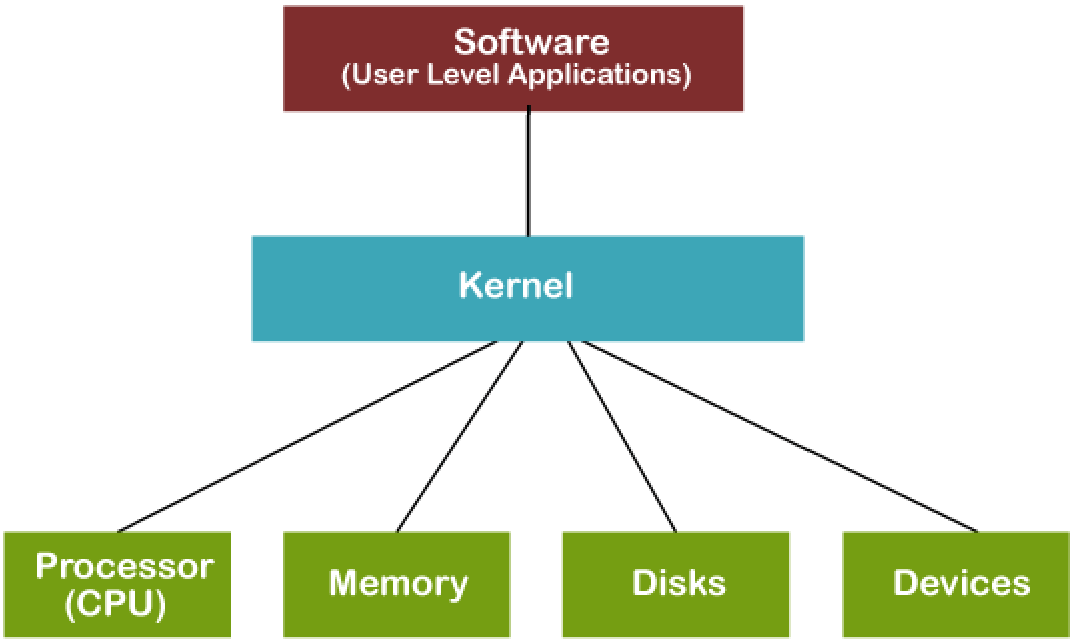
Day 4 Task: Basic Linux Shell Scripting for DevOps Engineers.

What is Kernel

Linux operating system contains a kernel Layer. A Kernel is the core component of any Linux based operating system. It is responsible for each of the major actions of the Linux OS. It virtualizes the common hardware resources of the computer to provide each process with its virtual resources. This makes the process seem as if it is the sole process running on the machine. The kernel is also responsible for preventing and mitigating conflicts between different processes.



There are some of the important kernel types which are mentioned below:

· Monolithic Kernel
· Hybrid kernels
· Exo kernels
· Micro kernels

What is Shell

Shell is an interface among the kernel and user which hides the complexity of the kernel's functions from the users. It takes commands from the user and executes the kernel's functions. These operating systems are categorized into two different types, which are the graphical shells and command-line shells.

The graphical line shells facilitate the graphical user interface, while the command line shells facilitate the command line interface. Thus, both of these shells implement operations. However, the graphical user interface shells work slower as compared to the command-line interface shells.

There are some types of these shells which are mentioned below:
· Korn shell
· Bourne shell
· C shell
· POSIX shell

What is Linux Shell Scripting for DevOps?

A shell script is a computer program designed to be run by a Linux shell, a command-line interpreter. The shell scripts are the command line text file which consists of commands in the order of execution. Typical operations performed by shell scripts include file manipulation, program execution, printing text and file saving, etc.

1. It enables us to develop our custom operating systems with custom features in it to suit our needs.
2. It helps in developing software applications as per our needs and concerning specific platforms.

What is #!/bin/bash? can we write #!/bin/sh as well?

This first line (#!/bin/bash or #!/bin/sh) has a name. This is known as shebang, it tells the interpreter that the following lines are written for bash, so execute this file as bash script.

So the shebang determines how to interpret the script, sometimes it could be as follow:

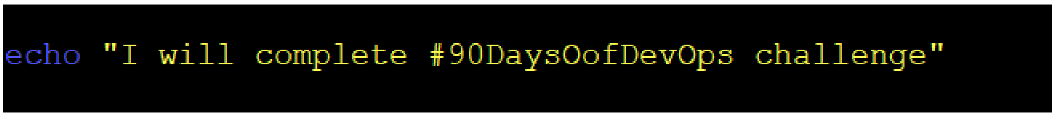
1. #!/bin/bash
2. #!/bin/sh
3. #!//usr/bin/perl
4. #!/usr/bin/python
5. #!/bin/ksh

    1. #!/bin/bash means the bash shell. It is the most common shell used as default shell for user login of the linux system. Bash can execute the vast majority of scripts and thus is widely used because it has more features, is well developed and better syntax.

    2. #!/bin/sh means the Bourne shell. It is an executable representing the system shell and usually implemented as a symbolic link pointing to the executable for whichever shell is the system shell. The system shell is basically the default shell that the script should use.

- Write a Shell Script which prints I will complete #90DaysofDevOps challenge

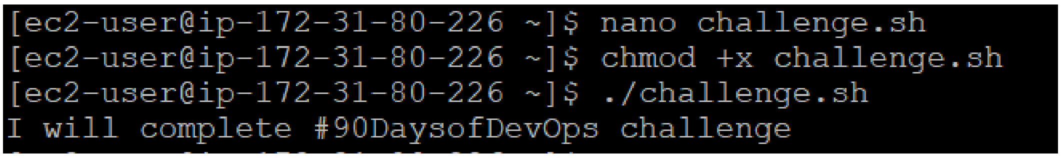
  Step 1 : Create a file with .sh using nano command (Eg: Challenge.sh) and write I will complete #90DaysOofDevOps challenge in echo.



```
echo "I will complete #90DaysOofDevOps challenge"
```

Step 2 : Give execute permission to the file using chmod command (eg : chmod +x Challenge.sh)

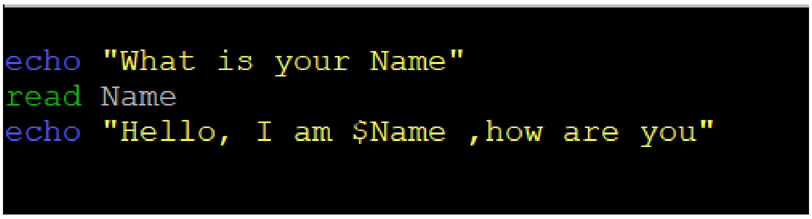
Step 3 : Now, run the file using ./ command (eg: ./challenge.sh)



```
[ec2-user@ip-172-31-80-226 ~]$ nano challenge.sh
[ec2-user@ip-172-31-80-226 ~]$ chmod +x challenge.sh
[ec2-user@ip-172-31-80-226 ~]$ ./challenge.sh
I will complete #90DaysofDevOps challenge
```

- Write a Shell Script to take user input, input from arguments and print the variables.

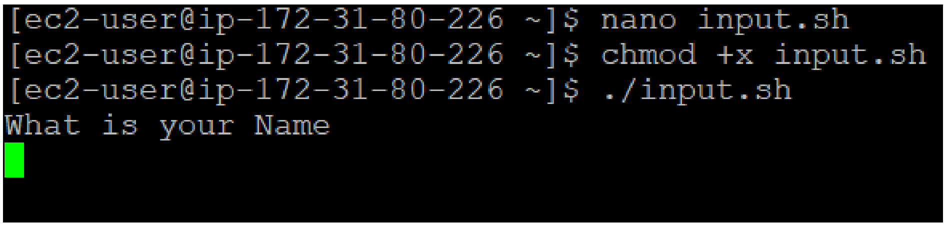
Step 1 : Create a file with .sh using nano command



```
echo "What is your Name"
read Name
echo "Hello, I am $Name ,how are you"
```

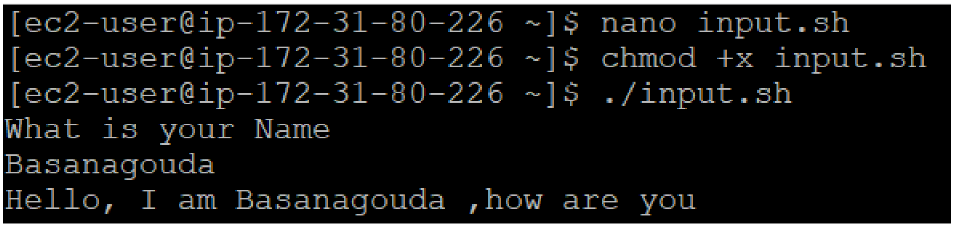
Step 2 : Give execute permission to the file using chmod command

Step 3 : Now, run the file using ./ command



```
[ec2-user@ip-172-31-80-226 ~]$ nano input.sh
[ec2-user@ip-172-31-80-226 ~]$ chmod +x input.sh
[ec2-user@ip-172-31-80-226 ~]$ ./input.sh
What is your Name
```

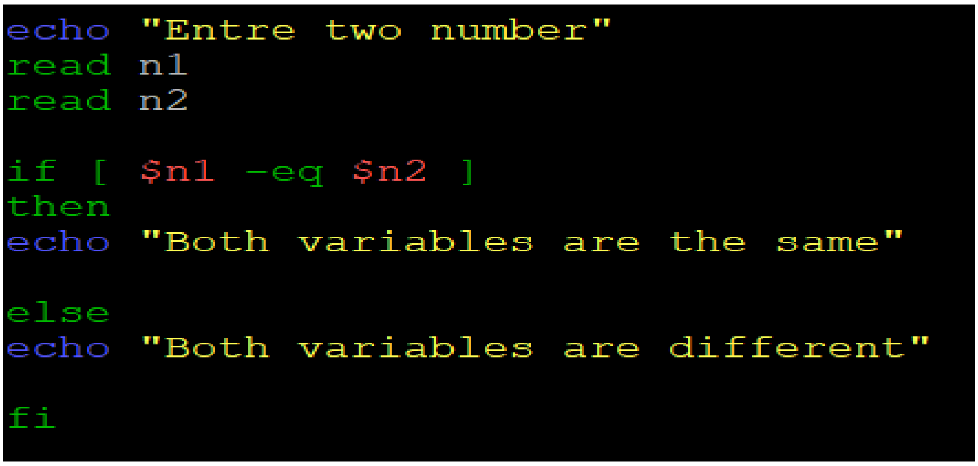
Cursor will ask to write your name ,after writing name press entre.



```
[ec2-user@ip-172-31-80-226 ~]$ nano input.sh
[ec2-user@ip-172-31-80-226 ~]$ chmod +x input.sh
[ec2-user@ip-172-31-80-226 ~]$ ./input.sh
What is your Name
Basanagouda
Hello, I am Basanagouda ,how are you
```

- Write an Example of If else in Shell Scripting by comparing 2 numbers was it difficult?

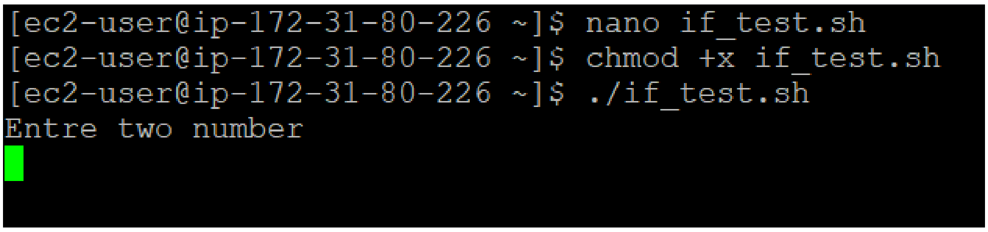
Step 1 : Create a file with .sh using nano command



```
echo "Entre two number"
read n1
read n2

if [ $n1 -eq $n2 ]
then
echo "Both variables are the same"

else
echo "Both variables are different"

fi
```

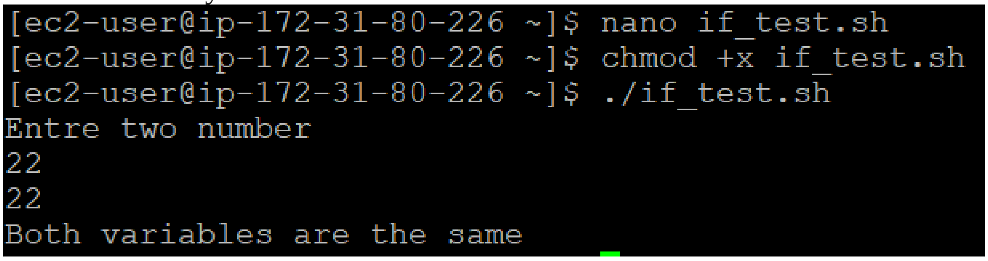
Step 2 : Give execute permission to the file using chmod command

Step 3 : Now, run the file using ./ command



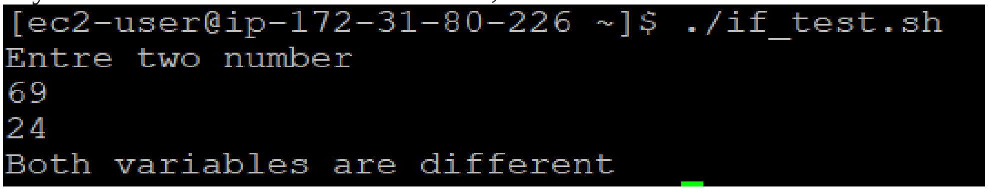
```
[ec2-user@ip-172-31-80-226 ~]$ nano if_test.sh
[ec2-user@ip-172-31-80-226 ~]$ chmod +x if_test.sh
[ec2-user@ip-172-31-80-226 ~]$ ./if_test.sh
Entre two number
```

Cursor will ask you to entre two number



```
[ec2-user@ip-172-31-80-226 ~]$ nano if_test.sh
[ec2-user@ip-172-31-80-226 ~]$ chmod +x if_test.sh
[ec2-user@ip-172-31-80-226 ~]$ ./if_test.sh
Entre two number
22
22
Both variables are the same
```

If you entre two different number then,



```
[ec2-user@ip-172-31-80-226 ~]$ ./if_test.sh
Entre two number
69
24
Both variables are different
```