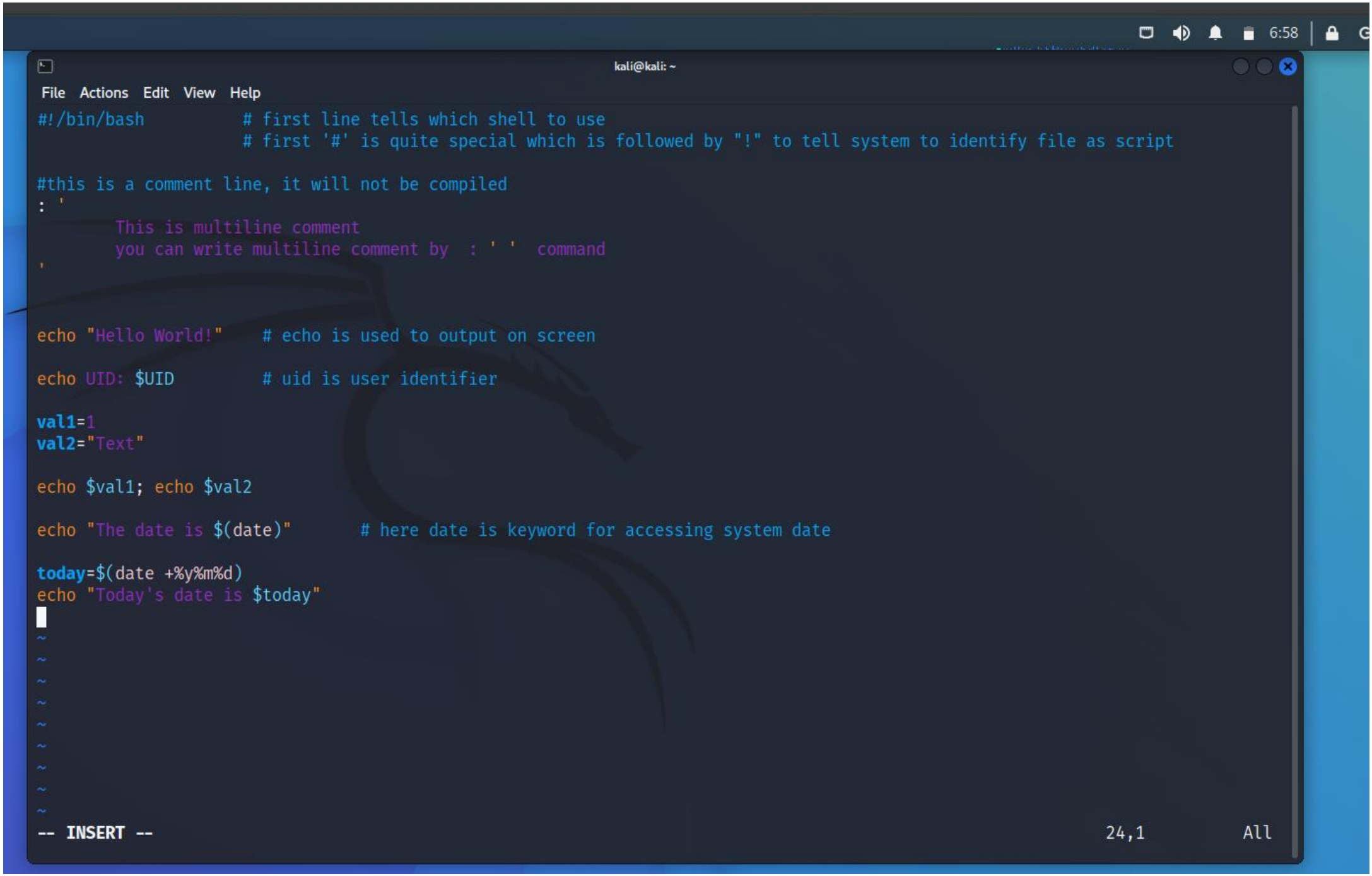
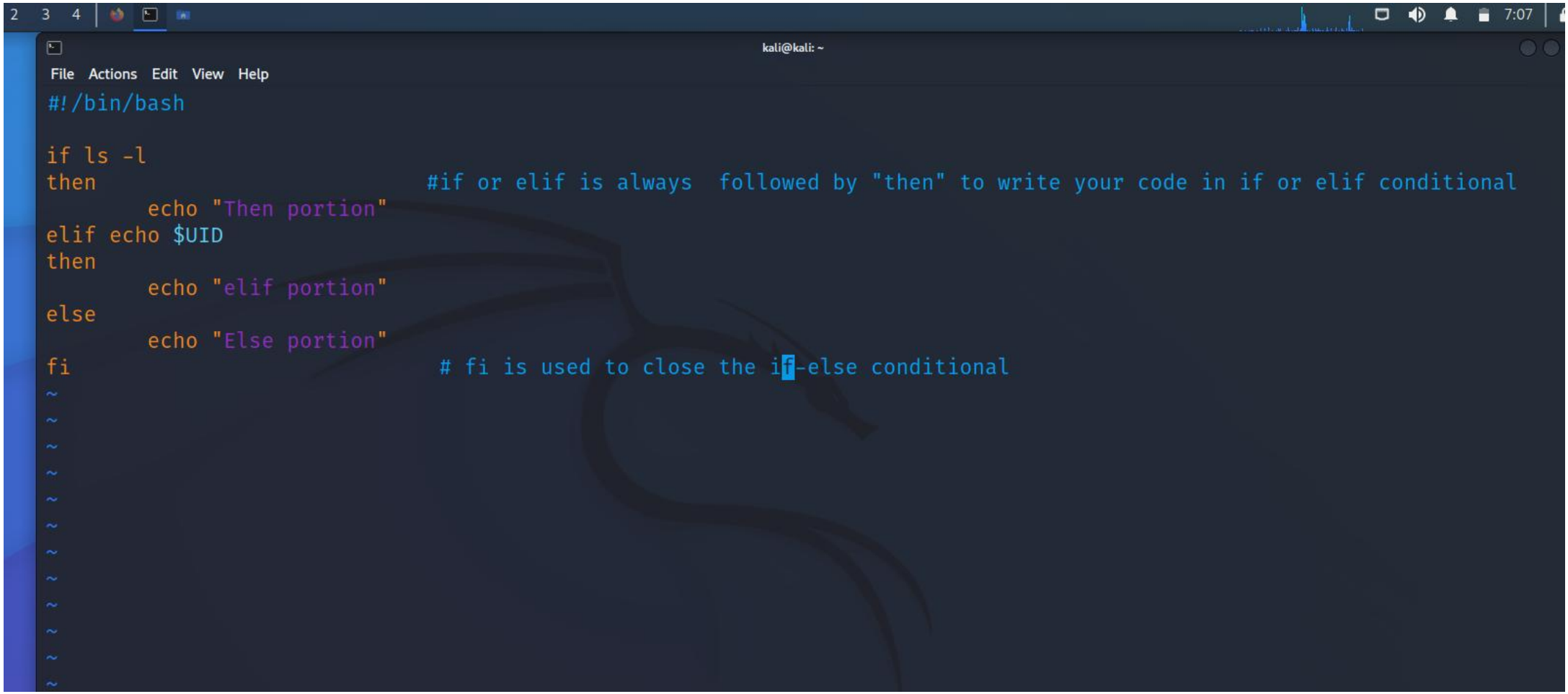
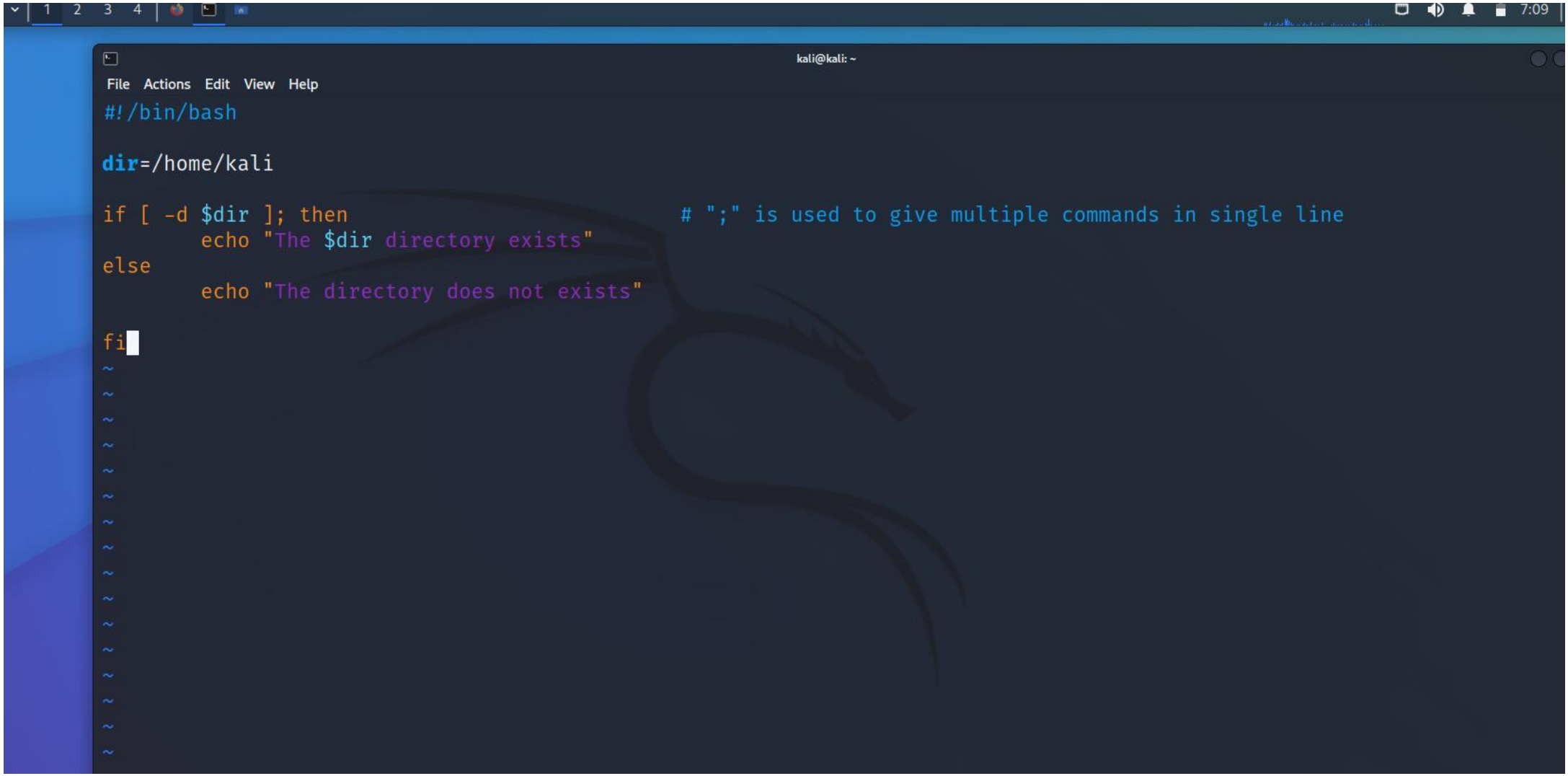
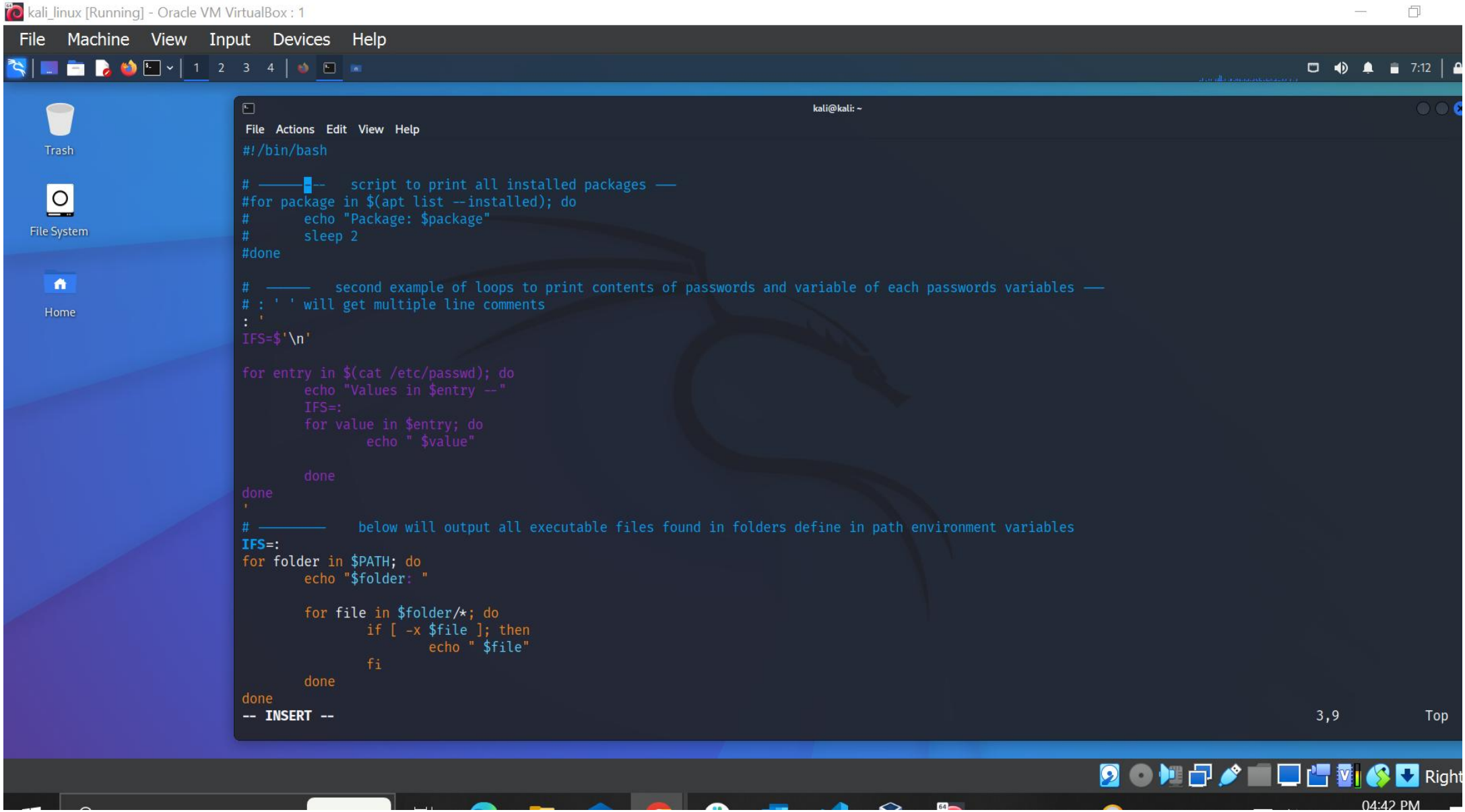
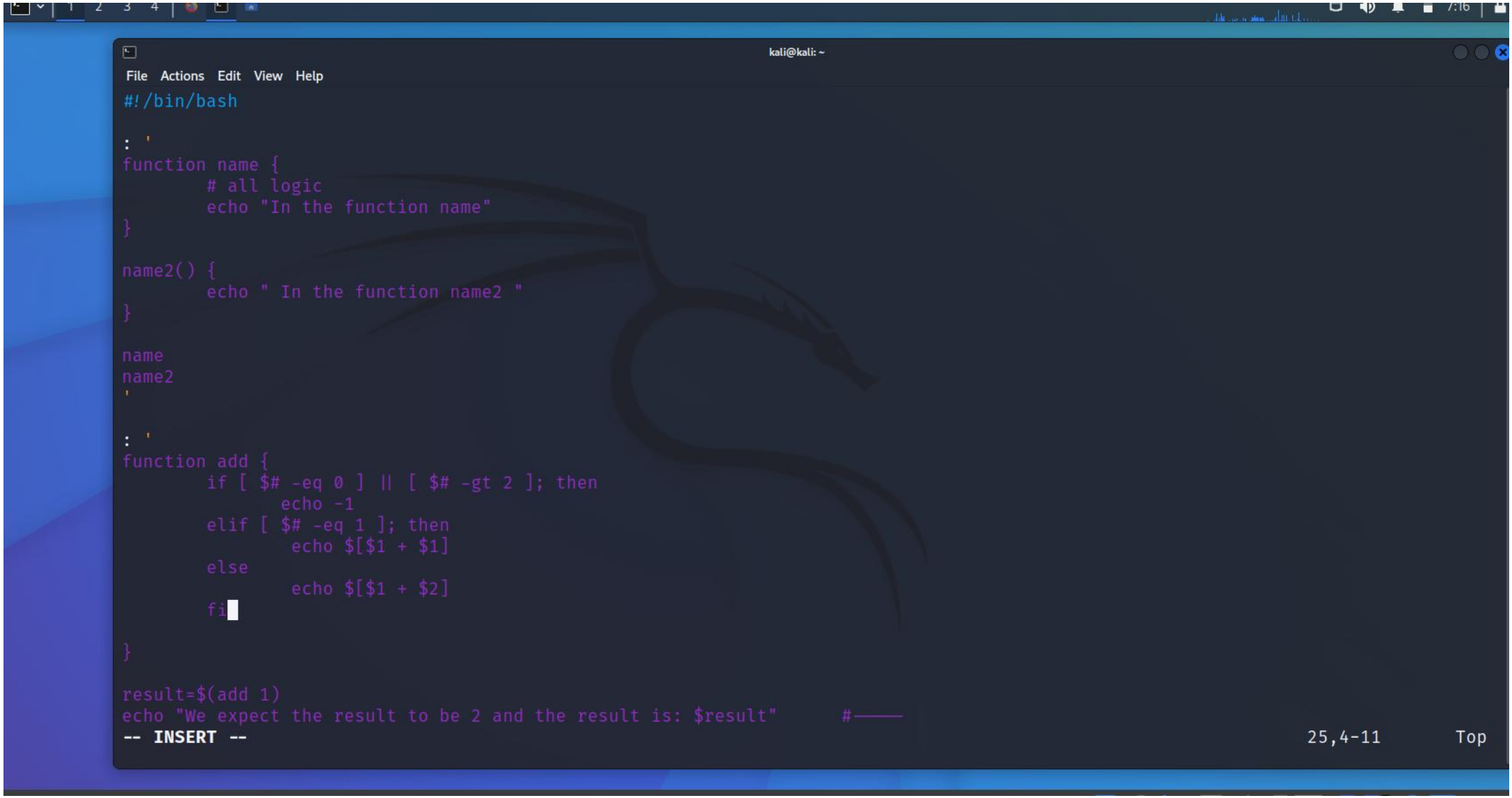
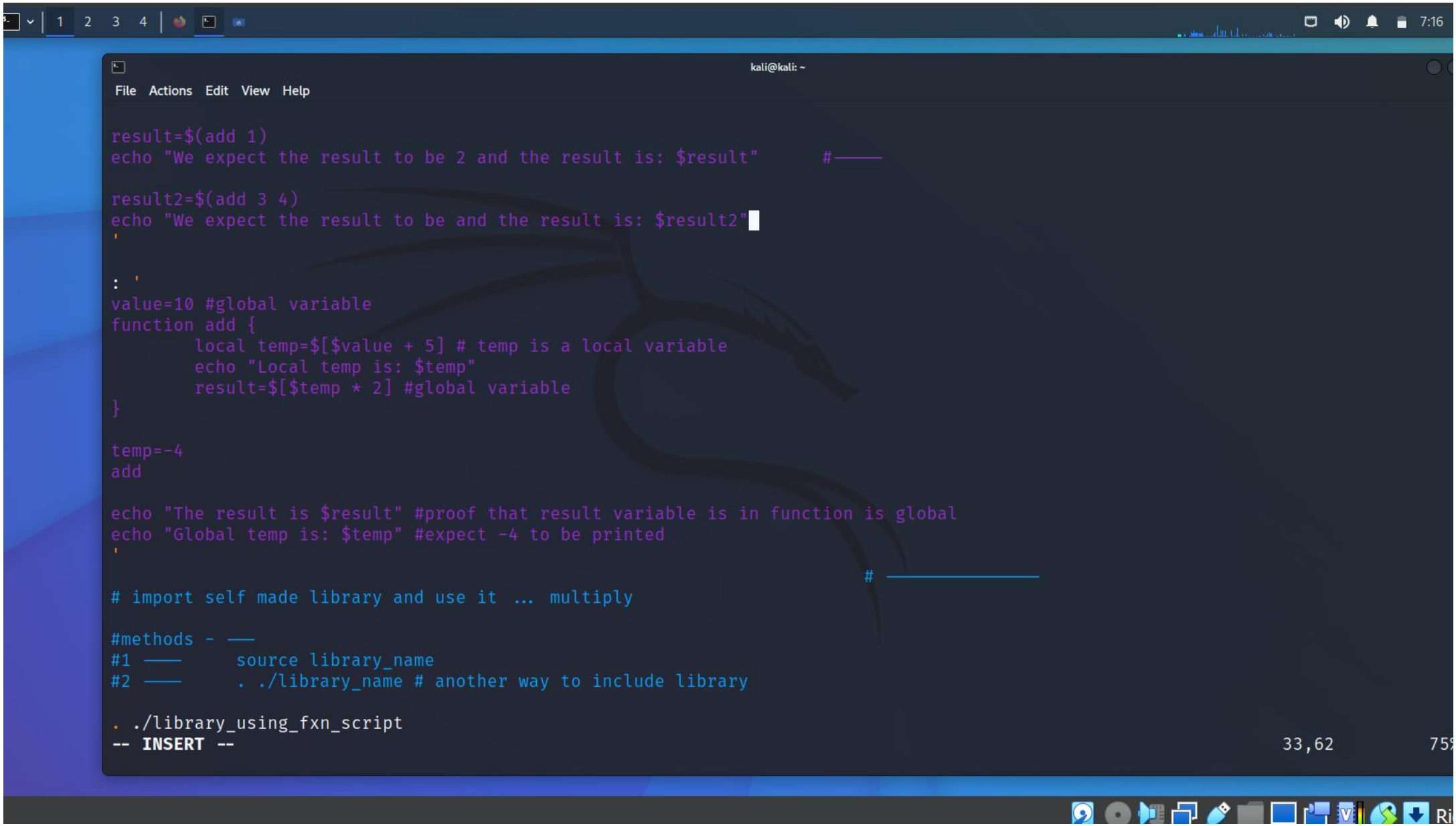

6:58
kali@kali: ~
File Actions Edit View Help
#!/bin/bash # first line tells which shell to use
first '#' is quite special which is followed by "!" to tell system to identify file as script
#this is a comment line, it will not be compiled
: '
This is multiline comment
you can write multiline comment by : ' ' command
'
echo "Hello World!" # echo is used to output on screen
echo UID: $UID # uid is user identifier
val1=1
val2="Text"
echo $val1; echo $val2
echo "The date is $(date)" # here date is keyword for accessing system date
today=$(date +%y%m%d)
echo "Today's date is $today"
-- INSERT --
24,1
All

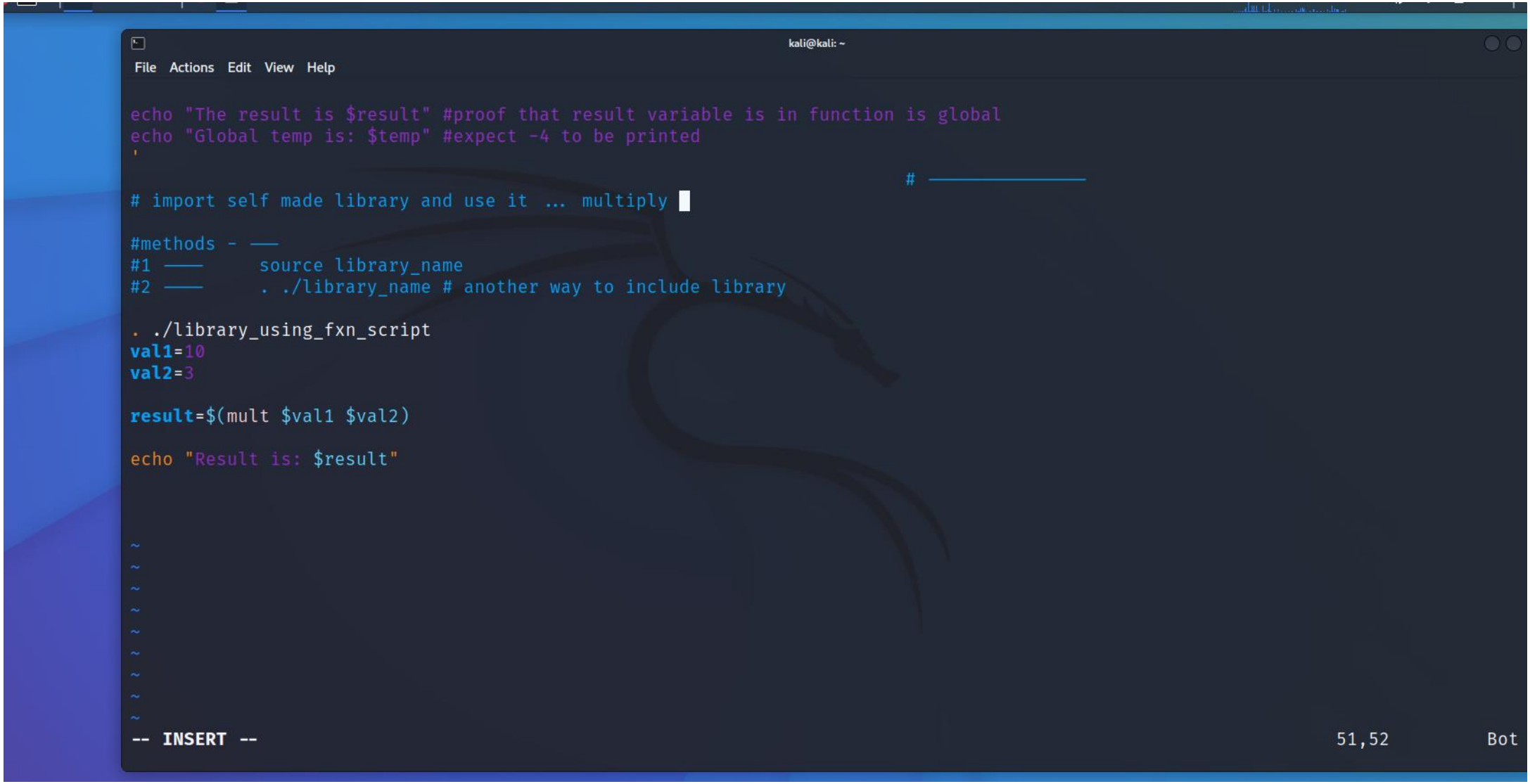

2 3 4
7:07
kali@kali: ~
File Actions Edit View Help
#!/bin/bash
if ls -l
then
#if or elif is always followed by "then" to write your code in if or elif conditional
echo "Then portion"
elif echo $UID
then
echo "elif portion"
else
echo "Else portion"
fi
fi is used to close the if-else conditional

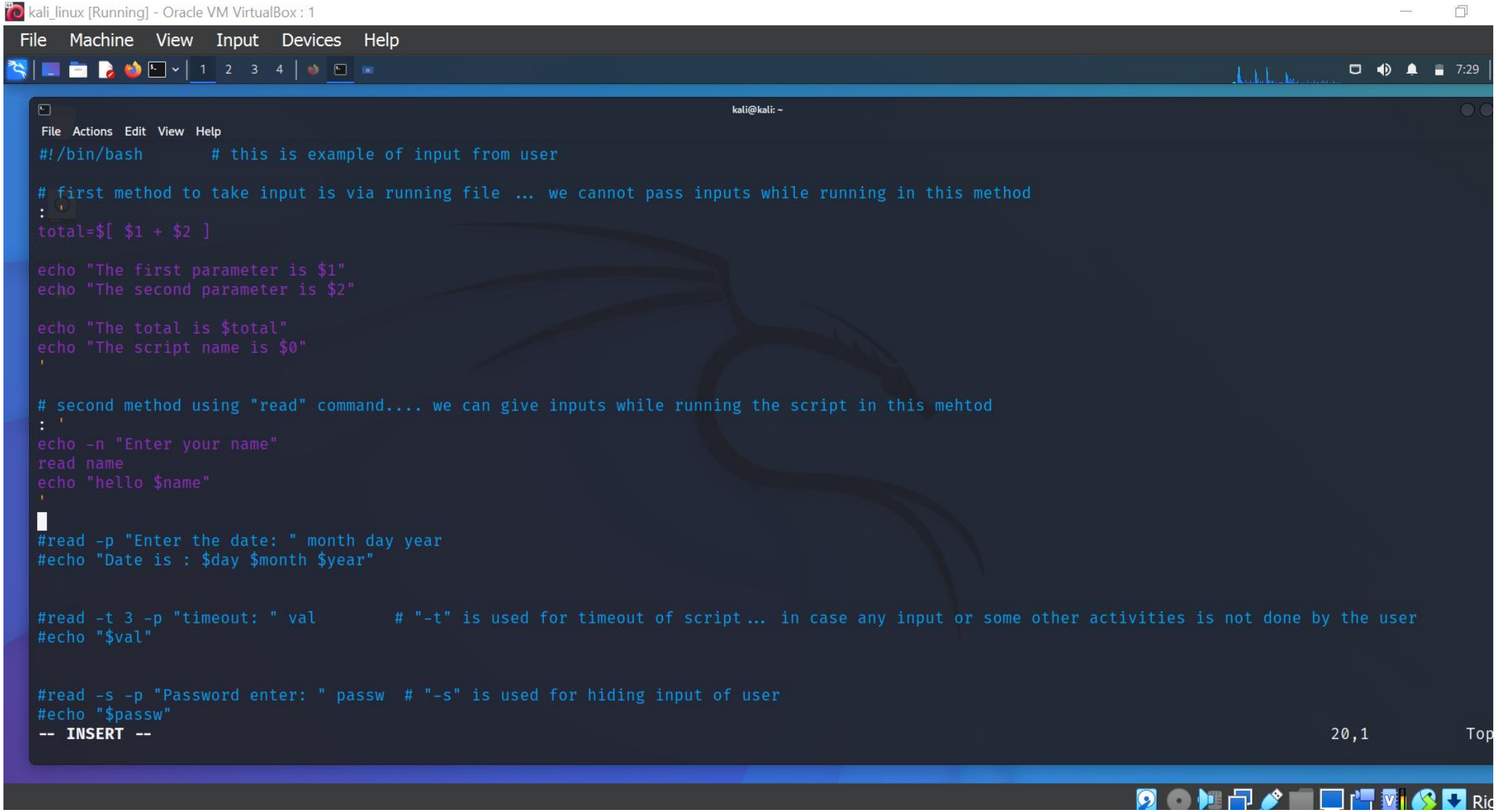

1 2 3 4
7:09
kali@kali: ~
File Actions Edit View Help
#!/bin/bash
dir=/home/kali
if [-d $dir]; then # ";" is used to give multiple commands in single line
echo "The $dir directory exists"
else
echo "The directory does not exists"
fi

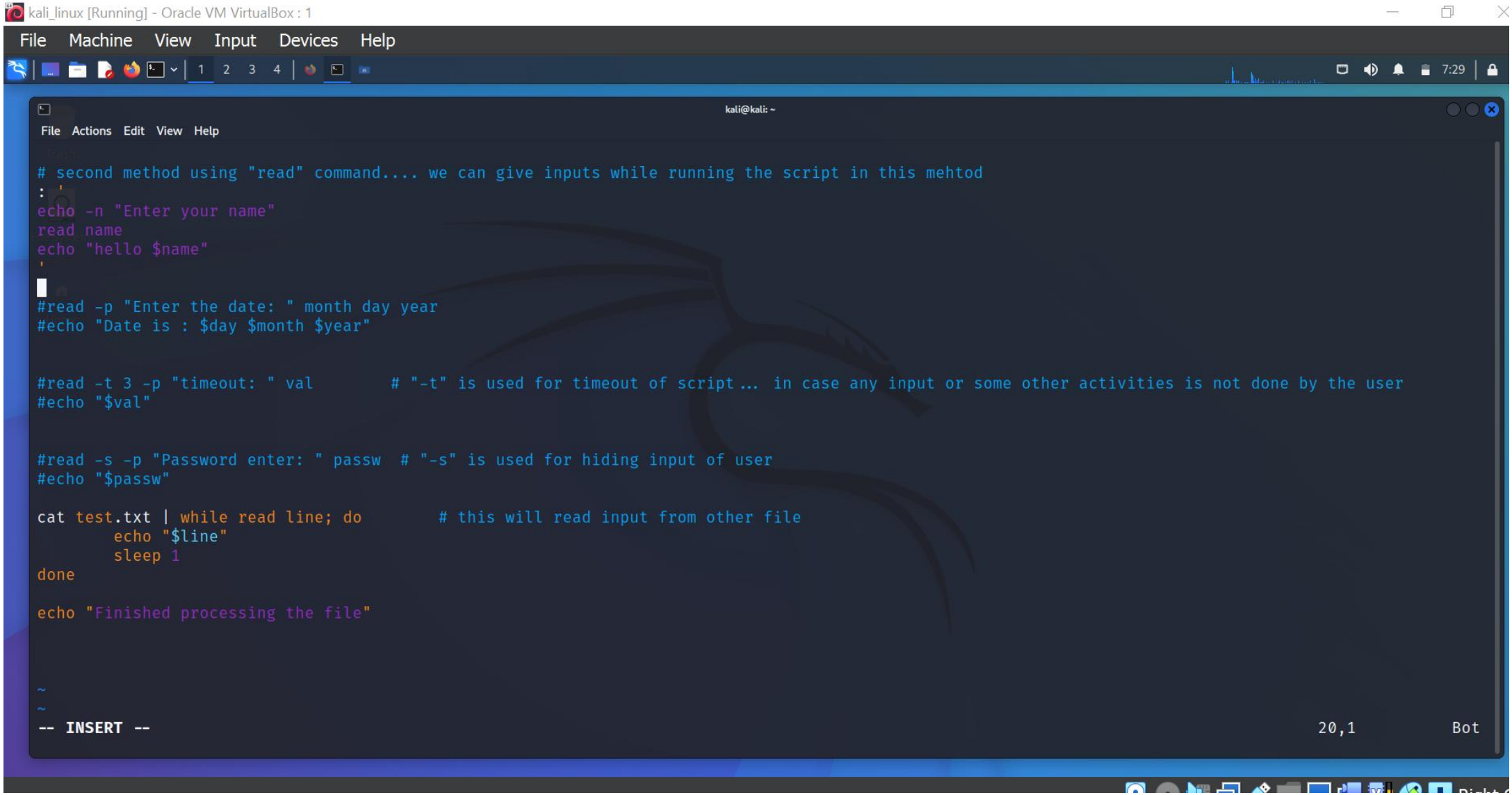

kali_linux [Running] - Oracle VM VirtualBox : 1
File Machine View Input Devices Help
1 2 3 4
7:12
Trash
File System
Home
kali@kali: ~
File Actions Edit View Help
#!/bin/bash
------- script to print all installed packages ----
#for package in $(apt list --installed); do
echo "Package: $package"
sleep 2
#done
------ second example of loops to print contents of passwords and variable of each passwords variables ----
: ' ' will get multiple line comments
: '
IFS=$'\n'
for entry in $(cat /etc/passwd); do
echo "Values in $entry --"
IFS=:
for value in $entry; do
echo " $value"
done
done
'
# ---------- below will output all executable files found in folders define in path environment variables
IFS=:
for folder in $PATH; do
echo "$folder: "
for file in $folder/*; do
if [-x $file]; then
echo " $file"
fi
done
done
-- INSERT --
3,9
Top
Right
04:42 PM


kali@kali: ~
File Actions Edit View Help
#!/bin/bash
: '
function name {
# all logic
echo "In the function name"
}
name2() {
echo " In the function name2 "
}
name
name2
'
: '
function add {
if [$# -eq 0] || [$# -gt 2]; then
echo -1
elif [$# -eq 1]; then
echo $[$1 + $1]
else
echo $[$1 + $2]
fi
}
result=$(add 1)
echo "We expect the result to be 2 and the result is: $result"
-- INSERT --
25,4-11
Top


1 2 3 4
7:16
kali@kali: ~
File Actions Edit View Help
result=$(add 1)
echo "We expect the result to be 2 and the result is: $result"
result2=$(add 3 4)
echo "We expect the result to be and the result is: $result2"
'
: '
value=10 #global variable
function add {
local temp=$[$value + 5] # temp is a local variable
echo "Local temp is: $temp"
result=$[$temp * 2] #global variable
}
temp=-4
add
echo "The result is $result" #proof that result variable is in function is global
echo "Global temp is: $temp" #expect -4 to be printed
'
import self made library and use it ... multiply
#methods - ——
#1 ——— source library_name
#2 ——— . ./library_name # another way to include library
. ./library_using_fxn_script
-- INSERT --
33,62
75%


kali@kali: ~
File Actions Edit View Help
echo "The result is $result" #proof that result variable is in function is global
echo "Global temp is: $temp" #expect -4 to be printed
'
import self made library and use it ... multiply
#methods - ——
#1 ——— source library_name
#2 ——— . ./library_name # another way to include library
. ./library_using_fxn_script
val1=10
val2=3
result=$(mult $val1 $val2)
echo "Result is: $result"
-- INSERT --
51,52
Bot


kali_linux [Running] - Oracle VM VirtualBox : 1
File Machine View Input Devices Help
1 2 3 4
7:29
kali@kali: ~
File Actions Edit View Help
#!/bin/bash # this is example of input from user
first method to take input is via running file ... we cannot pass inputs while running in this method
: '
total=$[$1 + $2]
echo "The first parameter is $1"
echo "The second parameter is $2"
echo "The total is $total"
echo "The script name is $0"
'
second method using "read" command.... we can give inputs while running the script in this mehtod
: '
echo -n "Enter your name"
read name
echo "hello $name"
'
#read -p "Enter the date: " month day year
#echo "Date is : $day $month $year"
#read -t 3 -p "timeout: " val # "-t" is used for timeout of script ... in case any input or some other activities is not done by the user
#echo "$val"
#read -s -p "Password enter: " passw # "-s" is used for hiding input of user
#echo "$passw"
-- INSERT --
20,1
Top


kali_linux [Running] - Oracle VM VirtualBox : 1
File Machine View Input Devices Help
1 2 3 4
7:29
kali@kali: ~
File Actions Edit View Help
second method using "read" command.... we can give inputs while running the script in this mehtod
: '
echo -n "Enter your name"
read name
echo "hello $name"
'
#read -p "Enter the date: " month day year
#echo "Date is : $day $month $year"
#read -t 3 -p "timeout: " val # "-t" is used for timeout of script ... in case any input or some other activities is not done by the user
#echo "$val"
#read -s -p "Password enter: " passw # "-s" is used for hiding input of user
#echo "$passw"
cat test.txt | while read line; do # this will read input from other file
echo "$line"
sleep 1
done
echo "Finished processing the file"
~
~
-- INSERT --
20,1 Bot