



LINUX



Table of content

Click on the link to select a title

Process

- [Get process id \(PID\)](#)
- [Parent process id \(PPID\)](#)
- [Kill process](#)

Shell Script

- [Create shell file](#)
- [Direct & append](#)
- [Comment in file](#)
- [Touch, cat files from shell](#)
- [If statement \(if, then\)](#)
- [If statement \(if, else\)](#)
- [Loop \(while, do\)](#)
- [Loop \(for, do\)](#)
- [Loop \(for, do with steps\)](#)
- [Combined loop & if](#)
- [Break a loop](#)
- [Continue a loop](#)
- [Continue a loop2](#)

Perl

- [Create & run](#)
- [If statement](#)
- [Arrays](#)

compression

- [gzip technique](#)
- [bzip2 technique](#)

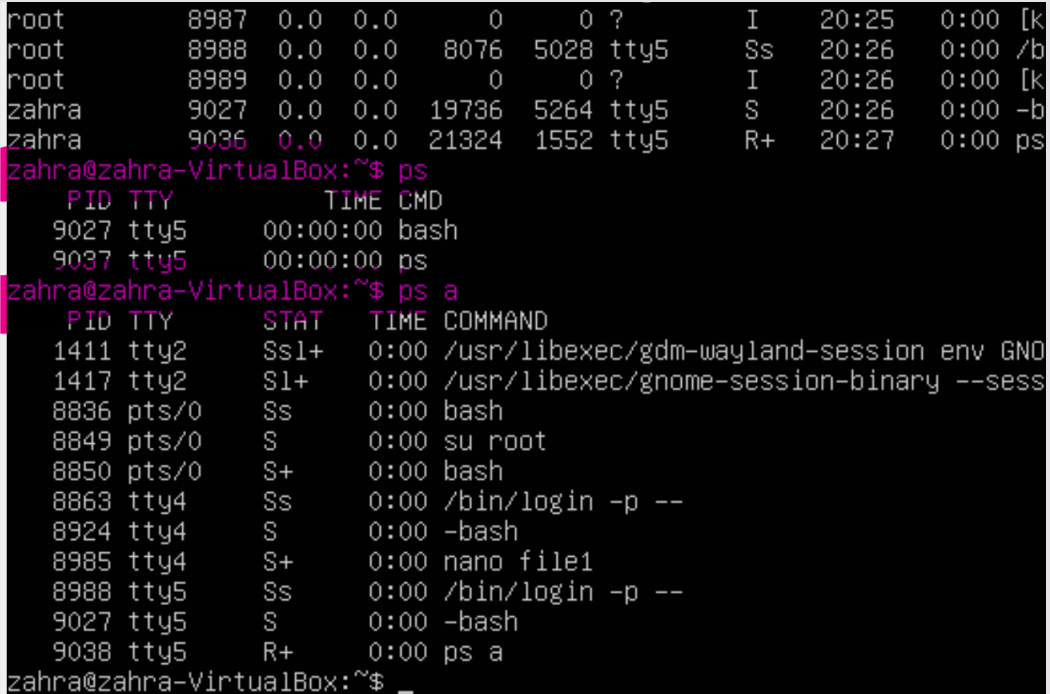
Process

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 in lowercase



```
root      8987  0.0  0.0   0   0 ?        I   20:25   0:00 [k
root      8988  0.0  0.0  8076 5028 tty5    Ss   20:26   0:00 /b
root      8989  0.0  0.0   0   0 ?        I   20:26   0:00 [k
zahra     9027  0.0  0.0 19736 5264 tty5    S    20:26   0:00 -b
zahra     9036  0.0  0.0 21324 1552 tty5    R+   20:27   0:00 ps
zahra@zahra-VirtualBox:~$ ps
  PID TTY          TIME CMD
 9027 tty5        00:00:00 bash
 9037 tty5        00:00:00 ps
zahra@zahra-VirtualBox:~$ ps a
  PID TTY          STAT TIME COMMAND
 1411 tty2        Ssl+   0:00 /usr/libexec/gdm-wayland-session env GNO
 1417 tty2        Sl+    0:00 /usr/libexec/gnome-session-binary --sess
 8836 pts/0        Ss     0:00 bash
 8849 pts/0        S      0:00 su root
 8850 pts/0        S+     0:00 bash
 8863 tty4        Ss     0:00 /bin/login -p --
 8924 tty4        S      0:00 -bash
 8985 tty4        S+     0:00 nano file1
 8988 tty5        Ss     0:00 /bin/login -p --
 9027 tty5        S      0:00 -bash
 9038 tty5        R+     0:00 ps a
zahra@zahra-VirtualBox:~$ _
```

[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

```
zahra 9081 8924 0 20:59 tty4 00:00:00 nano file1
root 9082 2 0 20:59 ? 00:00:00 [kworker/u4:2-event
zahra 9084 9027 0 21:00 tty5 00:00:00 ps -ef
zahra 9085 9027 0 21:00 tty5 00:00:00 -bash
~
~
~
```



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

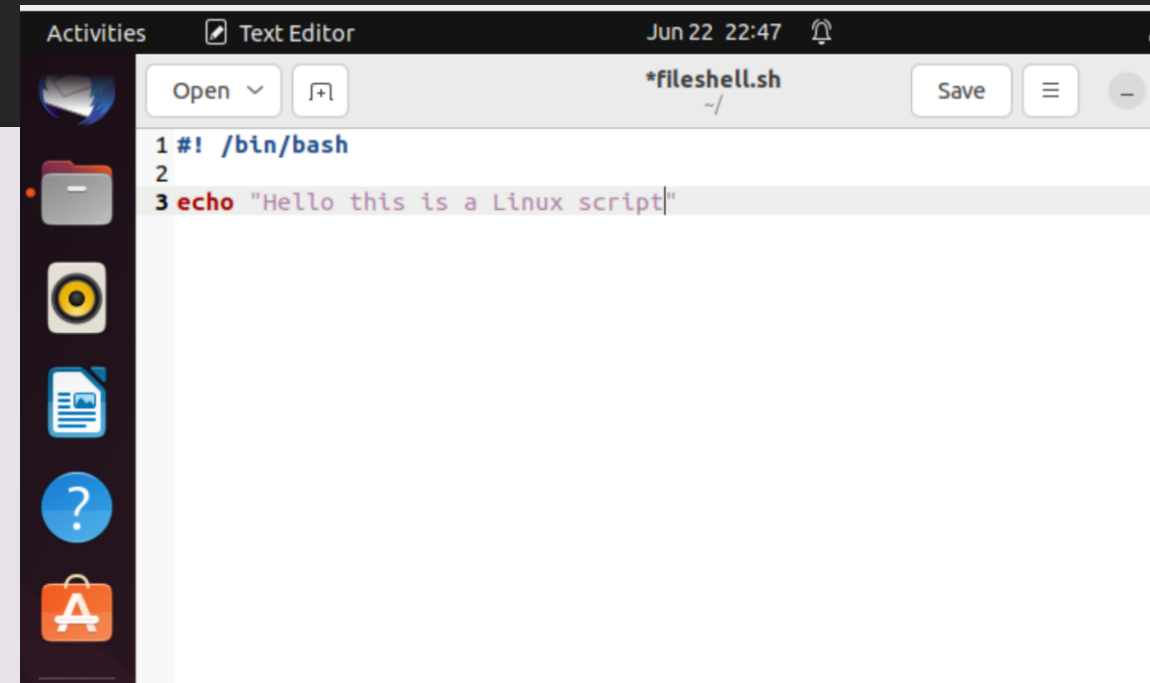
```
9037 ttg5 00:00:00 ps
zahra@zahra-VirtualBox:~$ ps a
  PID TTY          STAT       TIME COMMAND
 1411 tty2      Ssl+        0:00 /usr/libexec/gdm-u
 1417 tty2      Sl+         0:00 /usr/libexec/gnome
 8836 pts/0      Ss          0:00 bash
 8849 pts/0      S           0:00 su root
 8850 pts/0      S+          0:00 bash
 8863 tty4       Ss          0:00 /bin/login -p --
 8924 ttg4      S           0:00 bash
 8985 tty4       S+          0:00 nano file1
 8988 ttg5      Ss          0:00 /bin/login -p --
 9027 tty5      S           0:00 -bash
 9038 tty5      R+          0:00 ps a
zahra@zahra-VirtualBox:~$ kill -9 8985
zahra@zahra-VirtualBox:~$ _
```



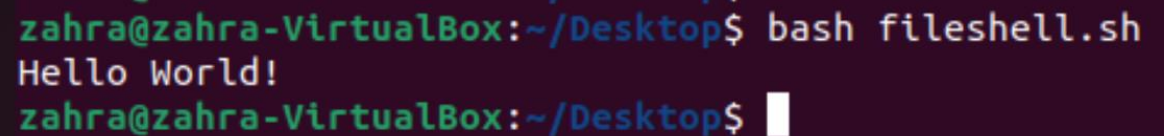
Shell Script

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 permissions.



```
1 #! /bin/bash
2
3 echo "Hello this is a Linux script"
```



```
zahra@zahra-VirtualBox:~/Desktop$ bash fileshell.sh
Hello World!
zahra@zahra-VirtualBox:~/Desktop$
```

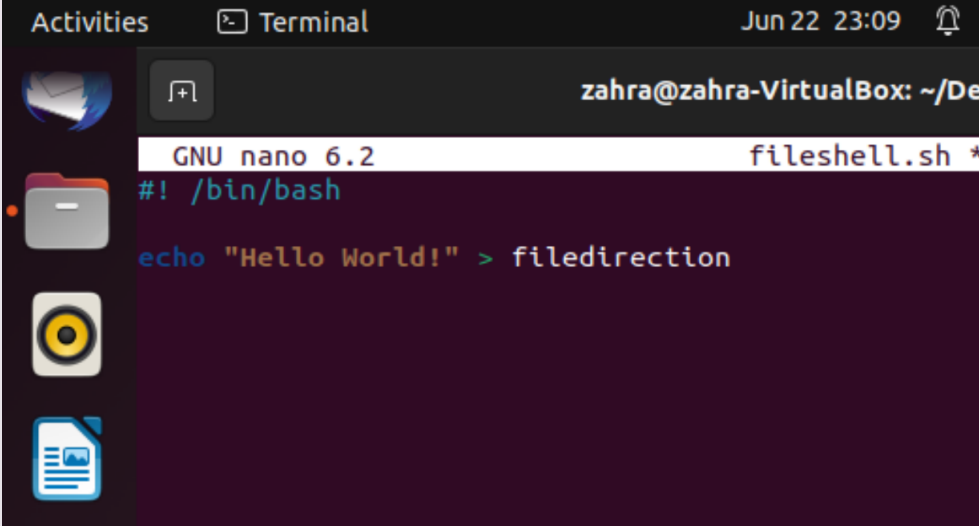
[Back to the table of contents](#)



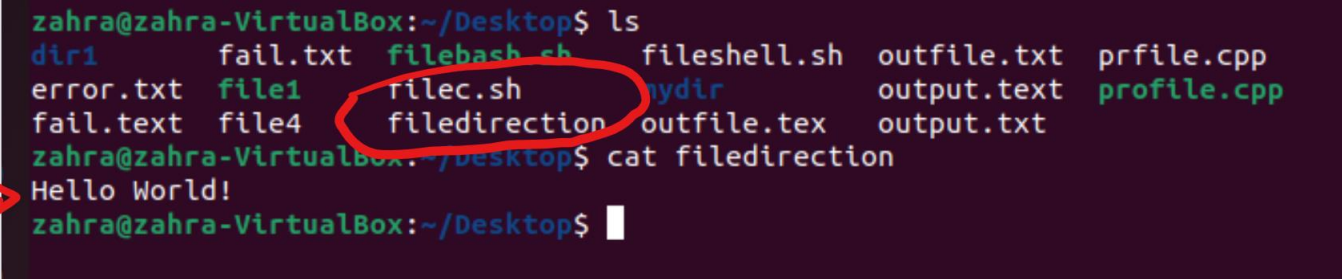
Shell Script

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 >>**



```
Activities Terminal Jun 22 23:09
zahra@zahra-VirtualBox: ~/De
GNU nano 6.2 fileshell.sh *
#!/bin/bash
echo "Hello World!" > filedirection
```



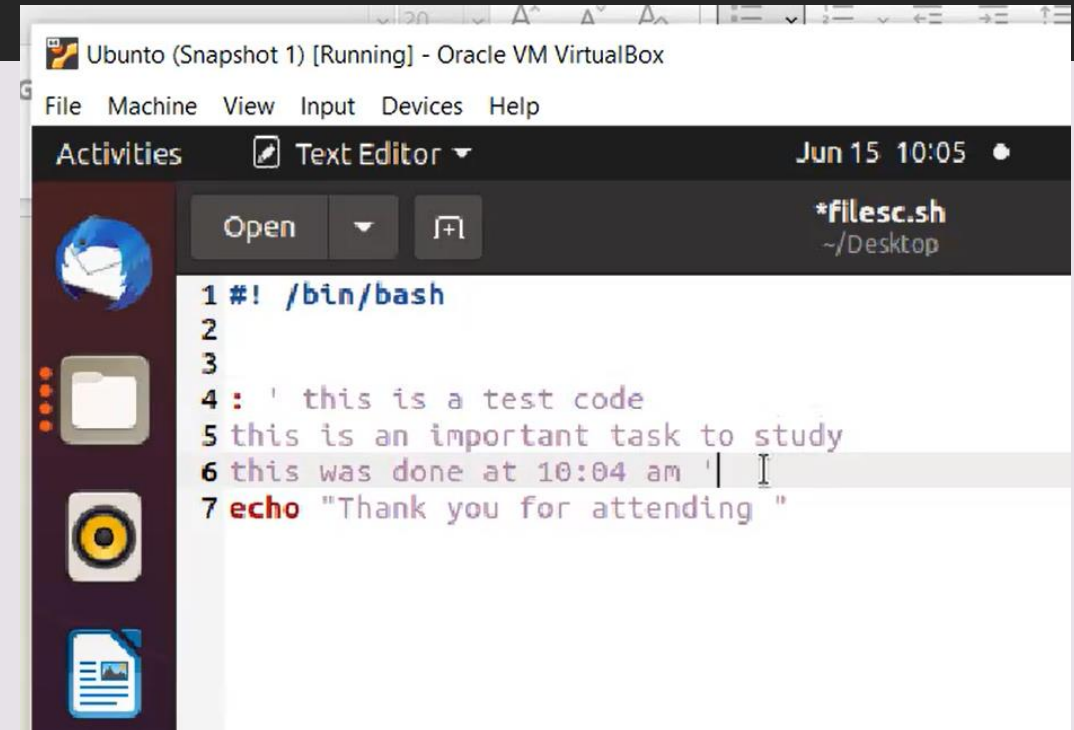
```
zahra@zahra-VirtualBox:~/Desktop$ ls
dir1      fail.txt  filebash.sh  fileshell.sh  outfile.txt  prfile.cpp
error.txt file1     rilec.sh     nydir         output.txt   profile.cpp
fail.txt  file4    filedirection outfile.tex   output.txt
zahra@zahra-VirtualBox:~/Desktop$ cat filedirection
Hello World!
zahra@zahra-VirtualBox:~/Desktop$
```

Shell Script

Comment in file

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

`:' this is a comment '`



The screenshot shows a text editor window titled "Ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox". The window has a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". Below the menu bar is a toolbar with "Activities", "Text Editor", and a date/time display "Jun 15 10:05". The editor is editing a file named "fileesc.sh" located at "~/Desktop". The file content is as follows:

```
1 #! /bin/bash
2
3
4 : ' this is a test code
5 this is an important task to study
6 this was done at 10:04 am ' |
7 echo "Thank you for attending "
```

[Back to the table of contents](#)



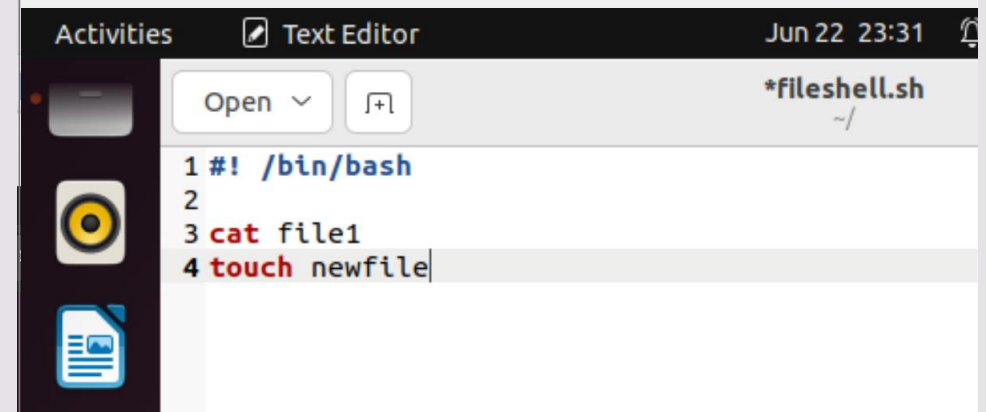
Shell Script

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.**

A screenshot of a Linux desktop environment. The top bar shows 'Activities', 'Text Editor', and the date 'Jun 22 23:31'. The main window is a text editor titled '*fileshell.sh' with a file icon. The editor contains four lines of code: '1 #! /bin/bash', '2', '3 cat file1', and '4 touch newfile'. The left sidebar shows icons for a file manager, a terminal, and a document.

```
1 #! /bin/bash
2
3 cat file1
4 touch newfile
```



Shell Script

If statement (if , then)

- **Syntax:**

```
count=10
```

```
if [ $count -eq 10 ]
```

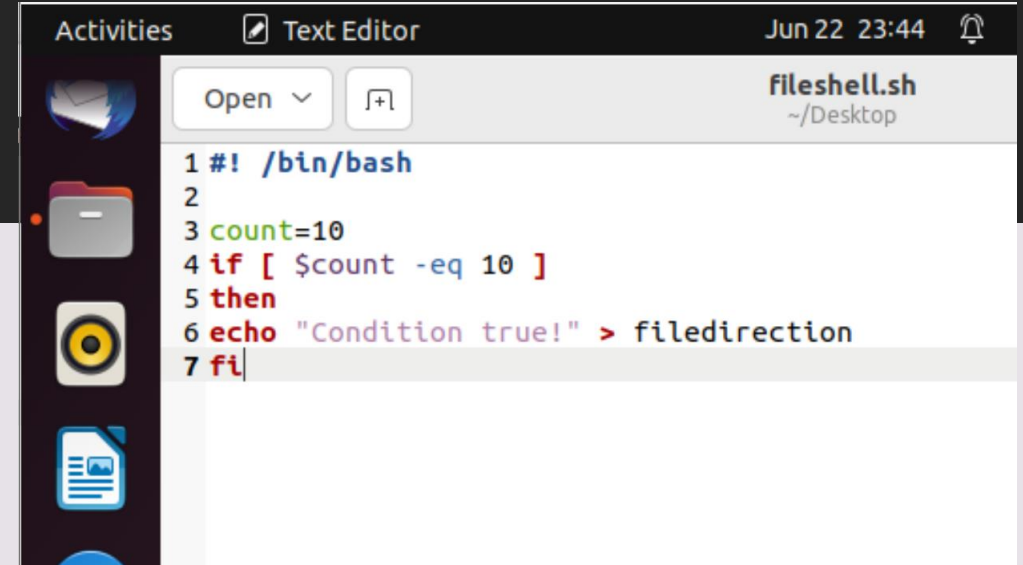
```
Then
```

```
echo "true" >>filedirection
```

```
fi
```

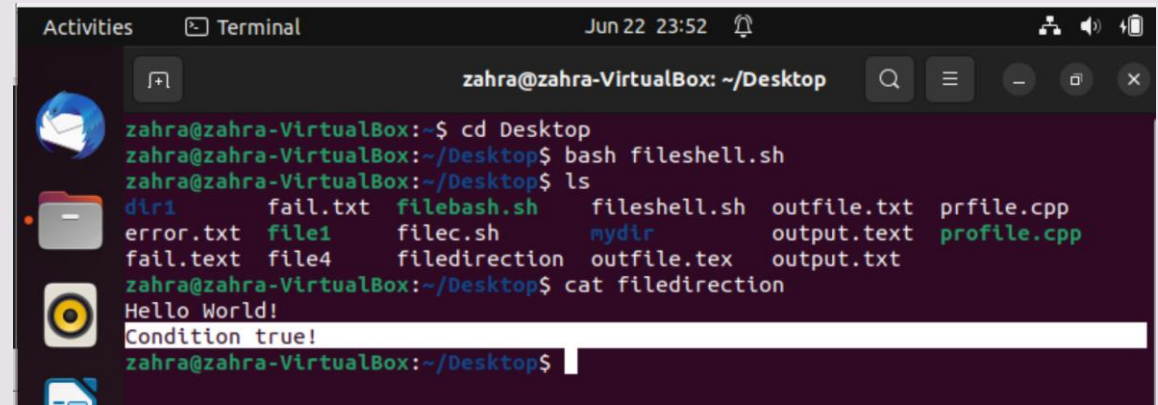
- In this example checking the value of the variable if the condition is true it will append filedirection.

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



The screenshot shows a text editor window titled 'fileshell.sh' located at '~/Desktop'. The script content is as follows:

```
1 #!/bin/bash
2
3 count=10
4 if [ $count -eq 10 ]
5 then
6 echo "Condition true!" > filedirection
7 fi
```



The screenshot shows a terminal window titled 'zahra@zahra-VirtualBox: ~/Desktop'. The user has executed the following commands:

```
zahra@zahra-VirtualBox:~$ cd Desktop
zahra@zahra-VirtualBox:~/Desktop$ bash fileshell.sh
zahra@zahra-VirtualBox:~/Desktop$ ls
dir1      fail.txt  filebash.sh  fileshell.sh  outfile.txt  prfile.cpp
error.txt file1     filec.sh     mydir         output.txt   profile.cpp
fail.txt  file4    filedirection outfile.tex   output.txt
zahra@zahra-VirtualBox:~/Desktop$ cat filedirection
Hello World!
Condition true!
zahra@zahra-VirtualBox:~/Desktop$
```

[Back to the table of contents](#)



Shell Script

If statement (if , else)

- **Syntax:**

```
count=1
```

```
if [ $count -eq 10 ] && [ $count -lt 12 ]
```

```
Then
```

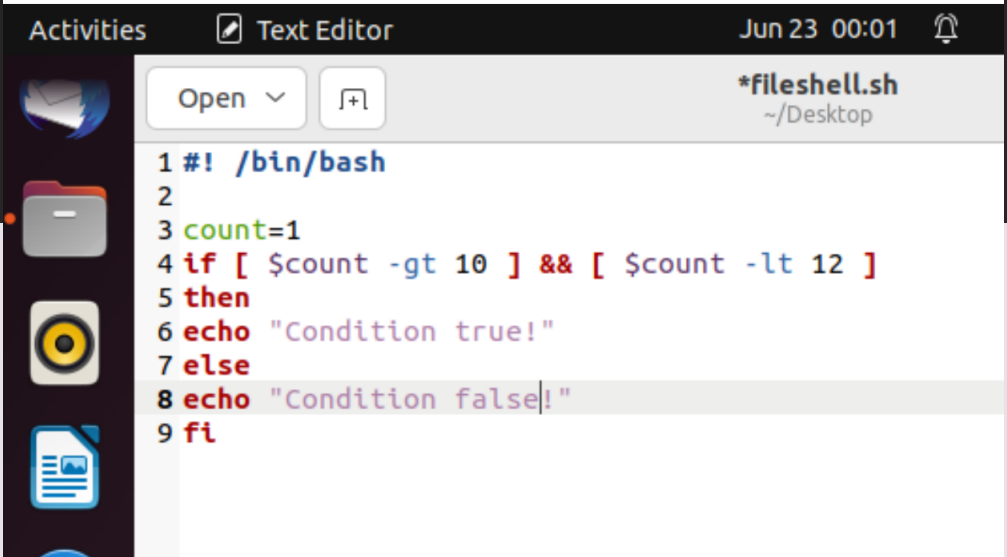
```
echo "condition true"
```

```
Else
```

```
echo "condition true"
```

```
fi
```

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



```
1 #!/bin/bash
2
3 count=1
4 if [ $count -gt 10 ] && [ $count -lt 12 ]
5 then
6 echo "Condition true!"
7 else
8 echo "Condition false!"
9 fi
```

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



Shell Script

Loop (while, do)

- **Syntax:**

```
count=1
```

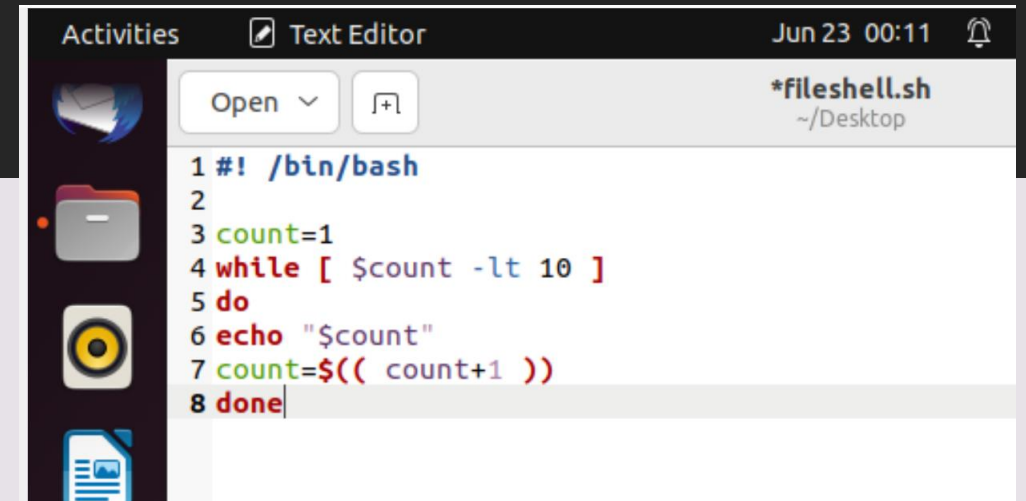
```
while [ $count -lt 10]
```

```
do
```

```
echo "$count"
```

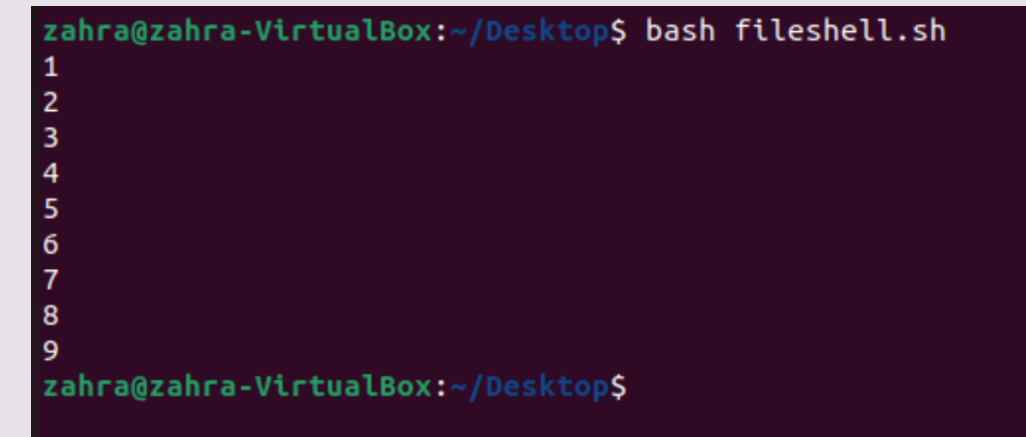
```
count=$(( count+1 ))
```

```
done
```



The screenshot shows a text editor window titled "Text Editor" with a file named "fileshell.sh" located on the Desktop. The script content is as follows:

```
1 #!/bin/bash
2
3 count=1
4 while [ $count -lt 10 ]
5 do
6 echo "$count"
7 count=$(( count+1 ))
8 done
```



The screenshot shows a terminal window with the prompt "zahra@zahra-VirtualBox:~/Desktop\$". The command "bash fileshell.sh" has been entered and executed. The terminal output shows the script running, with the count variable being printed from 1 to 9. The prompt returns to "zahra@zahra-VirtualBox:~/Desktop\$".

```
zahra@zahra-VirtualBox:~/Desktop$ bash fileshell.sh
1
2
3
4
5
6
7
8
9
zahra@zahra-VirtualBox:~/Desktop$
```

[Back to the table of contents](#)



Shell Script

Loop (for, do)

- **Syntax:**

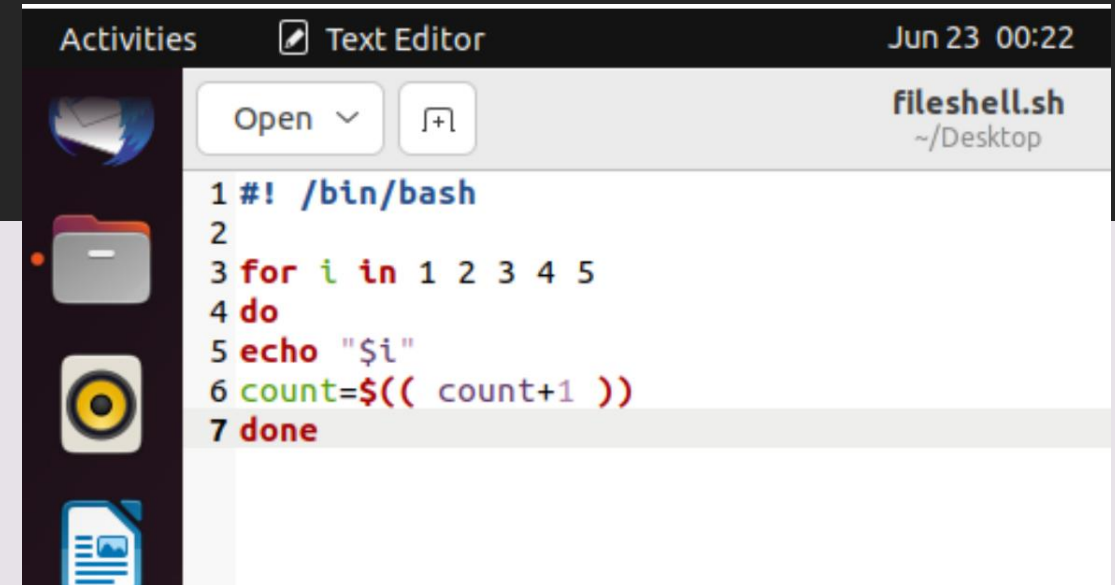
```
for i in 1 2 3 4 5
```

```
do
```

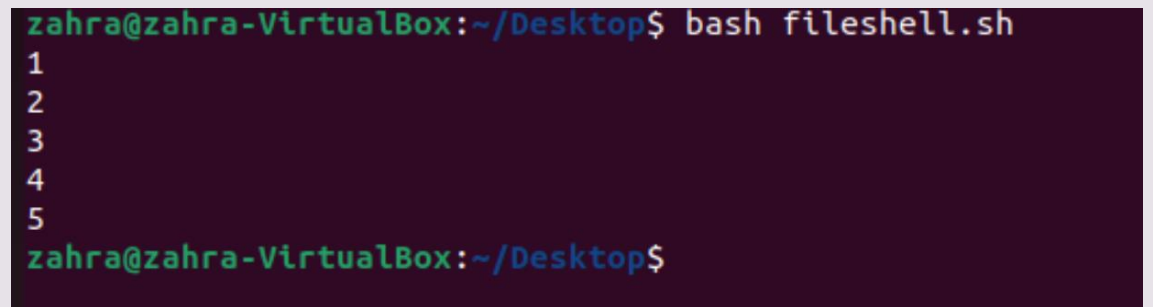
```
echo "$count"
```

```
count=$(( count+1 ))
```

```
done
```

A screenshot of a Linux desktop environment. The top panel shows the 'Activities' button, a 'Text Editor' window icon, and the date 'Jun 23 00:22'. The text editor window is titled 'fileshell.sh' and shows the following script:

```
1 #!/bin/bash
2
3 for i in 1 2 3 4 5
4 do
5 echo "$i"
6 count=$(( count+1 ))
7 done
```

A screenshot of a terminal window. The prompt is 'zahra@zahra-VirtualBox:~/Desktop\$'. The user has entered 'bash fileshell.sh'. The terminal shows the output of the script:

```
1
2
3
4
5
zahra@zahra-VirtualBox:~/Desktop$
```



Shell Script

Loop (for, do with steps)

- **Syntax:**

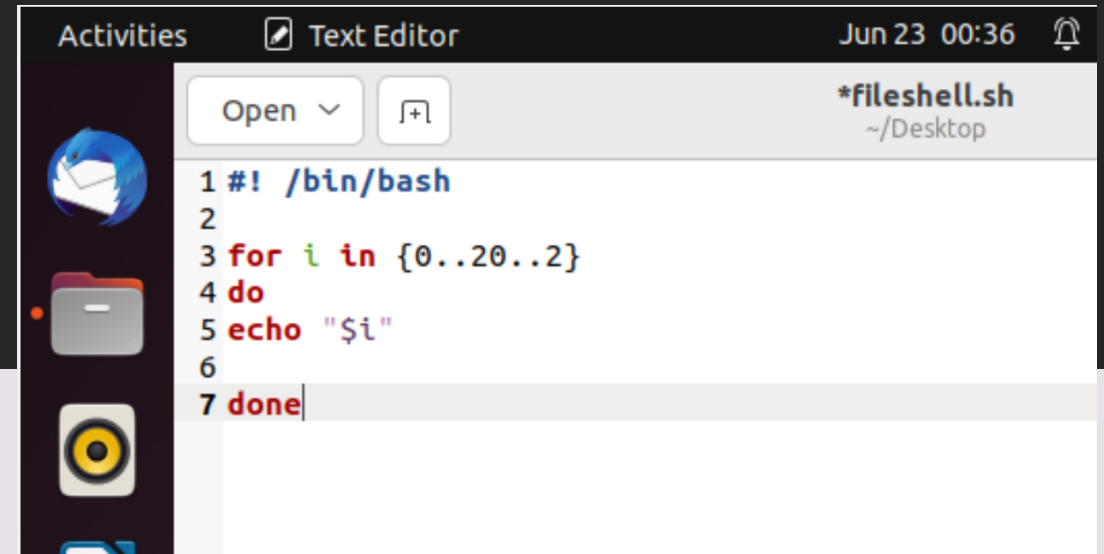
```
for i in { 0..20..2 }
```

```
do
```

```
echo "$i"
```

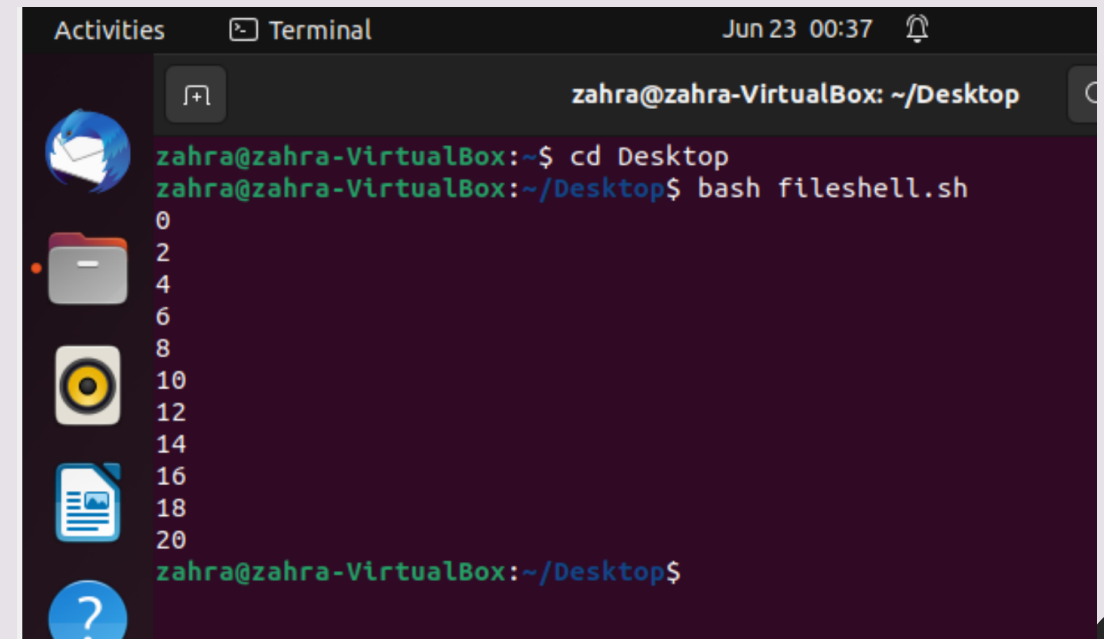
```
Done
```

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



The screenshot shows a text editor window titled 'Text Editor' with the file name '*fileshell.sh' and the path '~/Desktop'. The script content is as follows:

```
1 #! /bin/bash
2
3 for i in {0..20..2}
4 do
5 echo "$i"
6
7 done
```



The screenshot shows a terminal window titled 'Terminal' with the user 'zahra@zahra-VirtualBox' and the path '~/Desktop'. The terminal shows the following commands and output:

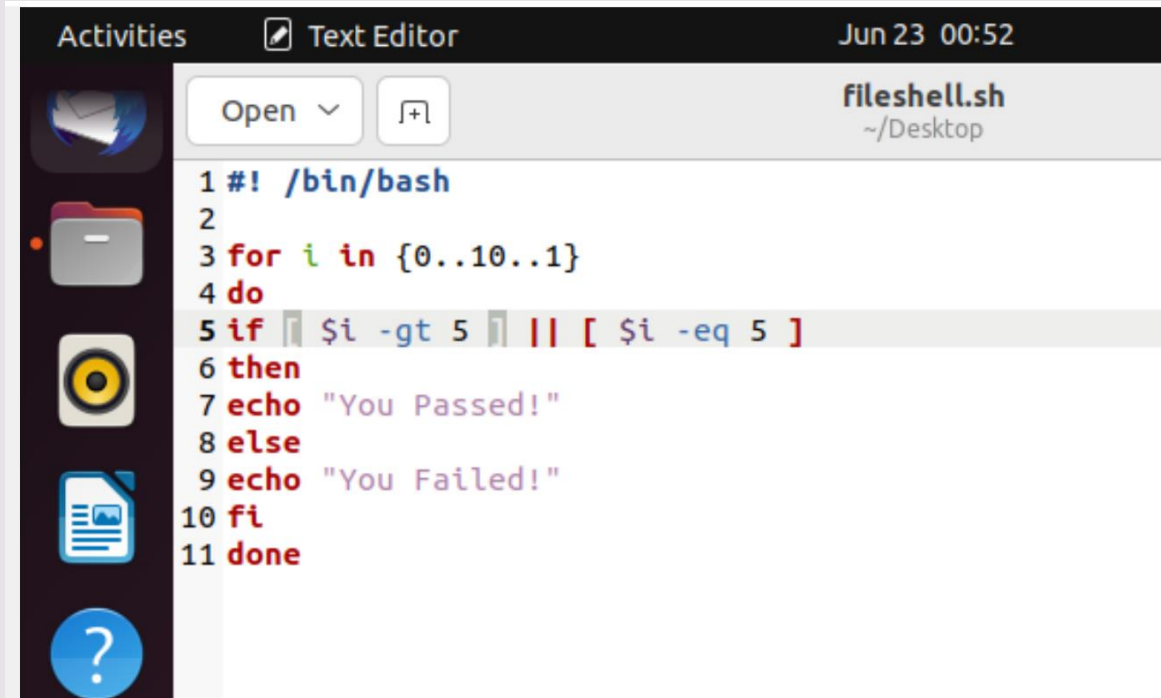
```
zahra@zahra-VirtualBox:~$ cd Desktop
zahra@zahra-VirtualBox:~/Desktop$ bash fileshell.sh
0
2
4
6
8
10
12
14
16
18
20
zahra@zahra-VirtualBox:~/Desktop$
```

[Back to the table of contents](#)



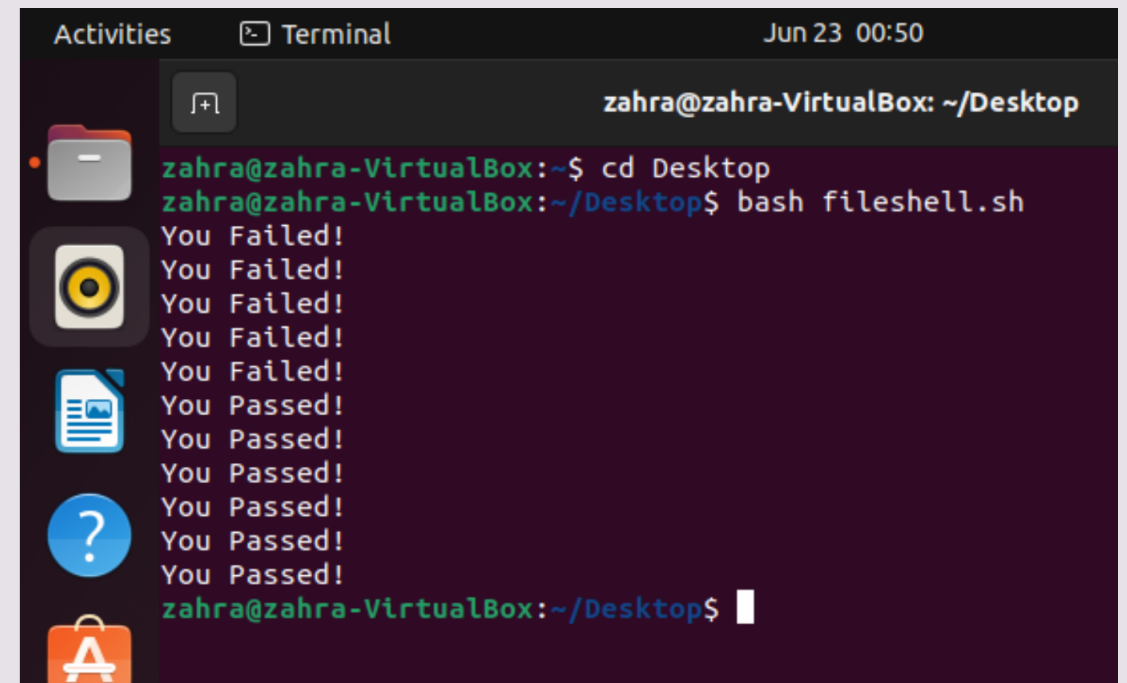
Shell Script

Combined loop & if



The screenshot shows a Text Editor window with the title bar "Text Editor" and the file name "fileshell.sh" located at "~/Desktop". The script content is as follows:

```
1 #!/bin/bash
2
3 for i in {0..10..1}
4 do
5 if [ $i -gt 5 ] || [ $i -eq 5 ]
6 then
7 echo "You Passed!"
8 else
9 echo "You Failed!"
10 fi
11 done
```



The screenshot shows a Terminal window with the title bar "Terminal" and the user prompt "zahra@zahra-VirtualBox: ~/Desktop". The terminal shows the following commands and output:

```
zahra@zahra-VirtualBox:~$ cd Desktop
zahra@zahra-VirtualBox:~/Desktop$ bash fileshell.sh
You Failed!
You Failed!
You Failed!
You Failed!
You Failed!
You Passed!
You Passed!
You Passed!
You Passed!
You Passed!
You Passed!
You Passed!
zahra@zahra-VirtualBox:~/Desktop$
```



Shell Script

Break a loop

- **Syntax:**

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

```
do
```

```
    if [ $i -gt 18 ]
```

```
    then
```

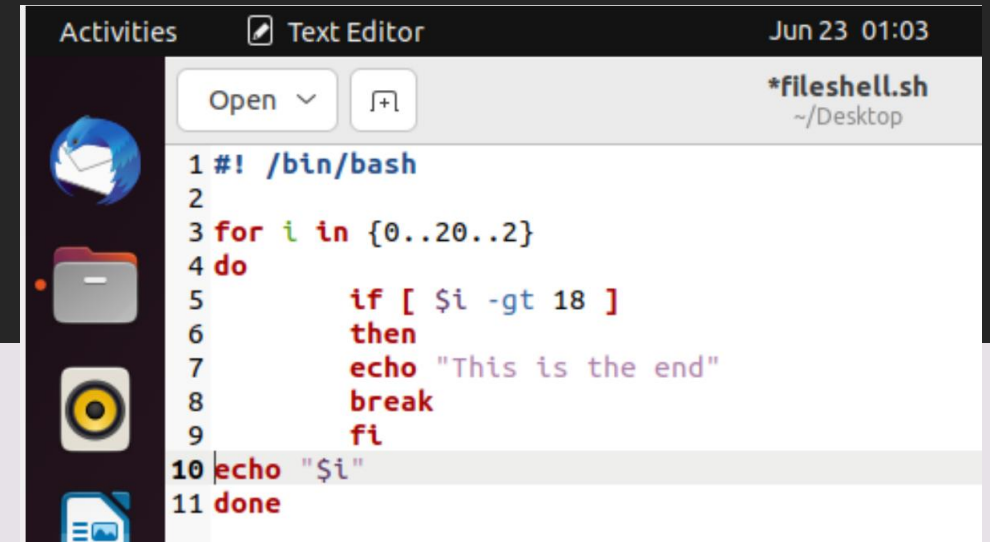
```
        break
```

```
    fi
```

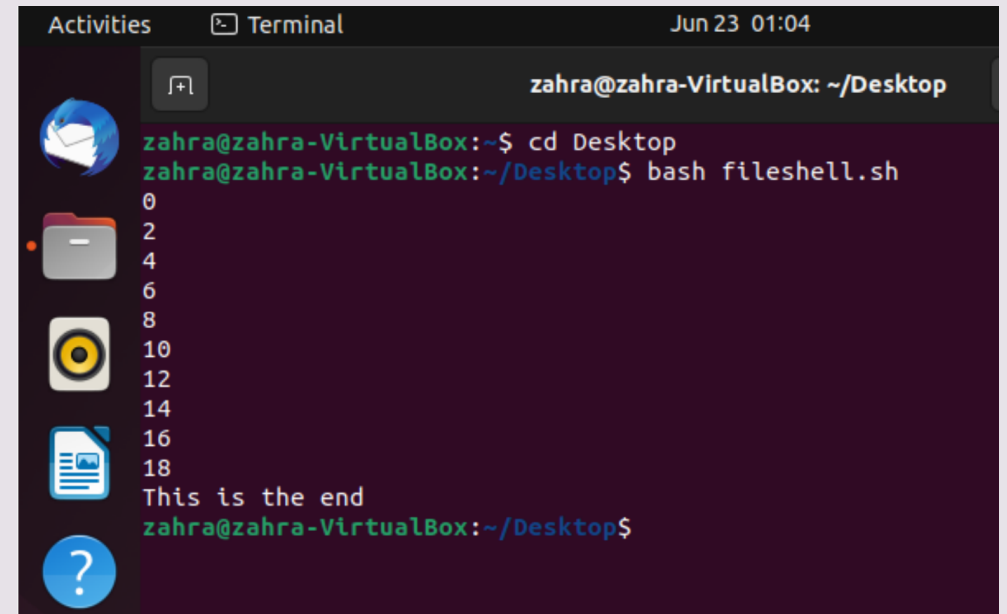
```
    echo "$i"
```

```
done
```

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



```
1 #!/bin/bash
2
3 for i in {0..20..2}
4 do
5     if [ $i -gt 18 ]
6     then
7         echo "This is the end"
8         break
9     fi
10 echo "$i"
11 done
```



```
zahra@zahra-VirtualBox: ~/Desktop
zahra@zahra-VirtualBox:~$ cd Desktop
zahra@zahra-VirtualBox:~/Desktop$ bash fileshell.sh
0
2
4
6
8
10
12
14
16
18
This is the end
zahra@zahra-VirtualBox:~/Desktop$
```

[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
```

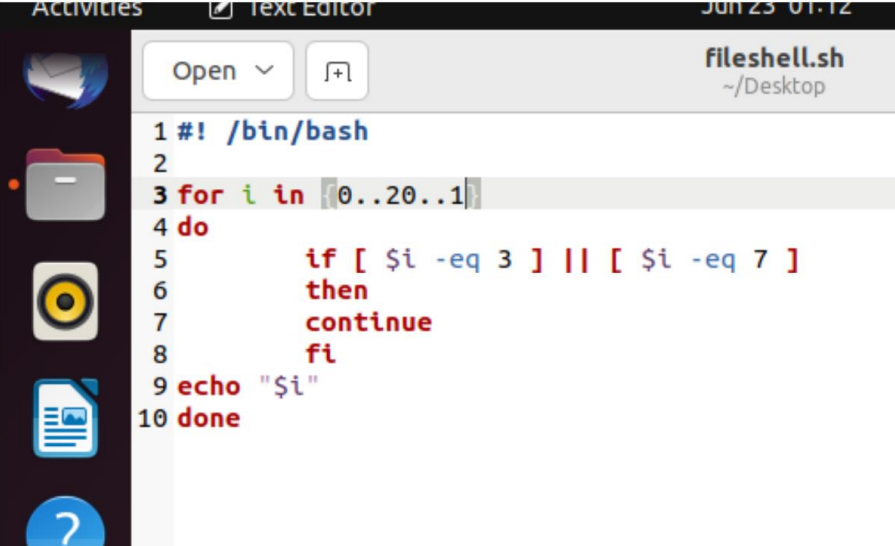
```
        continue
```

```
    fi
```

```
    echo "$i"
```

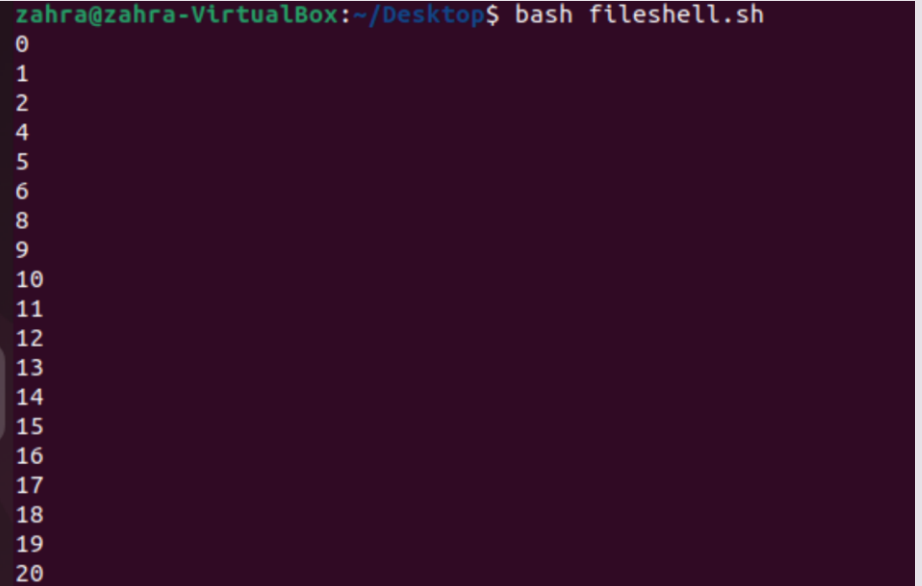
```
done
```

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



The screenshot shows a text editor window titled 'fileshell.sh' located at '~/Desktop'. The script content is as follows:

```
1 #!/bin/bash
2
3 for i in {0..20..1}
4 do
5     if [ $i -eq 3 ] || [ $i -eq 7 ]
6     then
7         continue
8     fi
9     echo "$i"
10 done
```



The screenshot shows a terminal window with the command 'bash fileshell.sh' executed. The output is a list of numbers from 0 to 20, with 3 and 7 omitted:

```
zahra@zahra-VirtualBox:~/Desktop$ bash fileshell.sh
0
1
2
4
5
6
8
9
10
11
12
13
14
15
16
17
18
19
20
```

[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
```

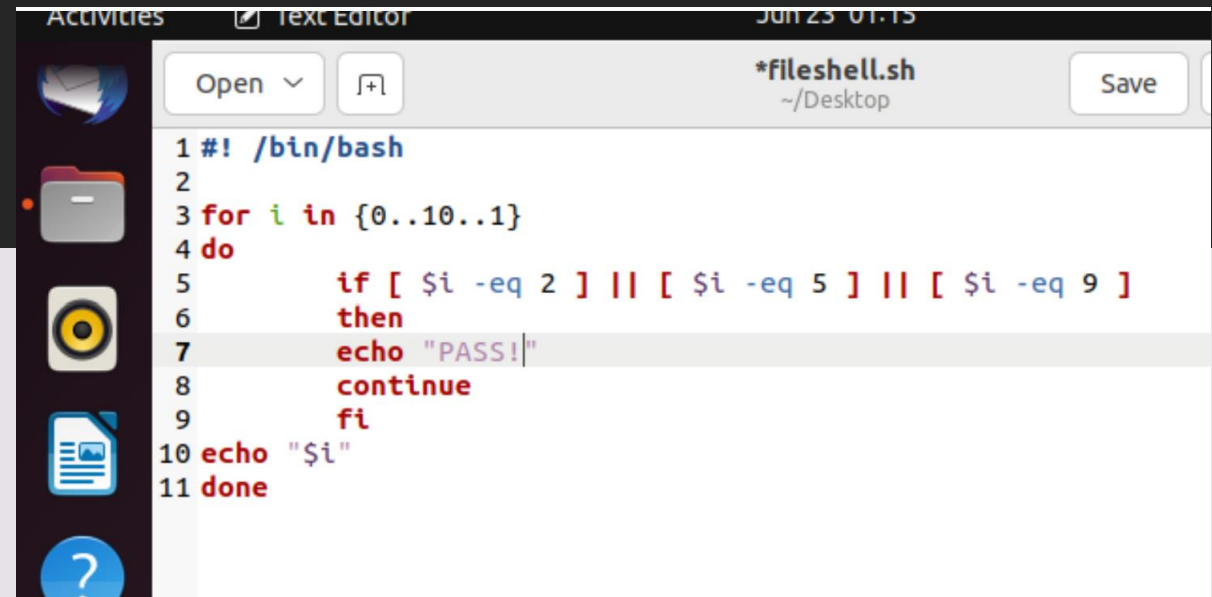
```
        echo "PASS!"
```

```
        continue
```

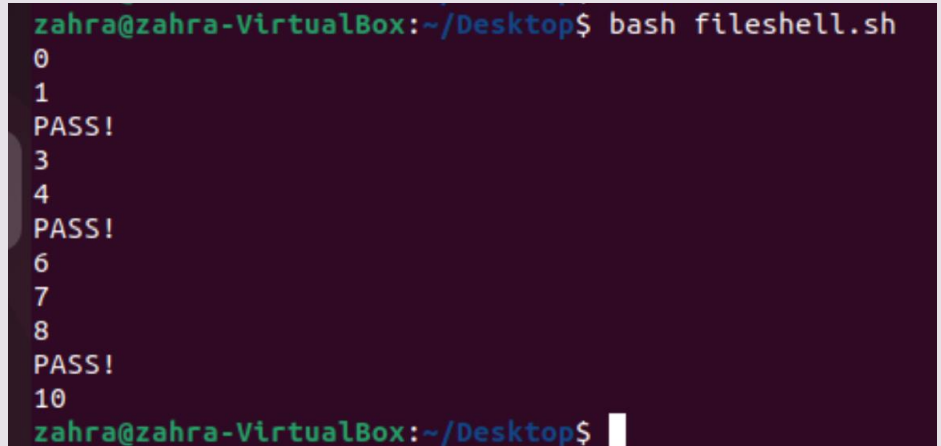
```
    fi
```

```
done
```

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



```
1 #!/bin/bash
2
3 for i in {0..10..1}
4 do
5     if [ $i -eq 2 ] || [ $i -eq 5 ] || [ $i -eq 9 ]
6     then
7         echo "PASS!"
8         continue
9     fi
10 echo "$i"
11 done
```



```
zahra@zahra-VirtualBox:~/Desktop$ bash fileshell.sh
0
1
PASS!
3
4
PASS!
6
7
PASS!
8
10
zahra@zahra-VirtualBox:~/Desktop$
```

[Back to the table of contents](#)



Perl

Create & run

- **Syntax:**

`#!/usr/bin/perl`

```
print( "What is your name?" );
```

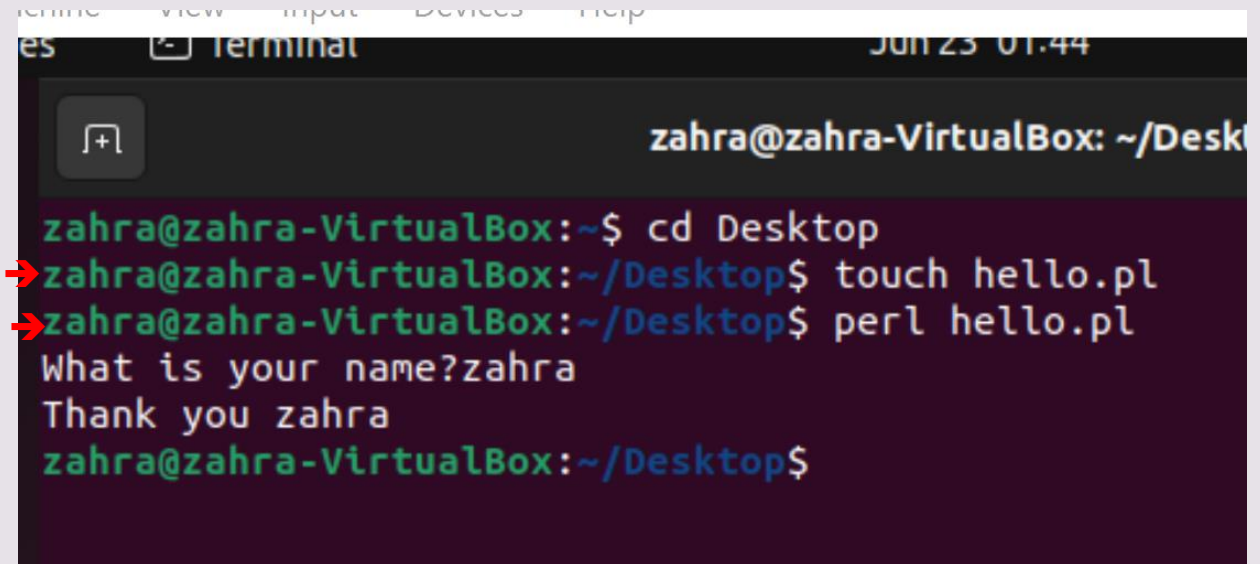
```
$name=<STDIN>;
```

```
print("Thank you $name");
```

Save file hello.pl

Create file →

Run the file →



The screenshot shows a terminal window titled 'Terminal' with a menu bar containing 'File', 'View', 'Input', 'Devices', and 'Help'. The terminal prompt is 'zahra@zahra-VirtualBox: ~/Desk'. The user enters 'cd Desktop' and the prompt changes to 'zahra@zahra-VirtualBox: ~/Desktop'. Then, the user enters 'touch hello.pl'. Next, the user enters 'perl hello.pl'. The program prompts 'What is your name?' and the user enters 'zahra'. The program then prints 'Thank you zahra'. Finally, the user enters a new command, and the prompt returns to 'zahra@zahra-VirtualBox: ~/Desktop\$'.

```
zahra@zahra-VirtualBox: ~/Desk
zahra@zahra-VirtualBox:~$ cd Desktop
zahra@zahra-VirtualBox:~/Desktop$ touch hello.pl
zahra@zahra-VirtualBox:~/Desktop$ perl hello.pl
What is your name?zahra
Thank you zahra
zahra@zahra-VirtualBox:~/Desktop$
```

[Back to the table of contents](#)



Perl

If statement

- **Syntax:**

```
#!/usr/bin/perl
```

```
print( "Enter your grade" );
```

```
$a=<STDIN>;
```

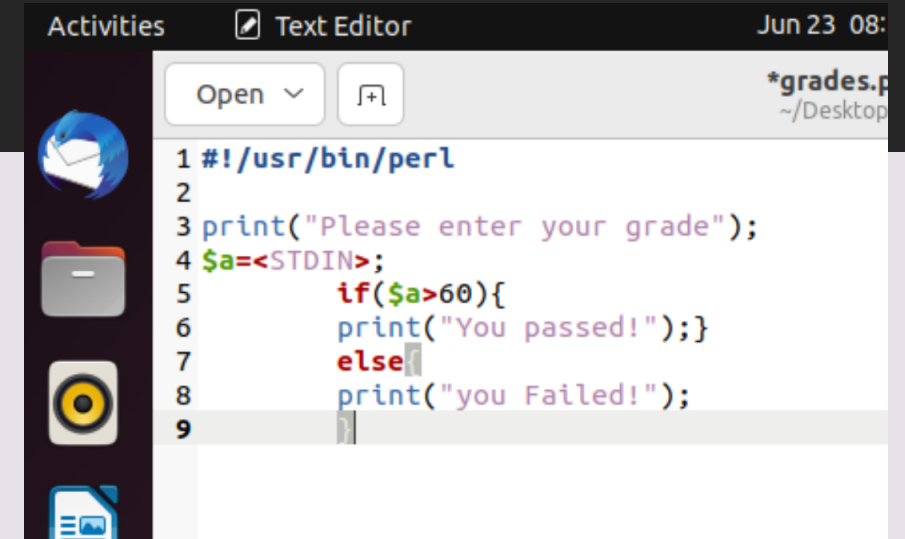
```
    if($a>60) {
```

```
        print("You Passed!");}
```

```
    else {
```

```
        print("You Failed!");}
```

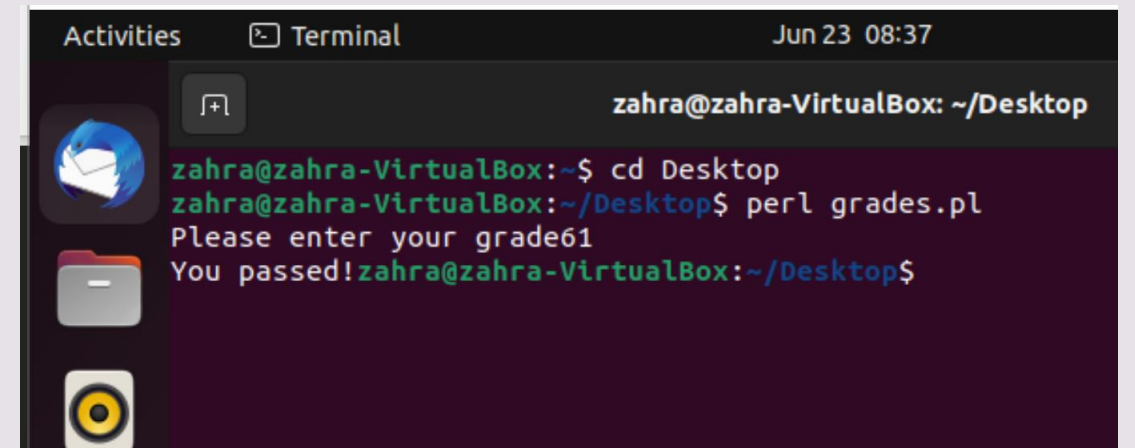
Save file grades.pl



Activities Text Editor Jun 23 08:37

*grades.pl ~/Desktop

```
1 #!/usr/bin/perl
2
3 print("Please enter your grade");
4 $a=<STDIN>;
5     if($a>60){
6         print("You passed!");}
7     else{
8         print("you Failed!");}
9
```



Activities Terminal Jun 23 08:37

zahra@zahra-VirtualBox: ~/Desktop

```
zahra@zahra-VirtualBox:~$ cd Desktop
zahra@zahra-VirtualBox:~/Desktop$ perl grades.pl
Please enter your grade61
You passed!zahra@zahra-VirtualBox:~/Desktop$
```

[Back to the table of contents](#)



Perl

Arrays

- **Syntax:**

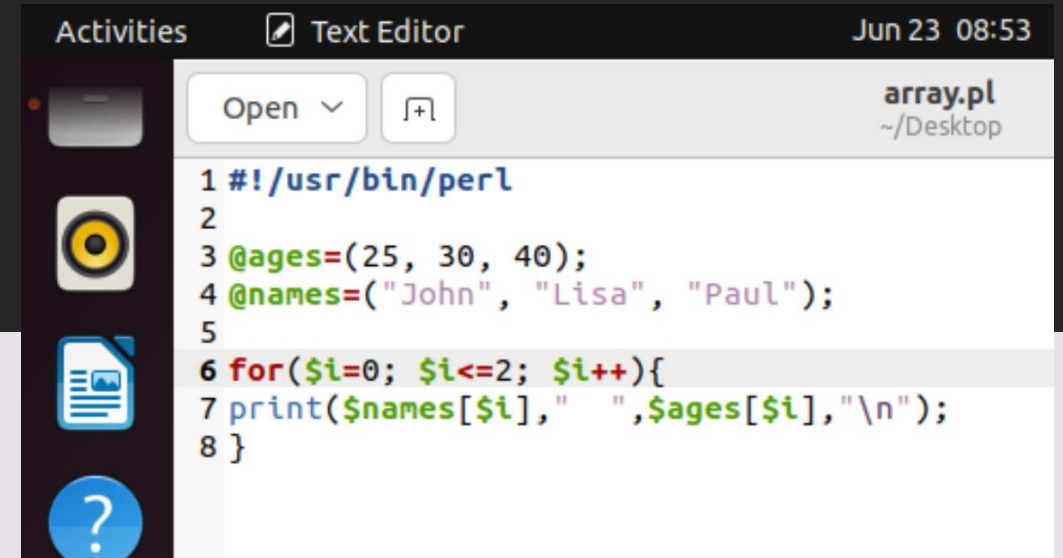
```
#!/usr/bin/perl
```

```
@ages = (25, 30, 40);
```

```
@names = ("John ", "Lisa", "Paul");
```

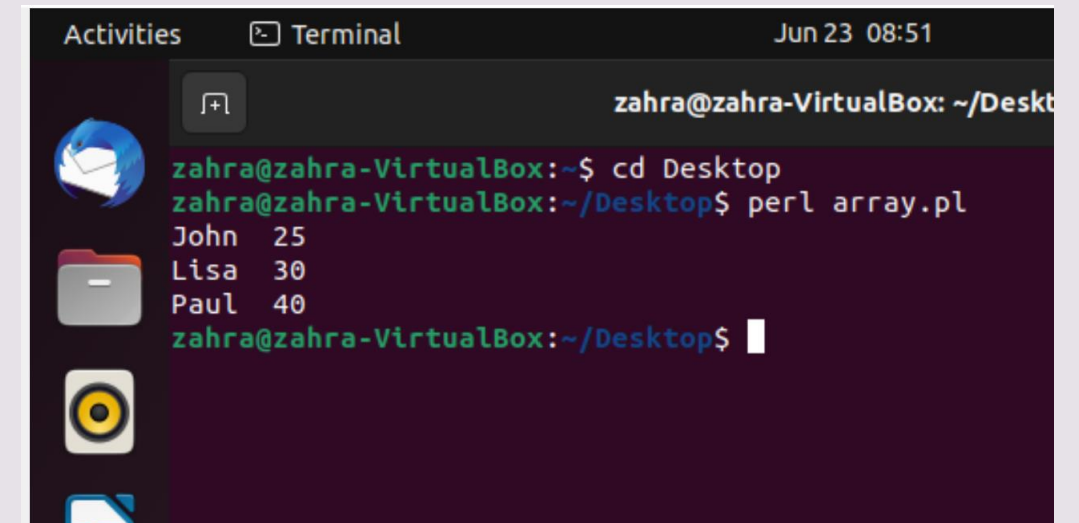
```
    for ($i=0; $i<=2; $i++){  
        print($names[$i], " ", $ages[$i], "\n");  
    }
```

Save file array.pl



The screenshot shows a text editor window titled 'array.pl' located at '~/Desktop'. The code inside is a Perl script that defines two arrays, @ages and @names, and then iterates over @names to print each name followed by its corresponding age from @ages.

```
1 #!/usr/bin/perl  
2  
3 @ages=(25, 30, 40);  
4 @names=("John", "Lisa", "Paul");  
5  
6 for($i=0; $i<=2; $i++){  
7     print($names[$i], " ", $ages[$i], "\n");  
8 }
```



The screenshot shows a terminal window with the user 'zahra' at a 'VirtualBox' prompt. The user has navigated to the 'Desktop' directory and executed the command 'perl array.pl'. The output of the script is displayed, showing the names and ages formatted as 'Name Age'.

```
zahra@zahra-VirtualBox: ~/Desktop  
zahra@zahra-VirtualBox:~$ cd Desktop  
zahra@zahra-VirtualBox:~/Desktop$ perl array.pl  
John 25  
Lisa 30  
Paul 40  
zahra@zahra-VirtualBox:~/Desktop$
```



File compression

gzip technique

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

```
zahra@zahra-VirtualBox: ~/Desktop
zahra@zahra-VirtualBox:~$ cd Desktop
zahra@zahra-VirtualBox:~/Desktop$ touch compfile
zahra@zahra-VirtualBox:~/Desktop$ time gzip compfile

real    0m0.001s
user    0m0.001s
sys     0m0.000s
zahra@zahra-VirtualBox:~/Desktop$ gunzip compfile.gz
zahra@zahra-VirtualBox:~/Desktop$ ls -lh
total 72K
-rw-rw-r-- 1 zahra zahra 136 Jun 23 08:51 array.pl
-rw-rw-r-- 1 zahra zahra  0 Jun 23 10:27 compfile
drwxrwx--- 2 root  root 4.0K Jun  9 22:16 dir1
-rw-rw-r-- 1 zahra zahra 27 Jun 14 00:43 error.txt
-rw-rw-r-- 1 zahra zahra 40 Jun 13 23:52 fail.text
-rw-rw-r-- 1 zahra zahra 40 Jun 14 00:11 fail.txt
-rwxrwx--- 1 zahra zahra 30 Jun 13 23:41 file1
-rw-rw-r-- 1 zahra zahra 23 Jun 14 00:18 file4
-rw-rw-r-- 1 zahra zahra  0 Jun 22 22:56 filec.sh
-rw-rw-r-- 1 zahra zahra 29 Jun 22 23:51 filedirection
-rw-rw-r-- 1 zahra zahra 137 Jun 23 01:19 fileshell.sh
-rw-rw-r-- 1 zahra zahra 132 Jun 23 08:36 grades.pl
-rw-rw-r-- 1 zahra zahra 87 Jun 23 01:43 hello.pl
dr-xrw-r-- 2 zahra zahra 4.0K Jun 14 21:19 mydir
-rw-rw-r-- 1 zahra zahra 61 Jun 14 00:09 outfile.tex
-rw-rw-r-- 1 zahra zahra 61 Jun 14 00:15 outfile.txt
-rw-rw-r-- 1 zahra zahra 30 Jun 13 23:52 output.text
-rw-rw-r-- 1 zahra zahra 86 Jun 14 00:45 output.txt
-rw-rw-r-- 1 zahra zahra 145 Jun 13 09:50 prfile.cpp
```



File compression

bzip2 technique

- **bzip2 technique saves 15% more space (0.003 sec)**
- **To compress:** `bzip2 fileName`
- **To expand (extract):** `bunzip2 fileName.bz2`
- **Check compressed file size :** `ls -lh fileName.bz2`
- **Check regular file size :** `ls -lh fileName`
- **To calculate execution time:** `time bzip2 fileName`

```
zahra@zahra-VirtualBox:~/Desktop$ time bzip2 compfile
real    0m0.003s
user    0m0.002s
sys     0m0.000s
zahra@zahra-VirtualBox:~/Desktop$
```

