**SLL Question Bank Answers for April 2024**

**MAKE SURE TO DO THIS!!**

**First super user by typing 'su' then enter password as sll731**

**1. Configure Apache Web Server.**

**yum install httpd**

y then enter if package not installed

**systemctl start httpd**

**gedit /var/www/html/index.html**

Do some basic html coding:

<!DOCTYPE html>

<html>

<head>

<title> Page Title</title>

</head>

<body>

<h1> This is H1 Heading</h1>

<p>This is a paragraph</p>

</body>

</html>

Make sure to save the file.

Go to terminal and type

**ip a**

and enter the ip in firefox address bar you got from the command:

It usually starts with **192.168.100.////**

***2. a) Create a text file. Infer the file permissions.***

**touch file.txt**

**ls -l file.txt**

***b) Using the Octal mode change the permission on a particular file as rw-rw-r–***

**chmod 664 file.txt**

**ls -l file.txt**

***c) Using the Symbolic mode change the permission on a particular file as rw-rw-r***

**chmod u=rw,g=rw,o=r file.txt**

**ls -l file.txt**

***3. Configure SSH server.***

Run the following commands on Server and Client.

***(Server is where you will access the files and Client is from where you will connect to the server to access the files)***

**Server:**

**yum install openssh-server**

**systemctl start sshd**

**systemctl status sshd**

**systemctl stop firewalld**

You can create a file or folder that the client can access in the server using **touch**

Get ip of the server pc by doing **ip a**

It usually starts with **192.168.100.///**

**Client:**

**ssh sfit@IP**

**sll731**

Now you can run various commands and check the directory with **ls** and **cd**

***4. a) Create a user by assigning the primary group explicitly.***

**groupadd mygroup**

**useradd username**

**usermod -g groupname username**

To check added user

**cat /etc/passwd**

***b) Demonstrate the account status using the password aging settings and infer on the various account status.***

**passwd -S username**

***c) Demonstrate locking and unlocking of the user password***

**passwd -l username**

**passwd -u username**

OR

**passwd -u username -f**

***5. Configure NFS server-client.***

Run the following commands on Server and Client.

***(Server is where you will access the files and Client is from where you will connect to the server to access the files)***

**yum install nfs-utils libnfsidmap**

**systemctl enable rpcbind nfs-server**

**systemctl start rpcbind rpc-statd nfs-server nfs-idmapd**

**systemctl stop firewalld**

**Server:**

Create a directory for demonstration:

**mkdir foldername**

**chmod a+rwx foldername**

**ls**

**gedit /etc/exports**

write following line in file:

**foldername client\_ip(rw,sync,no\_root\_squash)**

Save the file

**exportfs -rv**

**Client:**

**mkdir /mnt/directory\_name**

**mount server\_ip:/foldername /mnt/directory\_name**

Make sure to enter the **foldername** same as you created in the server

**df -h**

To unmount the directory

**umount /mnt/directory\_name**

***6. a) Write a shell script to reverse the word order in a list of strings. For example, if the input is Hello World, output should be World Hello.***

**gedit 6a.sh**

#!/bin/bash

read -p "Enter a list of strings separated by space: " input\_string

reversed\_string=""

for word in $input\_string; do

reversed\_string="$word $reversed\_string"

done

echo "Reversed word order: $reversed\_string"

**bash 6a.sh**

***b) Write a shell script to convert the user-given temperature in Celsius to Fahrenheit using a bash calculator***

**gedit 6b.sh**

#! /bin/bash

read -p "Enter temperature in Celcius: " celsius

farheneit=$(echo "scale=2; ($celsius \*9/5) + 32" | bc)

echo "$celsius in Farheneit is $farheneit."

**bash 6b.sh**

***7. Establish Telnet communication on port 43897.***

Run the following commands on Server and Client.

***(Server is where you will access the files and Client is from where you will connect to the server to access the files)***

**yum install telnet telnet-server**

**systemctl start telnet.socket**

**systemctl status telnet.socket**

**systemctl stop firewalld**

**semanage port -a -t telnetd\_port\_t -p tcp 43897** #Incase you get some error run this first

On server side edit the following file

**gedit /usr/lib/systemd/system/telnet.socket**

Change the ListenStream value to number **43897**

**systemctl daemon-reload**

**systemctl restart telnet.socket**

Find **IP** with **ip a**

It usually starts with **192.168.100.///**

On client side:

**telnet IP**

**telnet IP 43897**

***8. Perform the following operations using yum package manager and interpret the output of command used: a) List all installed packages***

**yum list installed**

***b) Describe the info of a package***

**yum info package\_name**

***c) Find out which package installed a particular file***

**yum deplist package\_name**

***9. Demonstrate fdisk and df commands and interpret it’s output***

***FDISK:-***

1. View all Disk Partitions in Linux

**fdisk -l**

2. View Specific Disk Partition in Linux

**fdisk -l /dev/sda**

3. Check all Available fdisk Commands

**fdisk /dev/sda**

***DF:-***

1. Display all the file system

**df -a**

2. Show the file system type

**df -T**

3. Show the disc space usage in a readable format

**df -h**

***10. Configure Telnet server-client.***

Run the following commands on Server and Client.

***(Server is where you will access the files and Client is from where you will connect to the server to access the files)***

**yum install telnet telnet-server**

**systemctl start telnet.socket**

**systemctl status telnet.socket**

**systemctl stop firewalld**

**semanage port -a -t telnetd\_port\_t -p tcp 4000** #Incase you get some error run this first

On server side edit the following file

**gedit /usr/lib/systemd/system/telnet.socket**

Change the ListenStream value to number **4000**

**systemctl daemon-reload**

**systemctl restart telnet.socket**

Find **IP** with **ip a**

It usually starts with **192.168.100.///**

On client side:

**telnet IP**

**telnet IP 4000**

***11. a) Write a shell script that consists of a function that displays the number of files in the present working directory. Name this function “file\_count” and call it in your script.***

**gedit 11a.sh**

#!/bin/bash

file\_count() {

local count=$(ls -1 | wc -l)

echo "Number of files in the current directory: $count"

}

file\_count

**bash 11a.sh**

***b) Write a shell script to terminate the script if invoked by non-root user using function and appropriate exit codes.***

**gedit 11b.sh**

#!/bin/bash

check\_root() {

if [ "$(id -u)" != "0" ]; then

echo "Error: This script must be run as root."

exit 1

fi

}

check\_root

echo "Script executed successfully by root user."

**bash 11b.sh**

***12. Write a shell script to check the system status.***

**gedit 12.sh**

#!/bin/bash

machine()

{

echo -e "\e[29;44m\*\*\*\*\*\* HOSTNAME INFORMATION \*\*\*\*\*\e[0m"

hostnamectl

echo " "

echo -e "\e[29;44m\*\*\*\*\*\* FILE SYSTEM DISK SPACE USAGE \*\*\*\*\*\e[0m"

df -h

echo " "

echo -e "\e[29;44m\*\*\*\*\*\* FREE & USED MEMORY \*\*\*\*\*\e[0m"

free

echo " "

echo -e "\e[1;32m\*\*\*\*\*\* SYSTEM UPTIME $ LOAD \*\*\*\*\*\e[0m"

uptime

echo " "

echo -e "\e[29;44m\*\*\*\*\*\* CURRENTLY LOGGED-IN USERS \*\*\*\*\*\e[0m"

who

echo " "

echo -e "\e[29;44m\*\*\*\*\*\* TOP 5 MEMORY-CONSUMING PROCESSES \*\*\*\*\*\e[0m"

ps -eo %mem,%cpu,comm --sort=-%mem | head -n 6

echo " "

echo -e "\e[1;32Done.\e[0m"

}

machine

**bash 12.sh**

***13. Configure FTP server.***

**yum install vsftpd**

**systemctl start vsftpd**

**systemctl status vsftpd**

**gedit /etc/vsftpd/vsftpd.conf**

Make following changes in file:

• Change

**anonymous\_enable=YES to anonymous\_enable=NO**

• uncomment

**ascii\_upload\_enable=YES & ascii\_Ddownload\_enable=YES**

• uncomment

**Welcome to blah ftp service**

• add at the end of file

**use\_localtime=YES**

**systemctl start vsftpd**

**systemctl enable vsftpd**

**systemctl stop firewalld**

To connect to server from client side:

**yum install ftp**

Find **IP** with **ip a**

It usually starts with **192.168.100.///**

**ftp IP**

Enter username & password and do the following:

**bin** ( To enter into binary mode )

**hash** ( To see # progress of transfer )

**put file\_name** ( upload file from clients home directory )

**get file\_name** ( download file )

**mget file\_1 file\_2** ( download multiple files )

**bye** ( close connection )

***14. Write a shell script to automate the following a) Adding a user***

***b) Changing the group of the user***

**gedit 14.sh**

#!/bin/bash

add\_user() {

read -p "Enter username: " username

read -p : "Enter fullname: " userfullname

echo ""

read -p "Enter group name: " groupname

groupadd $groupname

useradd -g $groupname -c "$userfullname" $username

echo "User $username added successfully to group $groupname."

}

change\_group() {

read -p "Enter username: (that already exists!) " username

read -p "Enter new group name: " new\_groupname

groupadd $new\_groupname

sudo usermod -g $new\_groupname $username

echo "Group of user $username changed to $new\_groupname."

}

echo "Select an option:"

echo "1. Add a user"

echo "2. Change group of a user"

read -p "Enter your choice: " choice

case $choice in

1) add\_user ;;

2) change\_group ;;

\*) echo "Invalid choice. Please select 1 or 2." ;;

esac

***15. Establish SSH communication on port 41101.***

Run the following commands on Server and Client.

***(Server is where you will access the files and Client is from where you will connect to the server to access the files)***

**Server:**

**yum install openssh-server**

**systemctl start sshd**

**systemctl status sshd**

**systemctl stop firewalld**

Create a file or folder that the client can access in the server using **touch**

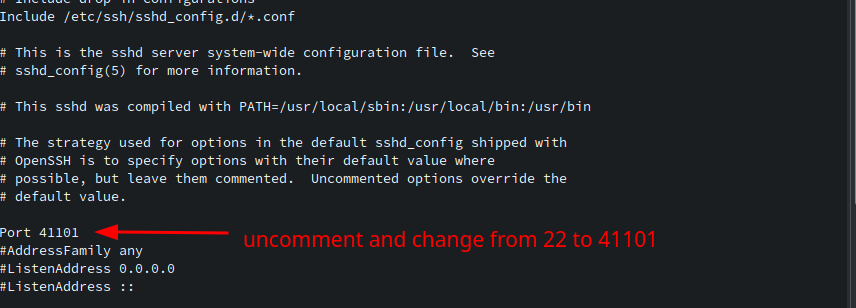
Get ip of the server pc by doing **ip a**

It usually starts with **192.168.100.///**

To configure port to 41101 type

**gedit /etc/ssh/sshd\_config**

And uncomment the Port and type 41101.



**systemctl restart sshd**

**Client:**

**ssh sfit@IP**

**sll731**

Now you can run various commands and check the directory with **ls** and **cd**

For 41101 port:

**ssh sfit@IP -p 41101**

***16. Write a shell script to automate the following***

***a. Installing a package using yum***

***b. Checking package info using yum***

**gedit 16.sh**

#!/bin/bash

install\_package() {

read -p "Enter the name of the package to install: " package\_name

sudo yum install $package\_name -y

echo "Package $package\_name installed successfully."

}

check\_package\_info() {

read -p "Enter the name of the package to check info: " package\_name

sudo yum info $package\_name

}

echo "Select an option:"

echo "1. Install a package using yum"

echo "2. Check package info using yum"

read -p "Enter your choice: " choice

case $choice in

1) install\_package ;;

2) check\_package\_info ;;

\*) echo "Invalid choice. Please select 1 or 2." ;;

esac

**bash 16.sh**

***17. Configure NFS server-client.***

Run the following commands on Server and Client.

***(Server is where you will access the files and Client is from where you will connect to the server to access the files)***

**yum install nfs-utils libnfsidmap**

**systemctl enable rpcbind nfs-server**

**systemctl start rpcbind rpc-statd nfs-server nfs-idmapd**

**systemctl stop firewalld**

**Server:**

Create a directory for demonstration:

**mkdir foldername**

**chmod a+rwx foldername**

**ls**

**gedit /etc/exports**

write following line in file:

**/foldername client\_ip(rw,sync,no\_root\_squash)**

Save the file

**exportfs -rv**

**Client:**

**mkdir /mnt/directory\_name**

**mount server\_ip:/foldername /mnt/directory\_name**

Make sure to enter the **foldername** same as you created in the server

**df -h**

To unmount the directory

**umount /mnt/directory\_name**

***18. Write a shell script to check if the user is root.***

**gedit 18.sh**

#!/bin/bash

# Check if the user is root

if [ "$(id -u)" = "0" ]; then

echo "User is root."

else

echo "User is not root."

fi

**bash 18.sh**