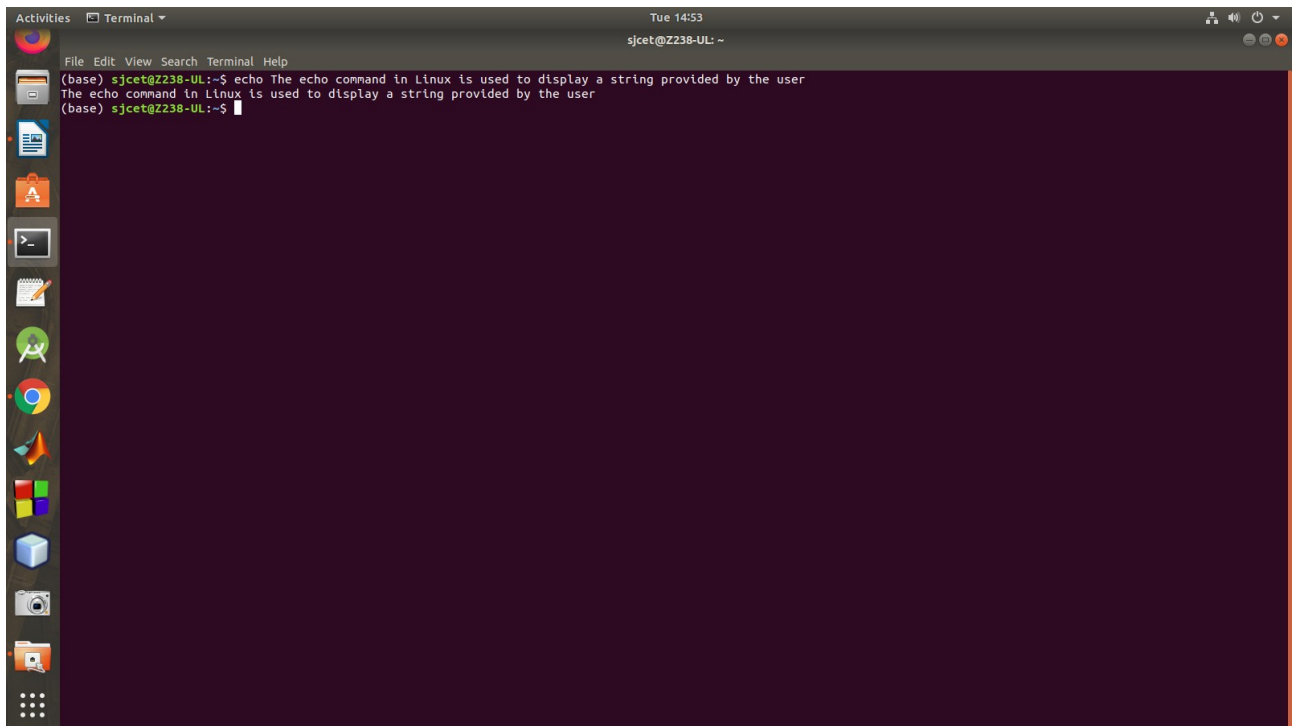


'echo' command in Linux

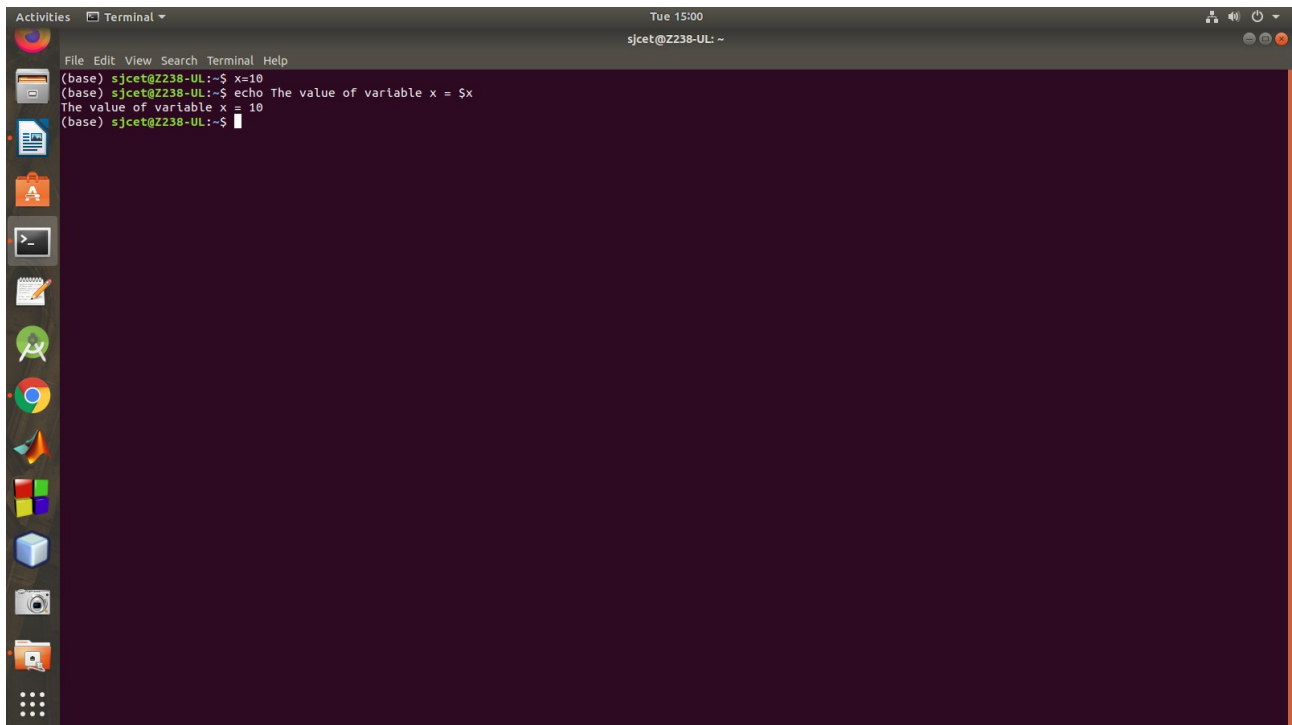
1. Input a line of text and display it on standard output



A screenshot of a Linux terminal window. The window title is 'Terminal' and the current user is 'sjcet@Z238-UL'. The terminal shows the command `echo The echo command in Linux is used to display a string provided by the user` being executed. The output of the command is displayed on the next line: `The echo command in Linux is used to display a string provided by the user`. The prompt `(base) sjcet@Z238-UL:~$` is visible at the end of the line.

```
(base) sjcet@Z238-UL:~$ echo The echo command in Linux is used to display a string provided by the user
The echo command in Linux is used to display a string provided by the user
(base) sjcet@Z238-UL:~$
```

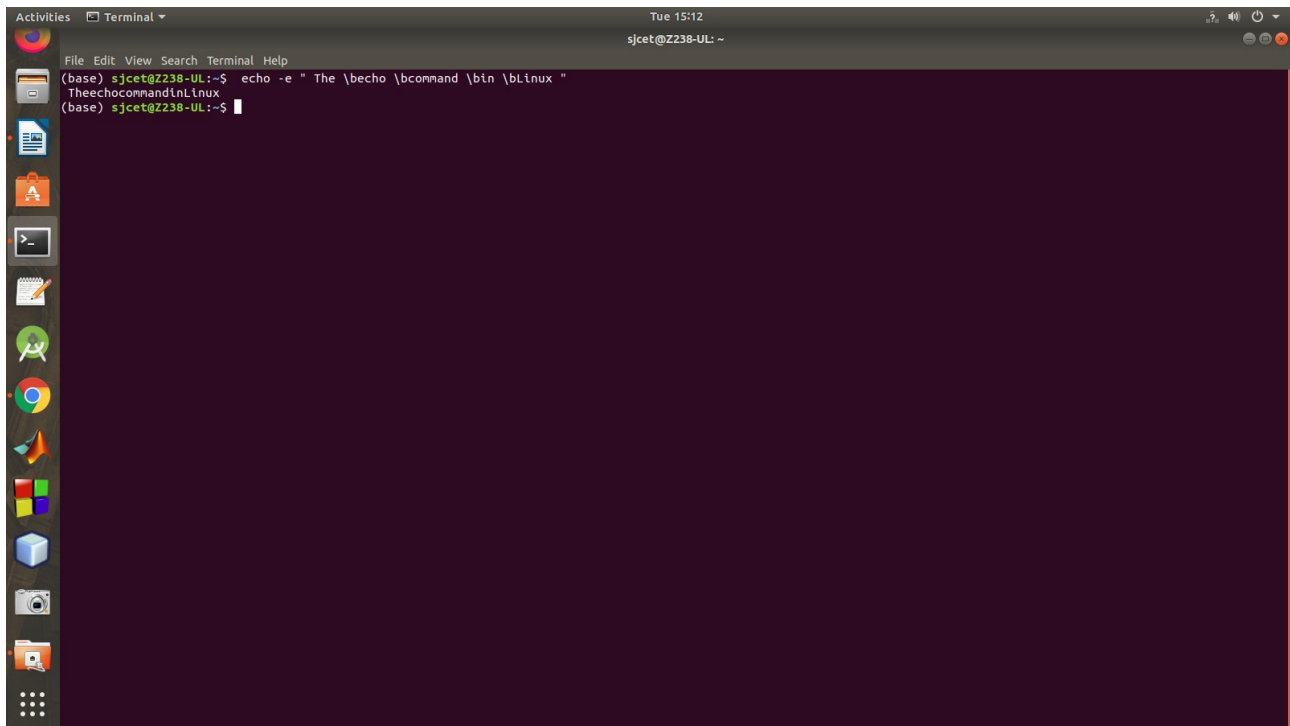
2. Declare a variable and echo its value. For example, Declare a variable of x and assign its value=10.



A screenshot of a Linux terminal window. The window title is 'Terminal' and the current user is 'sjcet@Z238-UL'. The terminal shows the command `x=10` being executed. The prompt `(base) sjcet@Z238-UL:~$` is visible at the end of the line. The next line shows the command `echo The value of variable x = $x` being executed. The output of the command is displayed on the next line: `The value of variable x = 10`. The prompt `(base) sjcet@Z238-UL:~$` is visible at the end of the line.

```
(base) sjcet@Z238-UL:~$ x=10
(base) sjcet@Z238-UL:~$ echo The value of variable x = $x
The value of variable x = 10
(base) sjcet@Z238-UL:~$
```

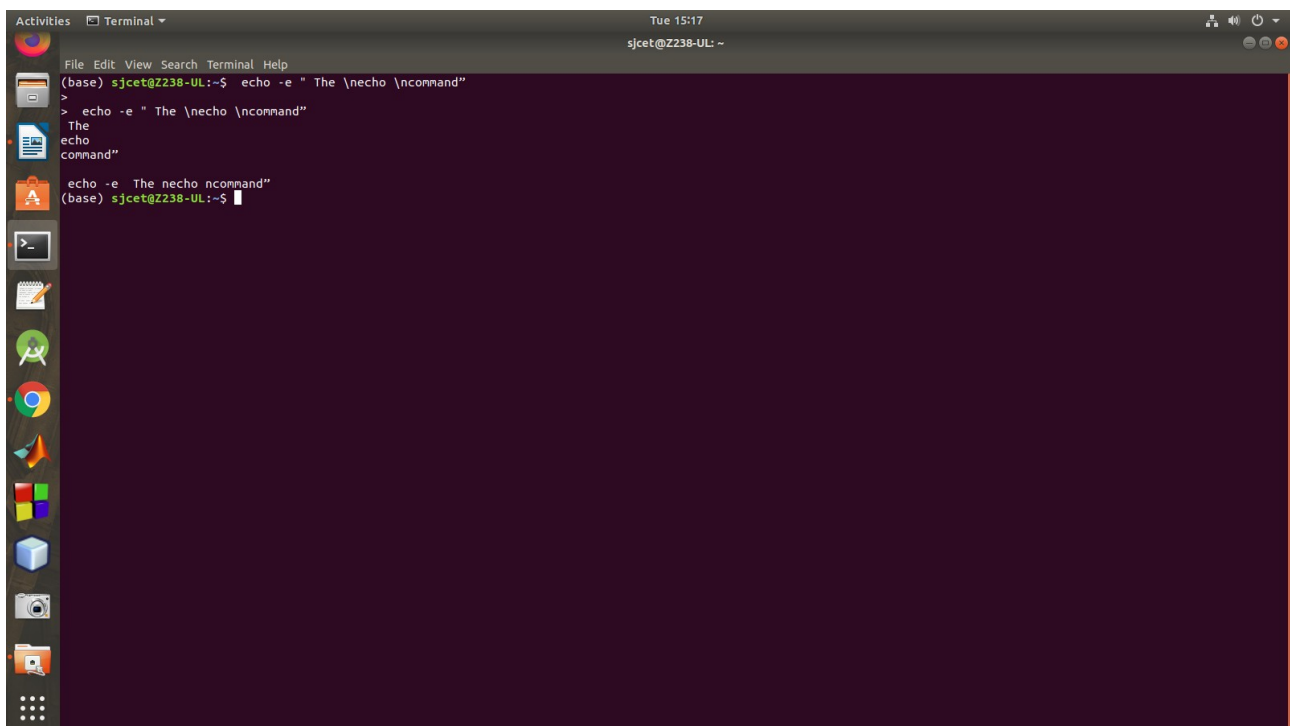
3. Using option '\b' - backspace with backslash interpreter '-e' which removes all the spaces in between.



A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Tue 15:12, sjcet@Z238-UL: ~). The terminal shows the following commands and output:

```
(base) sjcet@Z238-UL:~$ echo -e " The \becho \bcommand \bin \blinux "
TheechocommandinLinux
(base) sjcet@Z238-UL:~$
```

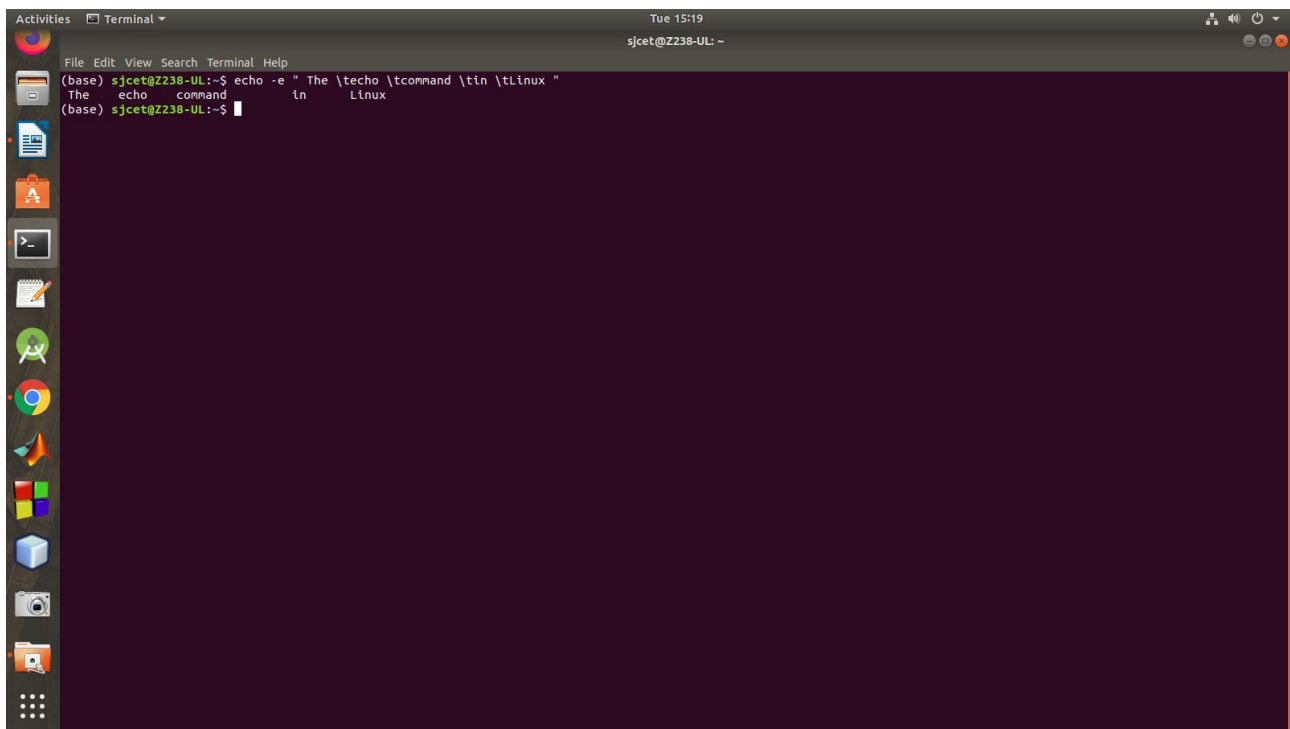
4. Using option '\n' - New line with backspace interpreter '-e' treats new line from where it is used.



A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Tue 15:17, sjcet@Z238-UL: ~). The terminal shows the following commands and output:

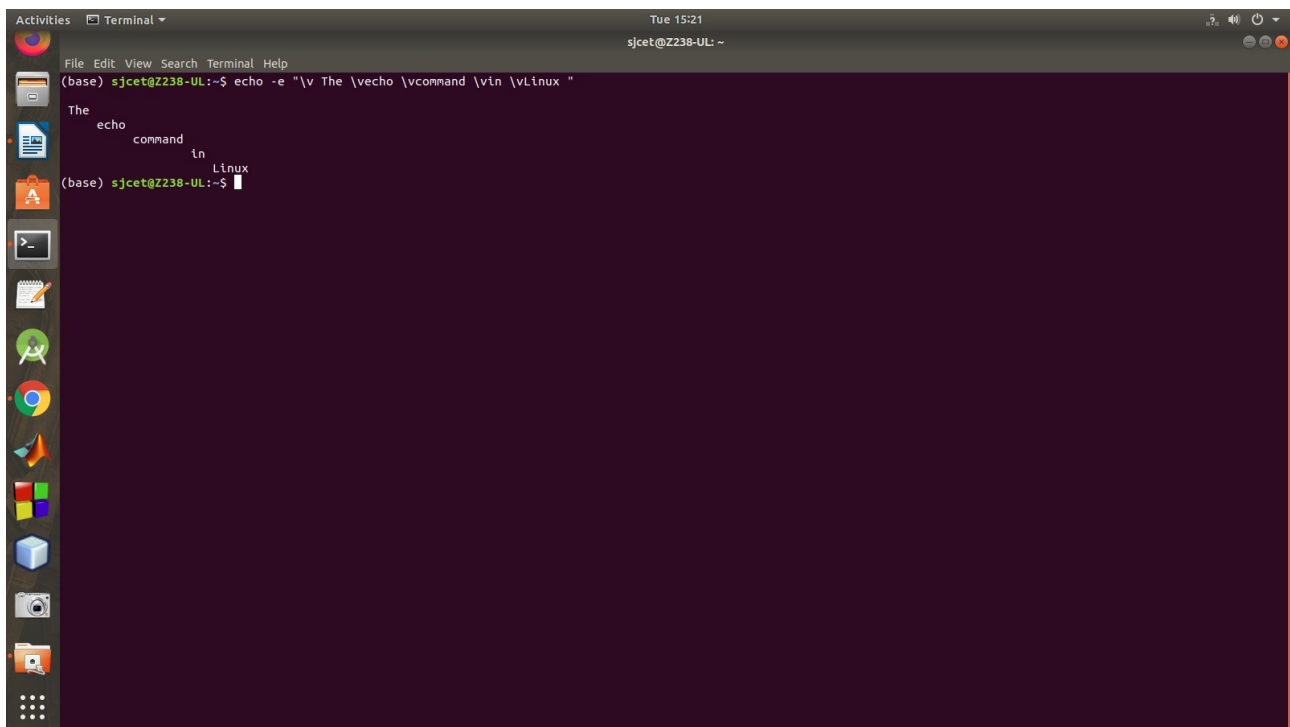
```
(base) sjcet@Z238-UL:~$ echo -e " The \necho \ncommand"
> echo -e " The \necho \ncommand"
The
echo
command"
echo -e " The necho ncommand"
(base) sjcet@Z238-UL:~$
```

5. Using option '\t' – horizontal tab with backspace interpreter '-e' to have horizontal tab spaces.



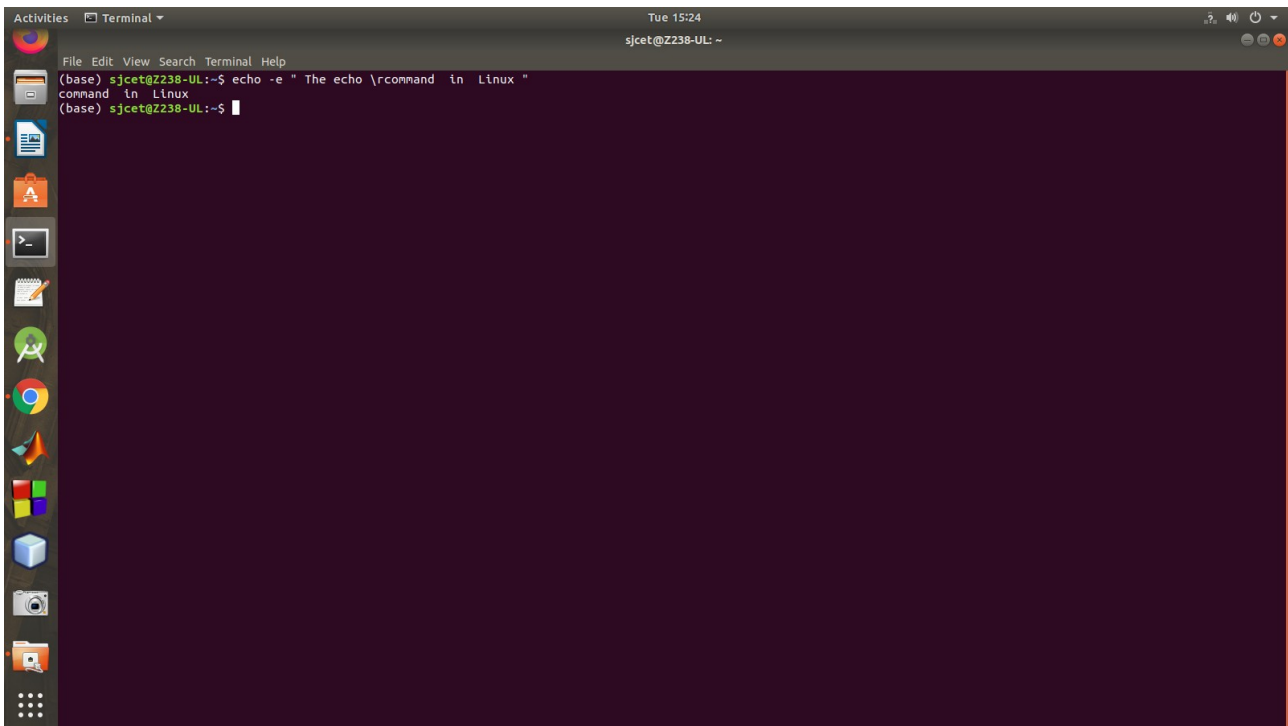
A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Tue 15:19, sjcet@Z238-UL: ~). The prompt is (base) sjcet@Z238-UL:~\$. The command entered is `echo -e "The \techo \tcommand \tin \tlinux "`. The output is displayed with horizontal tab spaces: `The echo command in Linux`. The prompt is (base) sjcet@Z238-UL:~\$.

6. Using option '\v' – vertical tab with backspace interpreter '-e' to have vertical tab spaces.



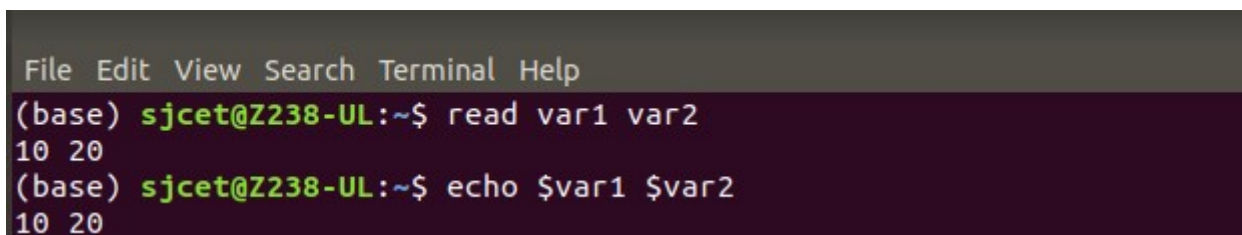
A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Tue 15:21, sjcet@Z238-UL: ~). The prompt is (base) sjcet@Z238-UL:~\$. The command entered is `echo -e "\v The \vecho \vcommand \vin \vLinux "`. The output is displayed with vertical tab spaces, showing the text on multiple lines: `The echo command in Linux`. The prompt is (base) sjcet@Z238-UL:~\$.

7. Using option '\r' - carriage return with backspace interpreter '-e' to have specified carriage return in output.

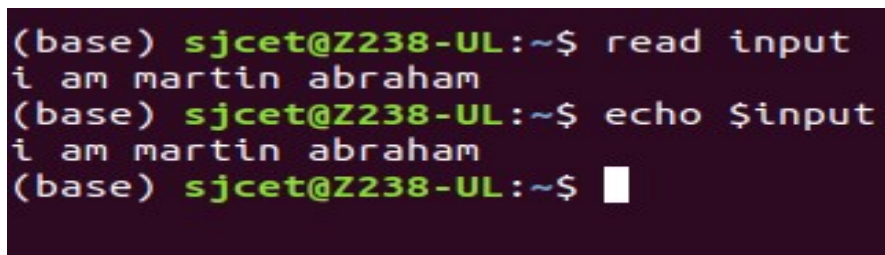


```
Activities  Terminal  Tue 15:24
sjcet@Z238-UL: ~
File Edit View Search Terminal Help
(base) sjcet@Z238-UL:~$ echo -e " The echo \rcommand in Linux "
command in Linux
(base) sjcet@Z238-UL:~$
```

READ



```
File Edit View Search Terminal Help
(base) sjcet@Z238-UL:~$ read var1 var2
10 20
(base) sjcet@Z238-UL:~$ echo $var1 $var2
10 20
```



```
(base) sjcet@Z238-UL:~$ read input
i am martin abraham
(base) sjcet@Z238-UL:~$ echo $input
i am martin abraham
(base) sjcet@Z238-UL:~$
```

```
(base) sjcet@Z238-UL:~$ read var1 var2
33 56
(base) sjcet@Z238-UL:~$ echo $var1
33
(base) sjcet@Z238-UL:~$ echo $var2
56
(base) sjcet@Z238-UL:~$
```

Piping

```
(base) sjcet@Z238-UL:~$ echo "the legend" | (read var1 var2; echo $var1; echo $var2)
the
legend
(base) sjcet@Z238-UL:~$
```

```
(base) sjcet@Z238-UL:~$ echo "rocky legend" | (read; echo "$REPLY")
rocky legend
(base) sjcet@Z238-UL:~$
```

```
(base) sjcet@Z238-UL:~$ echo "Linux is awesome." | (read var1 var2; echo -e "Var1: $var1 \nVar2: $var2")
Var1: Linux
Var2: is awesome.
(base) sjcet@Z238-UL:~$
```

Changing the Delimiter

```
(base) sjcet@Z238-UL:~$ echo "Linux:is:awesome." | (IFS=":" read -r var1 var2 var3; echo -e "$var1 \n$var2 \n$var3")
Linux
is
awesome.
(base) sjcet@Z238-UL:~$
```

```
(base) sjcet@Z238-UL:~$ echo "Linux::is:awesome." | \
> (IFS=":" read -r var1 var2 var3 var4; echo -e "Var1: $var1 \nVar2: $var2 \nVar3: $var3 \nVar4: $var4")
Var1: Linux
Var2:
Var3: is
Var4: awesome.
(base) sjcet@Z238-UL:~$
```

```
(base) sjcet@Z238-UL:~$ echo 'Linux_is-awesome.' | (IFS="-_" read -r var1 var2 var3; echo -e "$var1 \n$var2 \n$var3")
Linux
is
awesome.
(base) sjcet@Z238-UL:~$
```

Prompt String

```
(base) sjcet@Z238-UL:~$ while true; do
> read -r -p "Do you wish to reboot the system? (Y/N): " answer
> case $answer in
>     [Yy]* ) reboot; break;;
>     [Nn]* ) exit;;
>     * ) echo "Please answer Y or N.";;
> esac
> done
Do you wish to reboot the system? (Y/N):
```

```
Enter your password: (base) sjcet@Z238-UL:~$
(base) sjcet@Z238-UL:~$ read -r -s -p "Enter your password: "
Enter your password:
```

Assign the Words to Array

```
(base) sjcet@Z238-UL:~$ read -r -a MY_ARR <<< "Linux is awesome."
(base) sjcet@Z238-UL:~$
(base) sjcet@Z238-UL:~$ for i in "${MY_ARR[@]}"; do
> echo "$i"
> done
Linux
is
awesome.
(base) sjcet@Z238-UL:~$
```