

I ❤️ #!/bin/bash



Shell Scripting - Module 7

Regular Expressions

Awk

What can we do with AWK?

AWK in different scenarios

SED

Print Commands :

Editing a file

File modification

Regular Expressions

Awk

- Awk is used for manipulating data and generating reports.

What can we do with AWK?

- Scans a file line by line
- Splits each input line into fields
- Compares input line/fields to pattern

- **Performs action(s) on matched lines**

AWK in different scenarios

We shall see how to check `df -h` results in different ways using `awk`.

- **Print First column from output `df -h`**

```
df -h | awk {'print $1'}
```

- **Print only Row 2, column 1 from output.**

```
df -h | awk 'NR==2{print $1}'
```

- **Print Row 2, Column 1 & 5**

```
df -Ph | awk 'NR==2{print $1, $5}'
```

- **Print Column 1, Characters starting from 1 to 12 in output.**

```
df -h | awk -F'|' '{print substr($1,1,12)}'
```

- **Print Row 2, Column 1, Characters starting from 1 to 12 in output.**

```
df -h | awk -F'|' 'NR==2{print substr($1,1,12)}'
```

- **Find Present working used space**

```
df -h -P $PWD | awk {'print $3'}
```

Example code :

```
#!/bin/bash
```

```
#Variable declaration
echo -n "Please enter the IP: "
read IP

#The below step will do the ping of IP
count=`ping -c2 $IP | grep 'received' |awk -F',' '{print $2}' |awk '{print $1}'`

#Loop
if [ $count -eq 2 ];then
echo -e "Reachable : Server is up ${NC}"
else
echo -e "Reachable : Server is down ${NC}"
fi
```

SED

SED command in UNIX stands for **stream editor** and it can perform lot's of function on file like, searching, find and replace, insertion or deletion.

Print Commands :

- **Print a file using SED**

```
sed 'p' /etc/passwd
```

- **Print a file without duplications**

```
sed -n 'p' /etc/passwd
```

- **Print first 5 lines of your file**

```
sed -n '1,5 p' /etc/passwd
```

- **Print 25th line of your file**

```
sed -n '25 p' /etc/passwd
```

- **Print line from 20 to 25**

```
sed -n '20,25 p' /etc/passwd
```

- **Search and Prints for lines starting with word "user" in file.**
- **Search word `/^user/` should be limited with delimiter as per sed syntax**

```
sed -n '/^user/ p' /etc/group
```

Editing a file

- **Use "d" to delete lines from file**
- **Delete lines starting with "#" from file.**

```
sed '/^#/ d' sample.sh
```

- **Deleting first two lines from file**

```
sed '1,2 d' sample.sh
```

- **We are doing 3 operations with this command, all are seperated using ;**
- **Deleting lines starting with #**
- **Deleting empty lines by mentioning \$**
- **Deleting first 2 lines**

```
sed '/^#/ d; /^$/ d; 1,2 d' sample.sh
```

- **Whatever we are doing will not get implemented until we add "-i" with sed**

```
sed -i '/^#/ d' sample.sh
```

File modification

- **Finding a line starting with ^echo in config file and using 'a' - append to insert new line after that line.**

```
sed '/^read/ a new line' sample.sh  
  
# implement  
sed -i '/^read/ a new line' sample.sh
```

- **Insert line into a file using 'i'**

```
sed '/^read/ i new line' sample.sh  
  
# implement  
sed -i '/^read/ i new line' sample.sh
```