30/02/22

LSS

Assignment - 2

1) Discuss programming constructs.

=> 1) if - then if Statement

```
if [test_commond]
then
Commond - block.
fi.
```

2) if Statement

```
if [expression]
then
    Statement
fi
```

3) if-else Statement

```
if [expression]
then
    Statement 1
else
    Statement 2
fi
```

4) Switch - case

```
case in
  Pattern 1) Statement 1;;
  Pattern n) Statement n;;
```

5) logical operators.

-a (read as AND)

-o (read as OR)

! (read as NOT).

c) Looping Control Structure

1) While loop

It is used when we want to repread something. Checks for the conditions if the condition is true it enters the loop & loop continous until the condition becomes false.

```
while [test_condition]
do
        Command - block.

done
```

~~* Utility Statement~~

2) Utility Statement

```
Utility controll -Commond
do
        commond - block.
done
```

come back is executed until the control command remains false.

3) for Statement

```
for   control - variable   is   value1 value2 ---

do
      Commond -block
done
```

- control variable take's the value value1 from 1st action and so on.
- Number of times the loop gets busted depends on the number of value in the list spfited after keyword in.

2) wild Card Character.

i) * (Astrick)

- An astrick is replaced by any number of characters in a file name.
- It can be used either at beginning/end of a pattern or' at both the end of the pattern or in between. the pattern:

~~the pattern~~

Ex:- ac1 ac2 ac3 } files in same directory.

then we can use ac* to list out the files.

ii) ? (Question mark)

- it is replaced by any single character.
- ? is a Meta character.
- ? cannot be replaced with /

```
$ LS
$ LS ?st* → Match with one character.
$ LS ??st* → Matchs with two character.
```

3) scope of any variable in the region from which it can be accessed or over which it is defined. Ay

Global variable

local variable

- globally scoped variable defined in terminal can be accessed from ~~e~~ anywhere in that particular environment

Locally scoped

locally scoped variable is defined in terminal cannot be accessed by any program or process running in the terminal. It can only be assured by the terminal.

31/03/2022 LSS

Assignment -03

=> Process Management and user Management.

A process Management.

A process means program in execution. It generally takes an input, processes it and gives us the appropriate output. ~~Check~~ .

1) Foreground process: Such kind of processes are know as interactive processes. These are the processes which are to be executed or initiated by the user or the programmer, they can not be initialized by the system services.

2) Back ground process: Such kind of processes. are also know as non interactive processes. These are the processes that are to be exectued or initiated by the system itself or by user, through they can even be managed by users.

=> Examples of foreground process.

-> Sleep 5

These command will be executed in the terminal and we would be able to execute another command after the execution of the above command.

-> Jobs

To get the list of jobs that are either running or stooped.

-> CTRL + Z

will stop the execution of the command.

→ bg.

To run all the pending and forces stopped jobs in the background.

→ ps -ef | grep sleep

To get details of a process running in background.

→ fg

To run all the pending and force stopped jobs in the foreground.

→ nohup sleep 100 &

To run a process in the background without getting impacted by the closing of the terminals.

→ Sleep 100&

To run some process in the background directly.

→ nice -n 5 sleep 100

To run processes without priority.

→ top

To get the list of all running processes on your linux Machine.

User Management.

A user is an entity, in a linux operating system, that can Manipulate files and perform several other operations. Each user is assigned an ID that is unique for each user in the operating system. .

```
awk -F ':' '{print $1}' /etc/passwd.
```

To list out all the users in linux.
Here we are accessing a file and printing only first column. .

```
id username
```

Using id command, you can get the ID of any username.

```
sudo useradd username
```

The command to add a user. useradd command adds new user to the directory.

```
passwd username
```

The command is to assign a password to a user.

```
cat /etc/passwd.
```

This commands prints the data of the configuration file. This file contains information about the user in the format.

→ Usermod -u new_id username.

The command to change the user ID for a user.

→ Usermod -g new-group-id username

This command can change the group ID of a user and hence it can be used to move a user to an already existing group.

→ Sudo usermod -l new_login_name old_login_name

This command is used to change the user login name.

→ Usermod -d new_home_directoly_path username

This command is used to change the home directory. of the user whose username is given and sets the new home directory as the directoly whose path is provided.

→ userdel -r username

This command is used to delete a username.