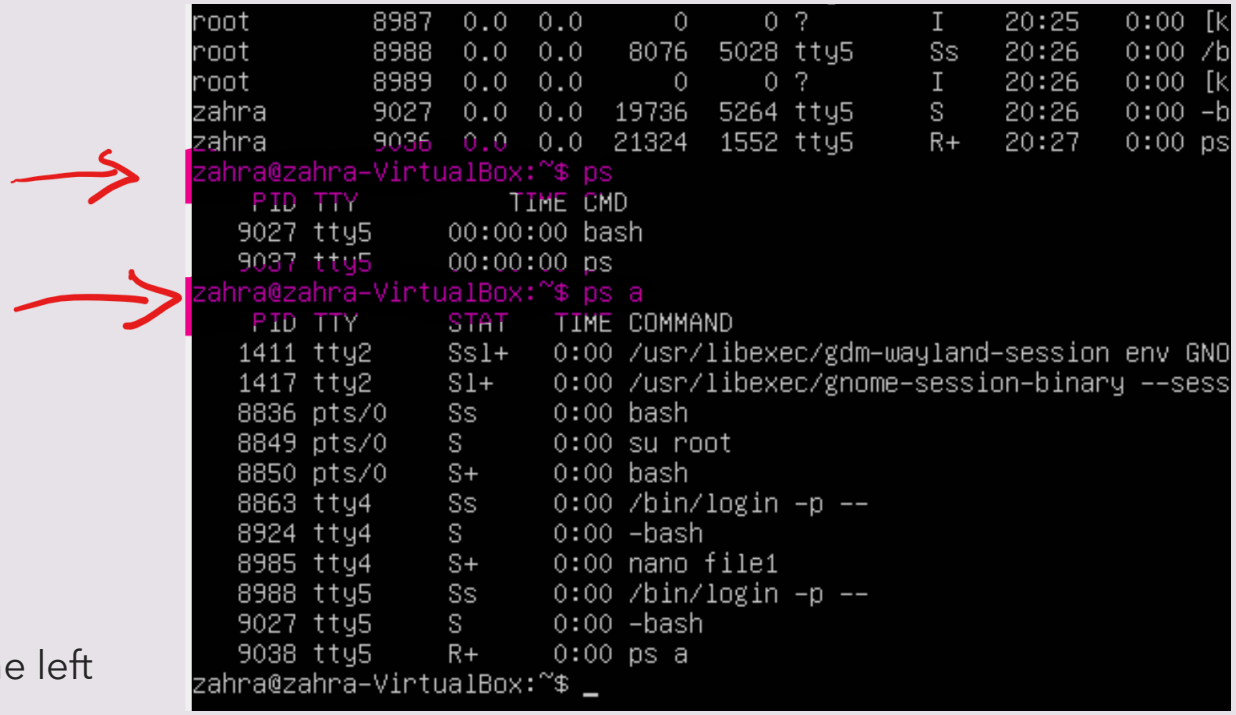
Process

Get process id (PID)

- **Access Root :** su root >> root password

- **Get id process for file1:**
  1. **Switch to tty:** ctrl+ alt + fn + f4
  2. **Nano file1:** keep it opened
  3. **Switch to another tty:** Alt + right arrow
     - ✓ ps For the current process in the same tty
     - ✓ pgrep nano for nano process
     - ✓ ps a For all process
     - ✓ ps aux For more details

  To get process id, ps a > look for nano file1> PID on the left



ps, ps a, ps aux <u>in lowercase</u>

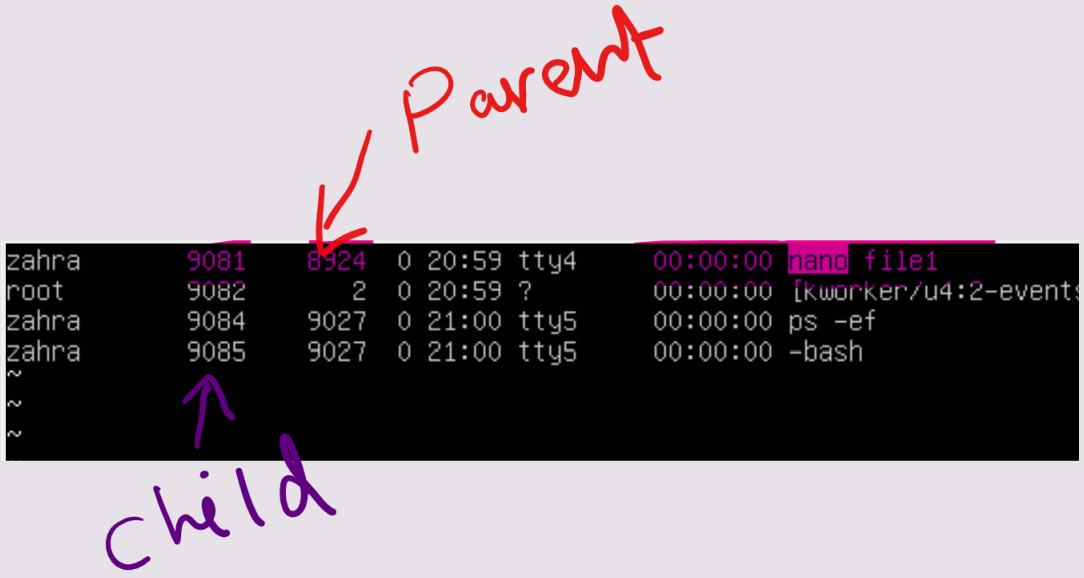
Back to the table of contents

Process

Parent process id (PPID)

1. **Create a process:** nano file1 for example
2. **Several ways to get parent :**
   - ✓ Ps
   - ✓ ps tree|less (space after ps only)
   - ✓ ps -ef|less (space after ps only)
3. **Search for the process:** /name of process

/nano for example



Back to the table of contents

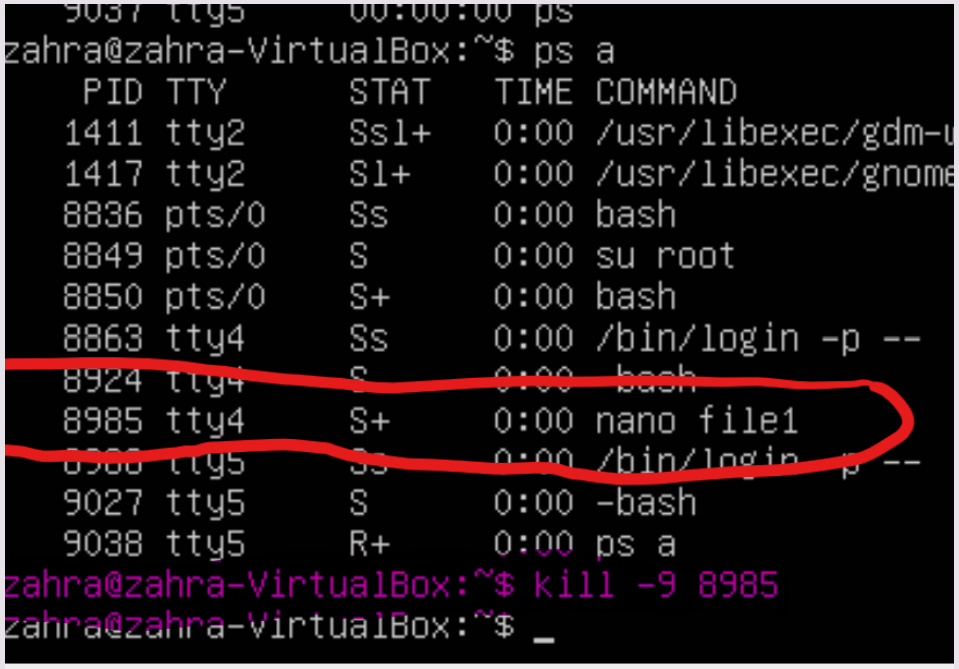
Process

Kill process

- **First get the process id:** previous slide.
- **Kill in soft manner:** kill processid

  Example: kill 8985
- **Kill in hard manner:** kill -9 processid

  Example: kill -9 8985



Back to the table of contents

Shell Script

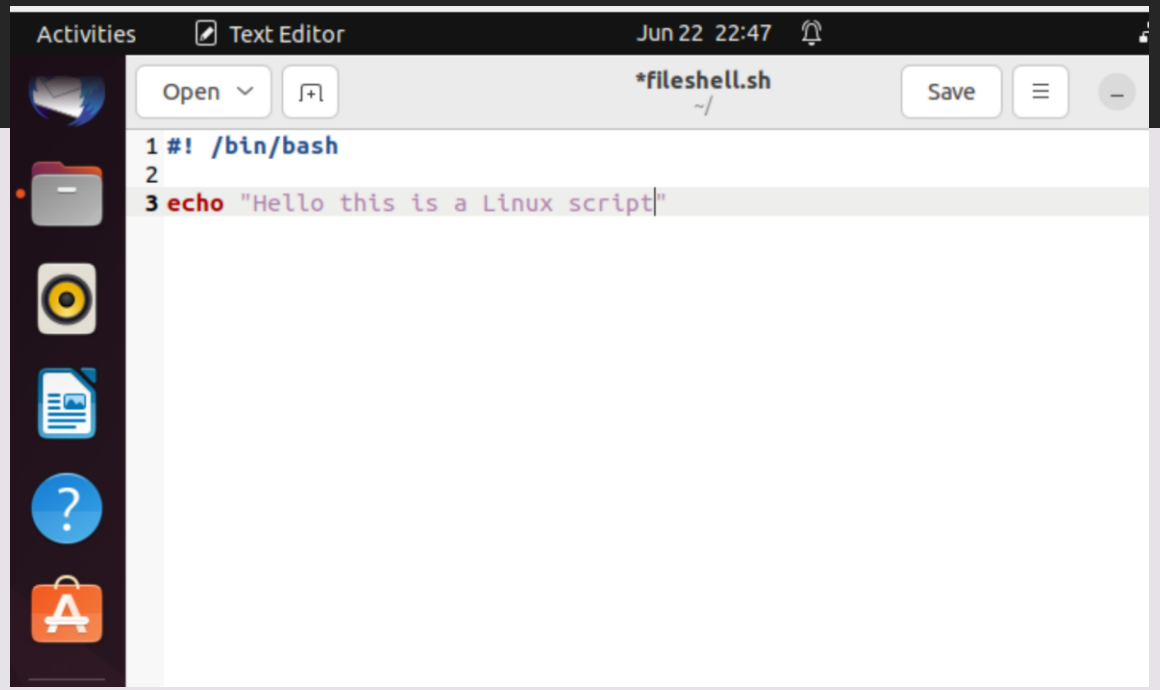
Create shell file

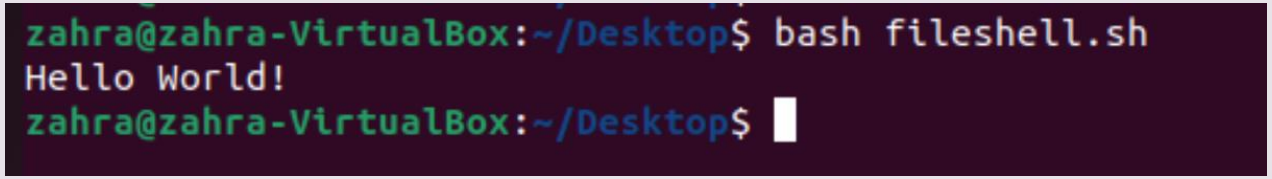
- **In Terminal:** touch fileshell**.sh**
- **Edit your file:** in file location
- **Syntax:**

#! /bin/bash

echo “Hello world!”

- **Save**
- **To run the file open terminal**
- **bash fileshell.sh or ./ fileshell.sh**
- **If used ./ fileshell.sh make sure to change file peremmissions.**



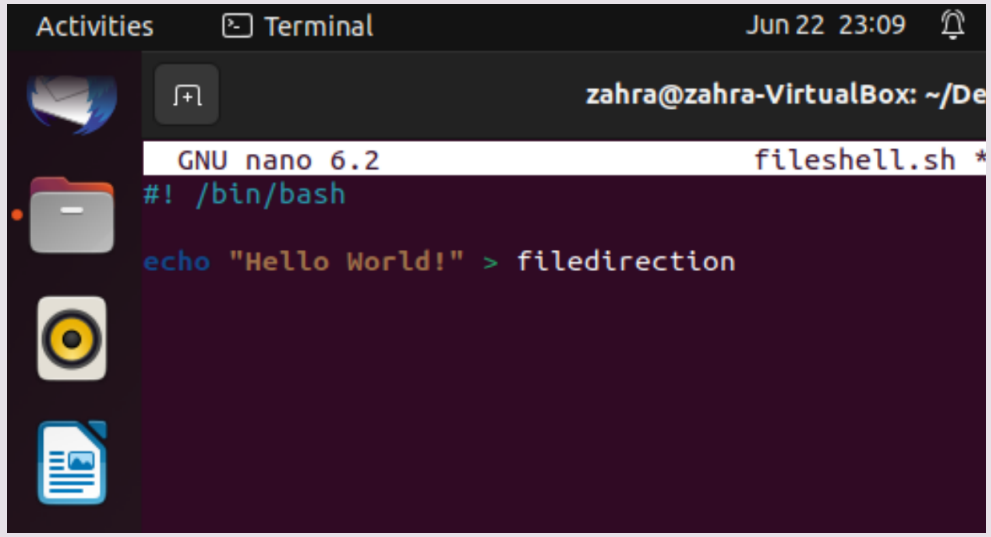


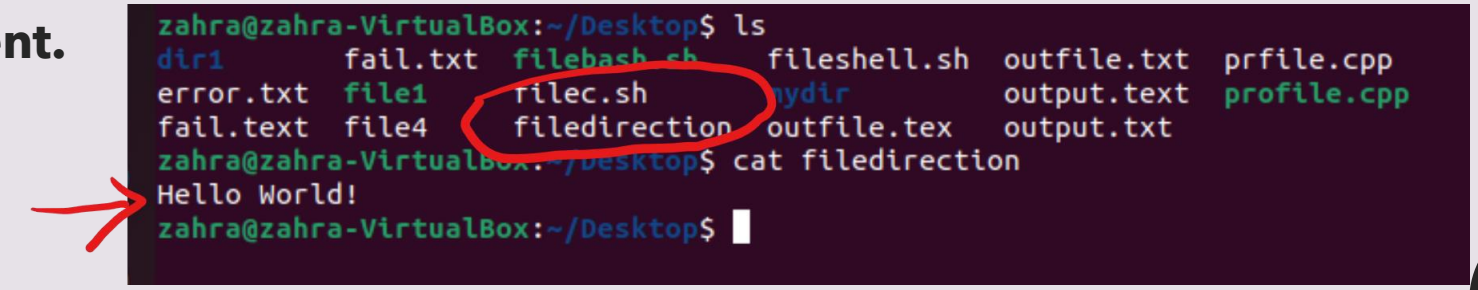
Back to the table of contents

Shell Script

Direct & append data to another file

- **Open fileshell.sh:** from nano or desktop
- **In echo line: echo** “Hello World” **>** filedirection
- This will direct data from fileshell.sh to new file name filedirection.
- **Cat filedirection to view content.**
- **To append use >>**



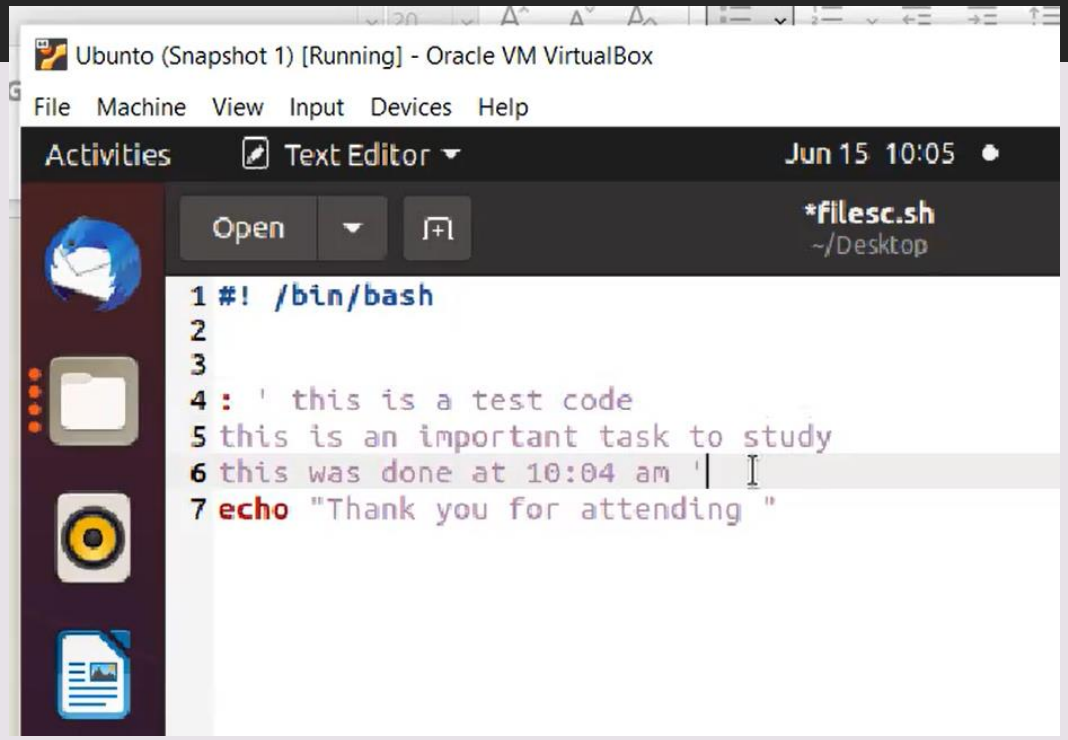


Shell Script

Comment in file

- **To comment in shell script file type your comment between single quote : ' '**

:' this is a comment '



Back to the table of contents

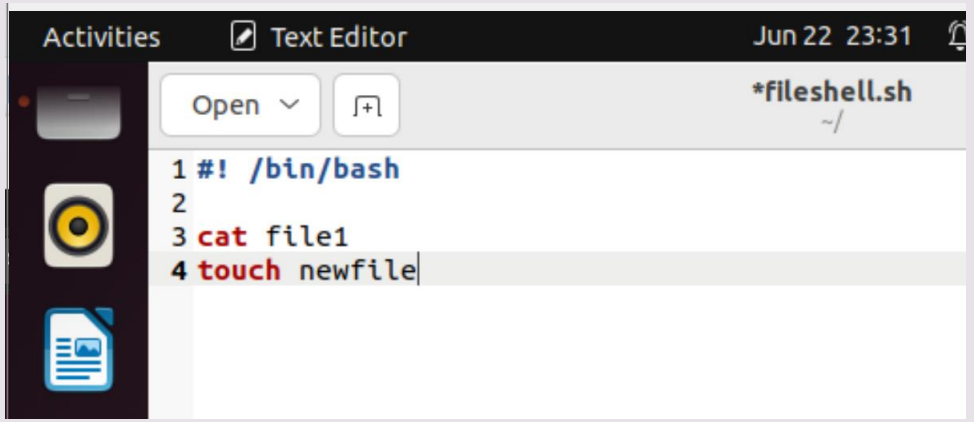
Shell Script

Touch, cat files from shell script file

- **Inside fileshell.sh :** cat file1

Touch newfile

- **Bash fileshell.sh**
- **You'll see the content of file1 and newfile was created.**



Back to the table of contents

Shell Script

If statement (if , then)

- **Syntax:**

count=10

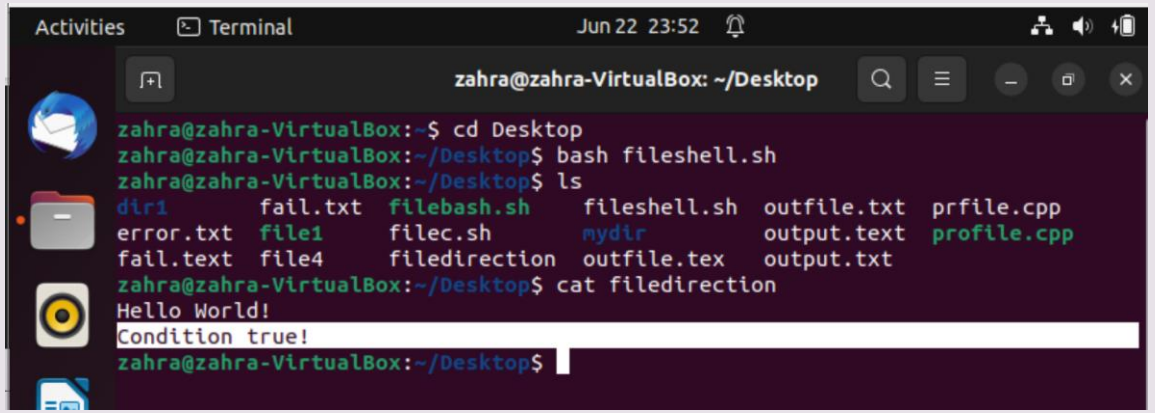
if [$count –eq 10]

Then

echo “true” >>filedirection

fi

Conditions to use
Greater than ➔ -gt
Equal to ➔ -eq
Less than ➔ -lt
Not equal ➔ -ne



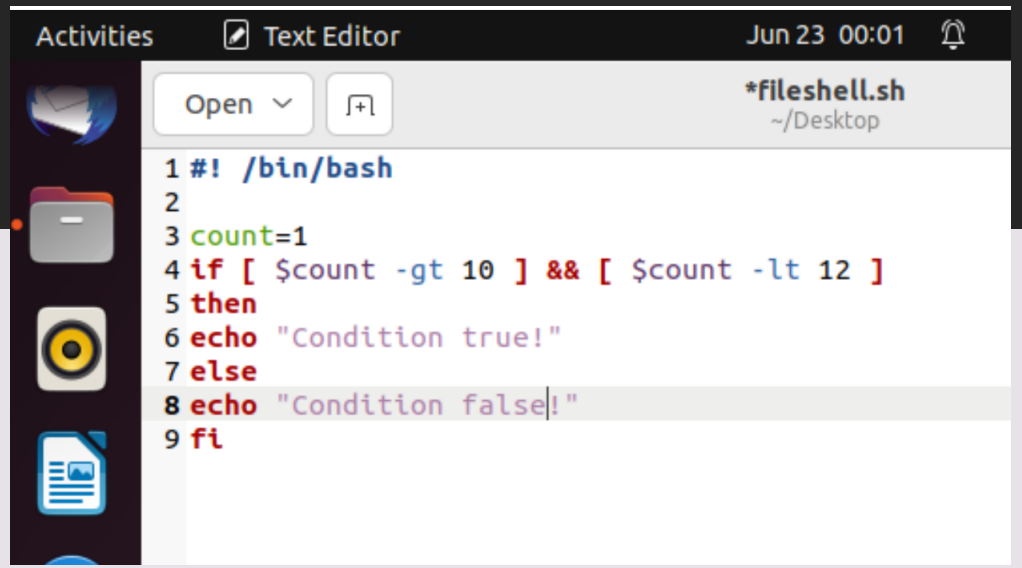
- **In this example checking the value of the variable if the condition is true it will append filedirection.**

Back to the table of contents

Shell Script

If statement (if , else)

Conditions to use
Greater than ➔ -gt
Equal to ➔ -eq
Less than ➔ -lt
Not equal ➔ -ne



- **Syntax:**

```
count=1
if [ $count  -eq 10 ] && [ $count  -lt 12 ]
Then
echo “condition true”
Else
echo “condition true”
 fi
```

```
zahra@zahra-VirtualBox:~/Desktop$ bash fileshell.sh
Condition false!
zahra@zahra-VirtualBox:~/Desktop$
```

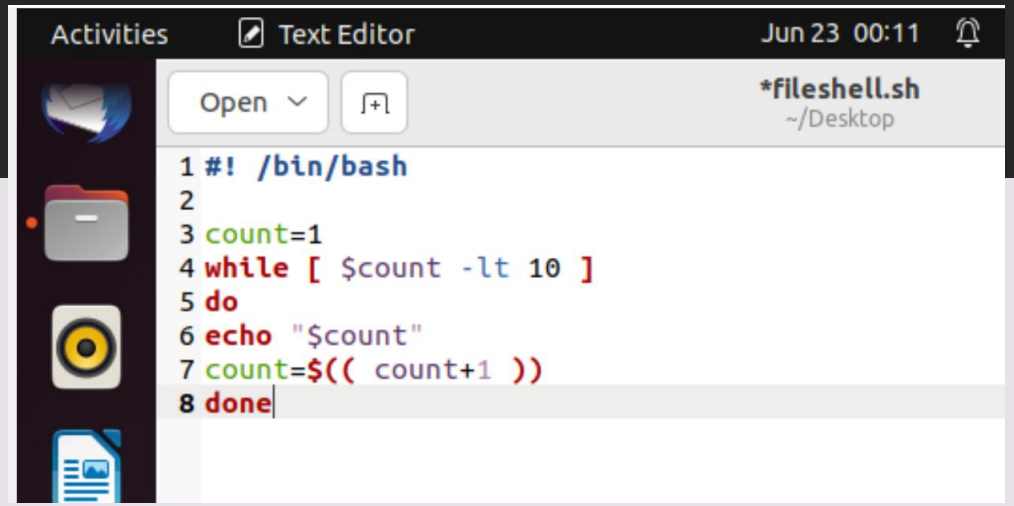
Back to the table of contents

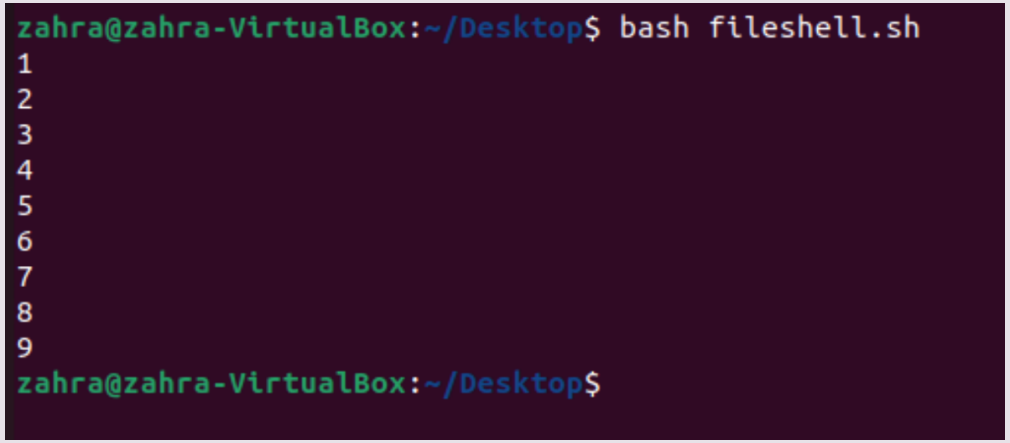
Shell Script

Loop (while, do)

- **Syntax:**

```
count=1
while [ $count –lt 10]
do
echo “$count”
count=$(( count+1 ))
done
```

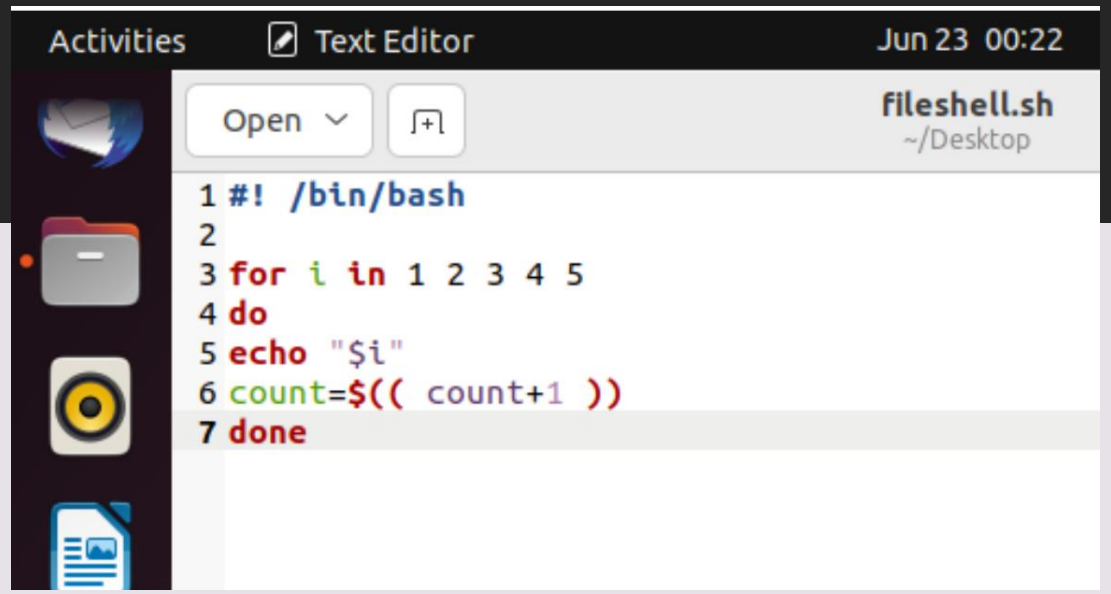




Back to the table of contents

Shell Script

Loop (for, do)



- **Syntax:**

for i in 1 2 3 4 5

do

echo “$count”

count=$((count+1))

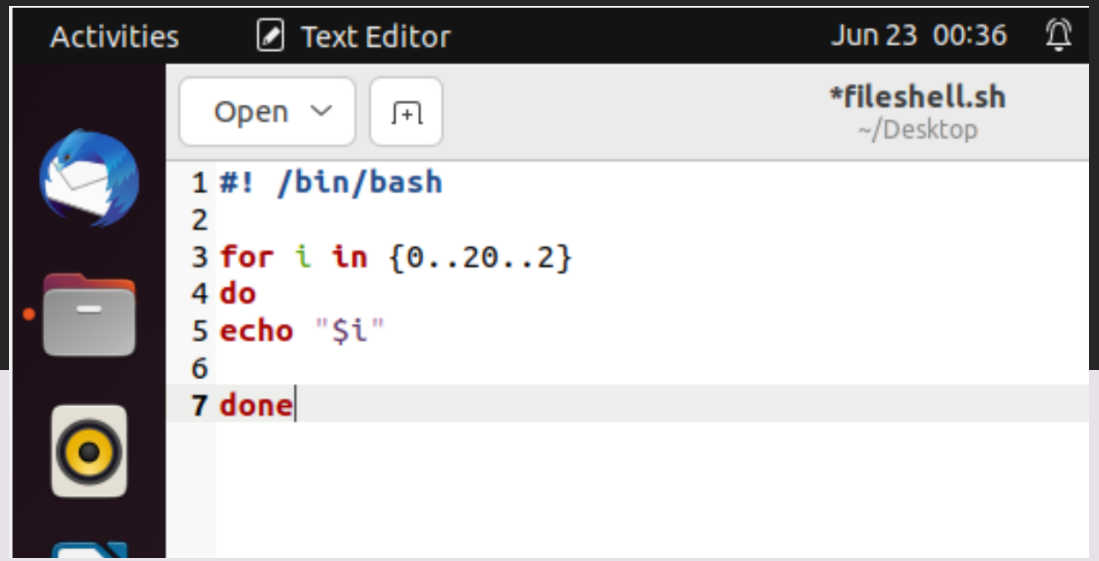
done

```
zahra@zahra-VirtualBox:~/Desktop$ bash fileshell.sh
1
2
3
4
5
zahra@zahra-VirtualBox:~/Desktop$
```

Back to the table of contents

Shell Script

Loop (for, do with steps)



- **Syntax:**

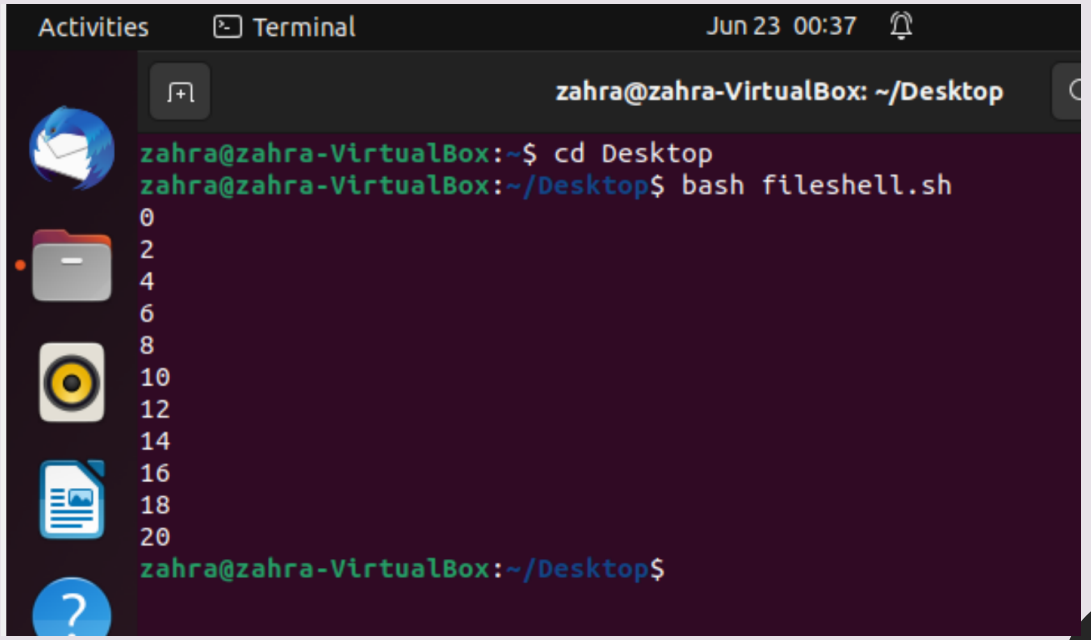
for i in { 0..20..2 }

do

eco “$i”

Done

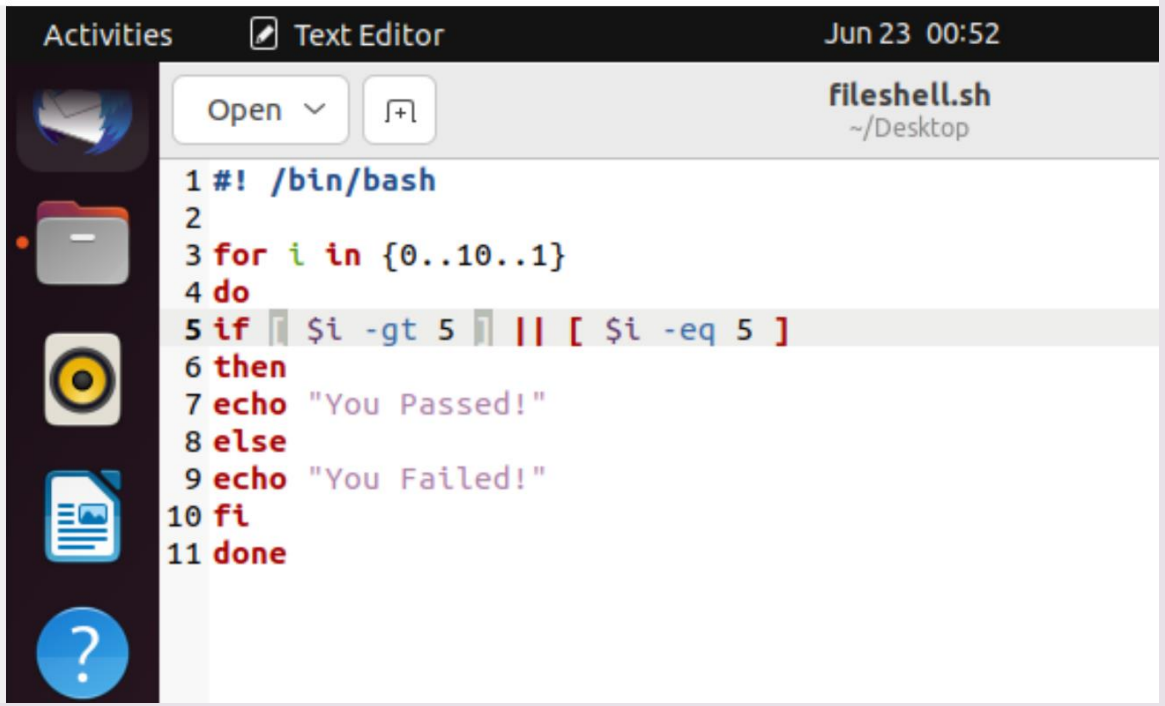
This will start counting from 0 to 20 jumping 2 steps at a time.

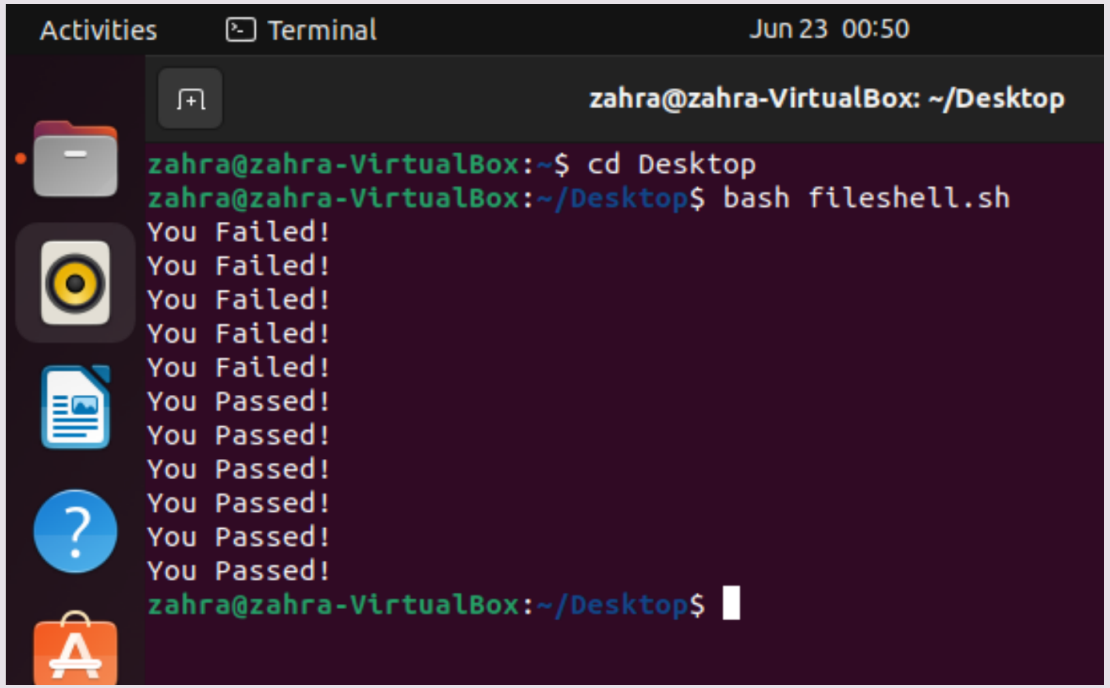


Back to the table of contents

Shell Script

Combined loop & if





Back to the table of contents

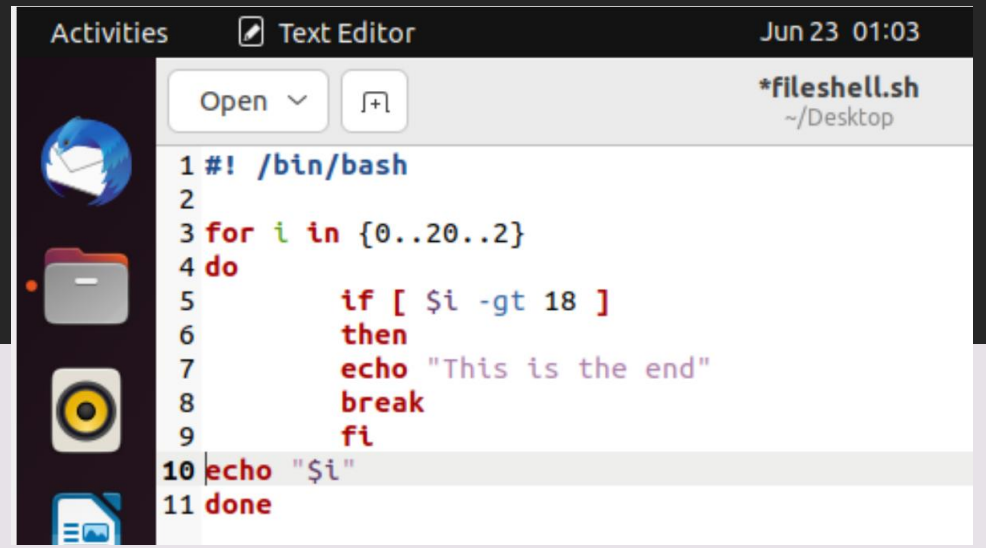
Shell Script

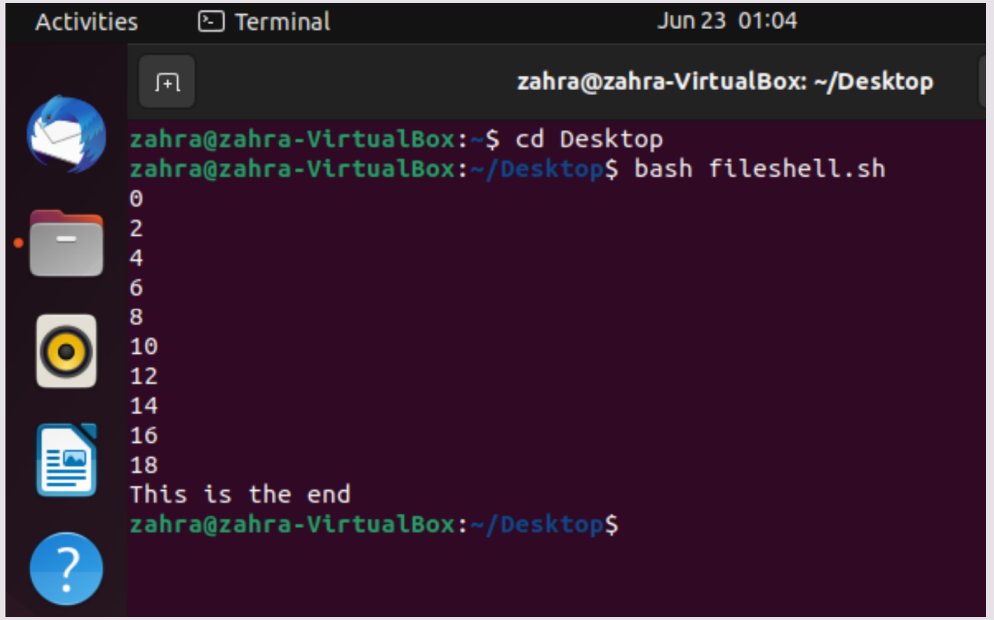
Break a loop

- **Syntax:**

```
for i in { 0..20..2 }
do
        if [ $i –gt 18 ]
        then
        break
        Fi
echo “$i”
done
```

This will start counting from 0 and break the loop and stop at 18.





Back to the table of contents

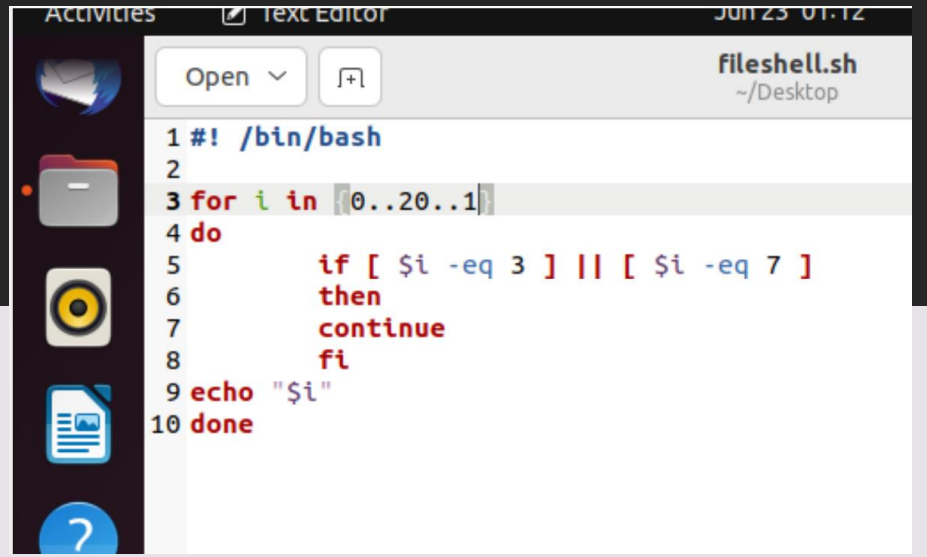
Shell Script

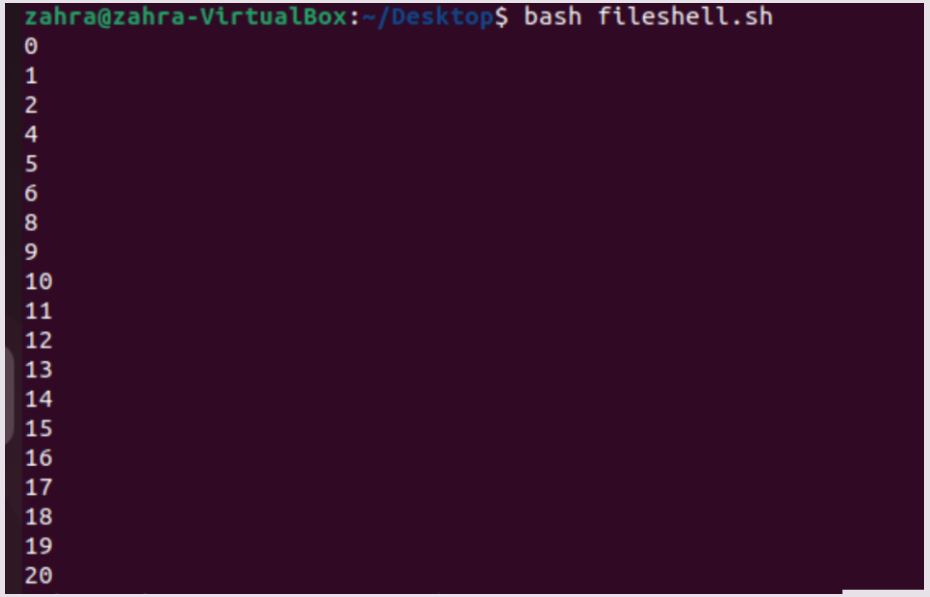
Continue a loop

- **Syntax:**

```
for i in { 0..20..1 }
do
        if [ $i –eq 3 ] || [ $i –eq 7 ]
        then
        continue
        fi
echo “$i”
done
```

This will start counting from 0 until 20 but skip 3 and 7.





Back to the table of contents

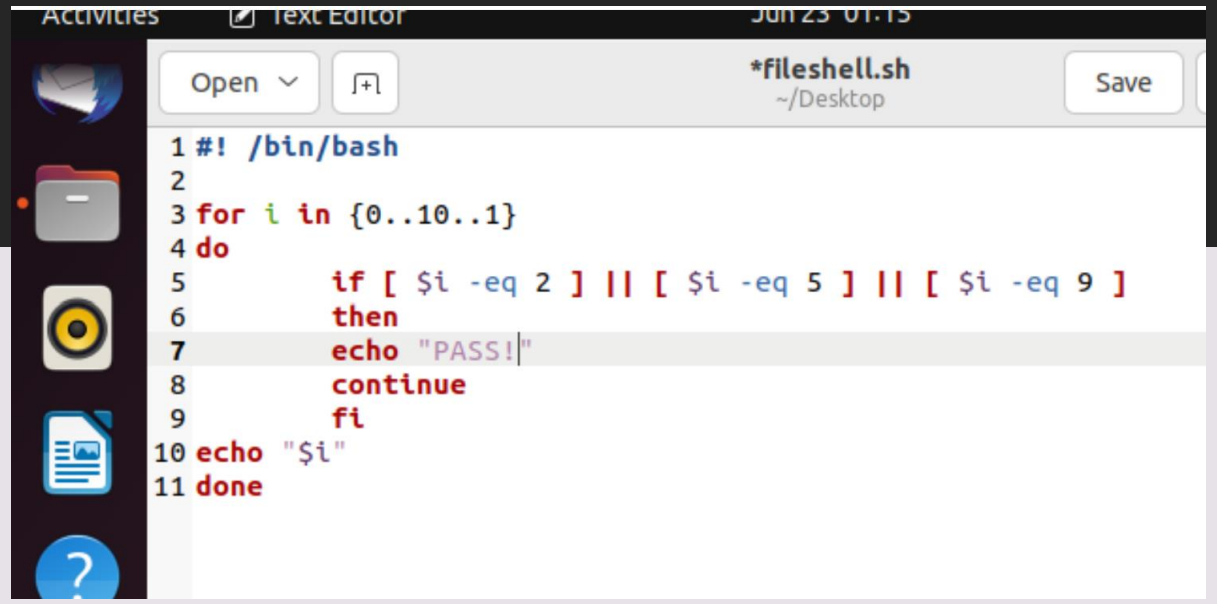
Shell Script

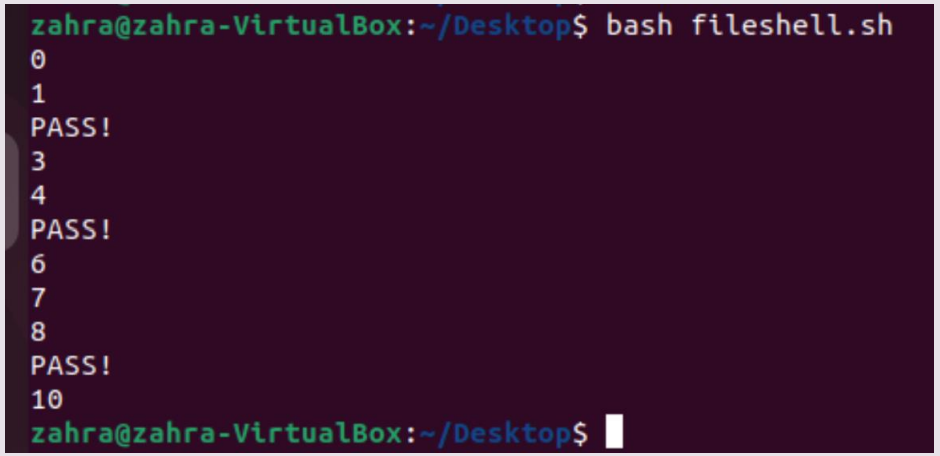
Continue a loop

- **Syntax:**

```
for i in { 0..10..1 }
do
    if [ $i –eq 2 ] || [ $i –eq 5 ] || [ $i –eq 9 ]
    then
    echo “PASS!”
    continue
fi
done
```

This will start counting from 0 until 10 but print PASS on 2, 5 and 9.





Back to the table of contents

Perl

Create & run

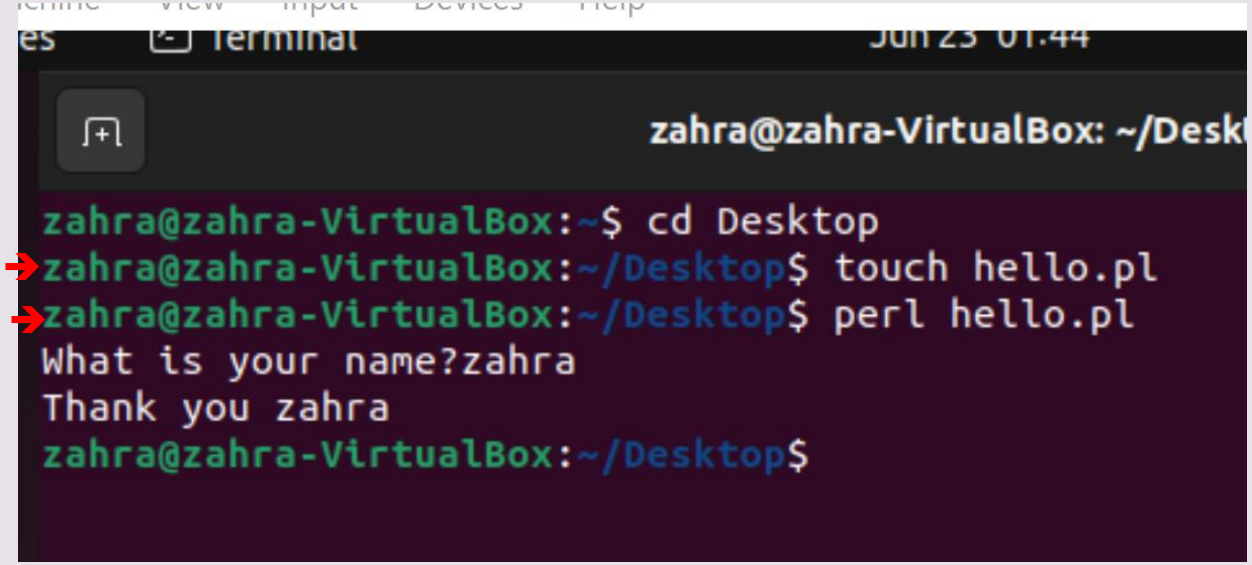
- **Syntax:**

#!/usr/bin/perl

```
print( "What is your name?" );
$name=<STDIN>;
print("Thank you $name");
```

Save file hello.pl

Create file →
Run the file →



Back to the table of contents

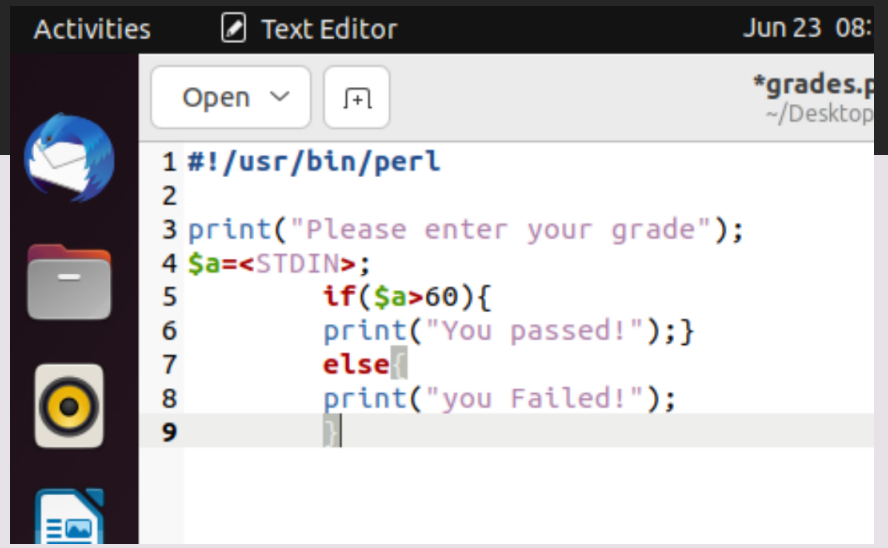
Perl

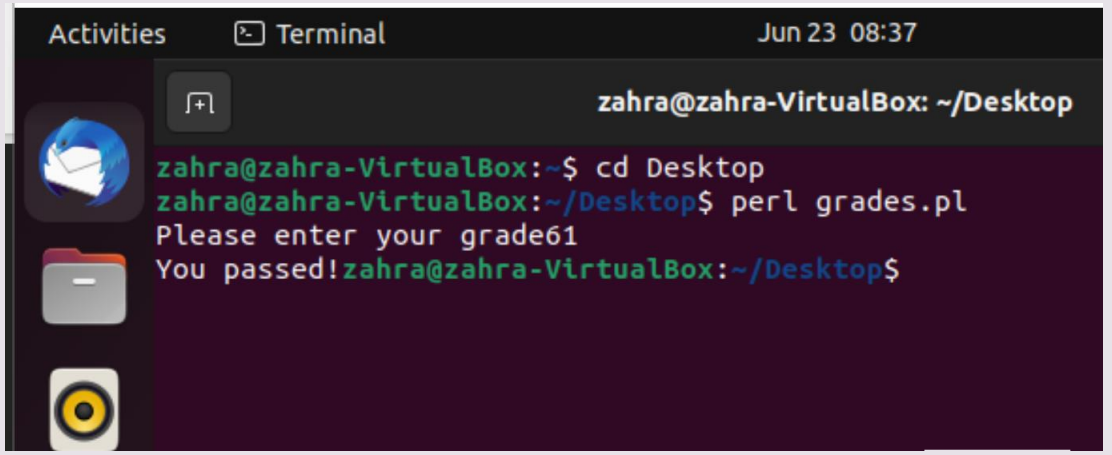
If statement

- **Syntax:**

```
#!/usr/bin/perl
print( “Enter your grade" );
$a=<STDIN>;
        if($a>60){
        print(“You Passed!");}
        else{
        print(“You Failed!");}
```

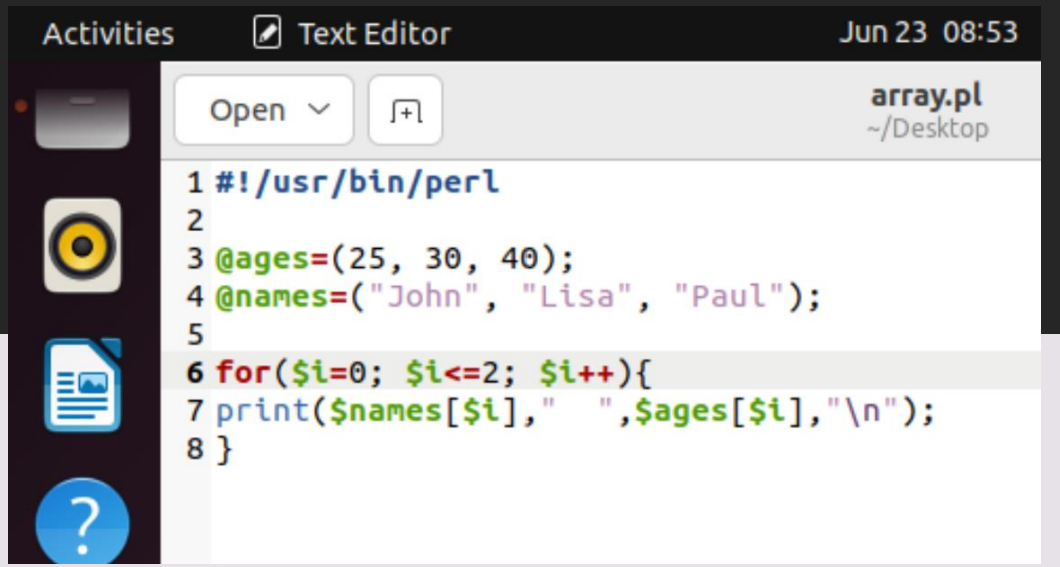
Save file grades.pl





Back to the table of contents

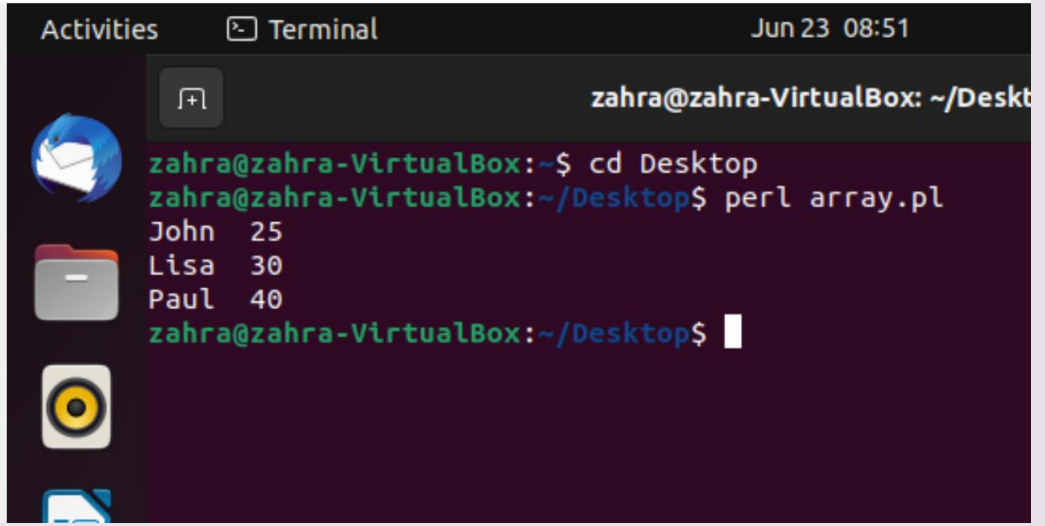
Perl
Arrays



- **Syntax:**

```
#!/usr/bin/perl
@ages = (25, 30, 40);
@names = ("John ", "Lisa", "Paul");
        for ($i=0; $i<=2; $i++){
        print($names[$i],"   " , $ages[$i], "\n");
        }
```

Save file array.pl

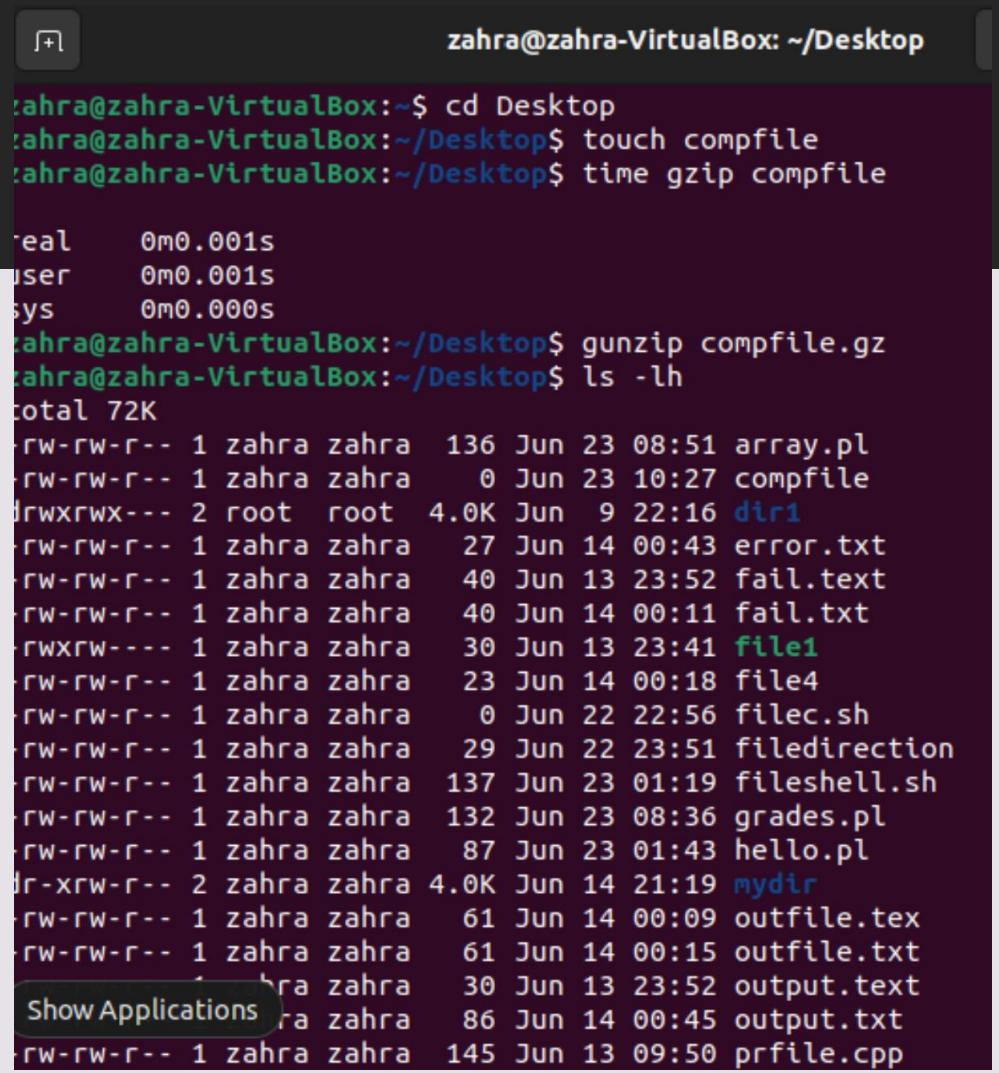


Back to the table of contents

File compression

gzip technique

- **Gzip technique is faster (0.002 sec)**
- **To compress:** gzip fileName
- **To expand (extract):** g**un**zip fileName**.gz**
- **Check compressed file size :** ls -lh fileName**.gz**
- **Check regular file size :** ls -lh fileName
- **To calculate execution time:** time gzip fileName

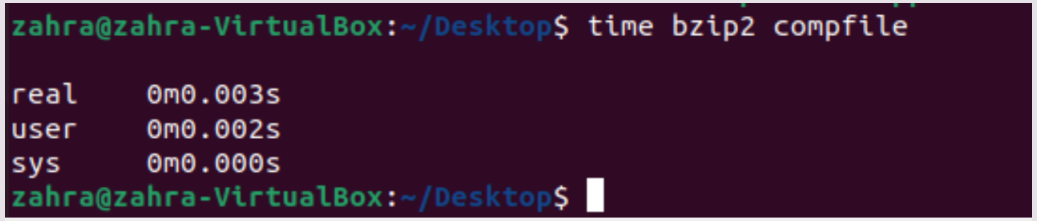


Back to the table of contents

File compression

bzip2 technique

- **bzip2 technique saves 15% more space (0.003 sec)**
- **To compress:** bzip**2** fileName
- **To expand (extract):** b**un**zip**2** fileName**.bz2**
- **Check compressed file size :** ls -lh fileName**.bz2**
- **Check regular file size :** ls -lh fileName
- **To calculate execution time:** time bzip2 fileName



Back to the table of contents