Shell Scripting

→ Kernel is made up of Programming lan like c++ etc. In that even doing for small count like making a folder You have to write a code so all of the People cannot be able to code for that they developed shell Interface through which we can use Kernel to perform our tasks.

→ At the time shell was developed It's name was sh → Bourne shell → Basically a type of shell who interact with linux Kernel. Their command cd, pwd, echo

→ Then came ksh → Korn shell where You can write scripts for ex. cd, Pwd, echo You want to run line by line then You can execute it after writing shell.

→ Open-source Contributer thought that sh does diff thing, ksh does diff thing, Develop a new shell like writing script, command, automatic execute them came bash.

most popular

Bash → Bourne again shell,

csh → c shell (based on c Programming)

zsh

⇒ shell vs bash → their is no difference plus bash is a type of shell which become Popular

Script

Shell commands

⇒ Go to ssh connect www

Open cmd → Go to Downloads (where pem is downloaded)

ssh -i ... → yes

Shebang

Shell commands

echo "Raju Baby bhaiya aapki Kidney bech dete hai"
echo "Babu bhaiyaa: Utha le le baba"

cd mazedar
cd devops
mkdir scripts

→ This are individual commands & If you want to convert them in a script.
  vim Phir-hera-pheri-dialogue.sh

→ Interpreter → Interpret whatever is written in that shell & give it to shell so that shell can make kernel understand what they're doing.

→ A shell script has an orchestra which is written at top so we to write that we have a particular type of syntax called shebang.
  #! → Hashtag & exclamation mark
  called starting of shebang

#!/bin/bash
#!/bin/ksh → tell interpreter to use Korn shell.

```
#!/bin/bash
echo "Raju babu bhaiya aapki Kidney"
echo "Babu bhaiya: Khopdi tod sale ka"
echo "Shyam: Golmal hain bhai";
```

→ Goes line by line execution & #!/bin/bash shows which interpreter to use.
  Esc : wq enter

→ Now you have to run this shell through bash
bash phir-hera-pheri-dialogue.sh
: O/P:

vim phir-hera-pheri-dialogue.sh
echo "shyam: meri anuradha wait kari hai kyunki
aaj" && date

echo "shyam: meri anuradha wait kar rahi kyunki
aaj" && date | awk '{print $1,$2,$3}'
O/P of this goes here

echo "Babu bhaiya: mera dil aise dhak dhak ho raha h
echo "Raju: kya"
echo "Raju bhaiya: kyunki RAM is full:" && free -h
↳ free-h is used to show RAM usage
show whole usage.

→ ls -l phir-hera-pheri-dialogue.sh
not execute permission

→ ./phir-hera-pheri-dialogue.sh
./ and then write script then it becomes executab
shows permission denied.

chmod 700 phir-hera-pheri-dialogue.sh
before color was white but after this color
becomes green.
./shellscript name → execute.

→ If I want to Print Something as variables

vim variables.sh

→ #!/bin/bash → Here I want to make a script

name= mayank for a name of variable

echo "My name is $name" → $ indicates variable you can access with $ sign → as variable

→ name= mayank Patel mayank Patel ✗ no space when echo

→ Anything that can hold a value which can be changed is called variable.

→ If I want to take a variable from use then

echo "My what is your channel name?"

read channel

echo "Do subscribe $channel!"

Script 3 vim arguments.sh

shebang

#!/bin/bash → echo "This is $1" → chmod 700 arguments.sh

→ when you write a shell script after which what you write becomes argument.

→ If you want to access argument you can access it through $1

./arguments.sh → This is

./arguments.sh awesome → This is awesome

Page No.
Date: / /

→ whatever you after this file you can enter it in shell script then you can do like with below
vim install_pkg_by_args.sh
#!/bin/bash
sudo apt install $1 → This script let you install anything through argum
⇒ esc : wq enter
chmod 700 install_pkg_by_args.sh

→ ./install_pkg_by_args.sh docker

interview
I want to calculate how many arguments are there
echo "Total characters are: $#"
$# gives you count of all the arguments.

Q I want to Print every argument then : $@

Q I want to install nginix and run a small website
vim install_nginix_and_serve_webpage.sh
#!/bin/bash
commend # update system
sudo apt-get update -y (Do you want to update type question)
# install nginx
sudo apt-get install nginx -y
# start and Enable Nginx
sudo systemctl start nginx
sudo systemctl enable nginx (means next time system starts make it enable start by default)