**Linux Shell Scripting: A project based approach to learning**

11 May 2021

01:00

THIRD SCRIPT:

1. Using not true in an if condition
2. To exit a program using the exit function
3. Taking input from the users (running the script in interactive mode) using read shell built in or command line arguments

* DRY - Don’t repeat yourself
* We can supply the exit status after the exit command
  + Command Exit [Exit status] --> exit 1
  + By default or by convention, a script when executes successfully gives a status as 0 otherwise, if it does execute properly, it exits with a non-zero status code
  + Exit is a shell bulti-in
  + Exit status is optional, if you don’t specify the exit status then, the command exit is the last command executed.
* Help test: give info on tests such as expressions, operators and string operations. Eg: -lt, -ne, -gt, -ge etc.
* ? - it is a special variable that is used to store the exit status of the most recently executed command.
* You can encapsulate the ? Variable like -> ${?}
* For instance, if USERNAME=$(id -un) run successfully then USERNAME is equal to 0 otherwise, it is any non-zero value.
* If you don’t specify the exit status in the script, the script will assume the exit status of the latest executed command as the final exit status.

* Taking user inputs
  + Read shell built-in:
    - For this we can use the command: read [-p] ['Prompt text'] [variable\_name]
    - For instance, read -p 'Tell me something: ' THING
    - This will prompt the user with the question "Tell me something: ", whatever, the user will enter say, AWESOME. The word 'AWESOME' will be assigned to the variable THINK which we can print later.
  + Using command line arguments
* Useradd - this command is used to create new users. It required admin or root privileges.
* "Sudo su - USERNAME or su -l" : the dash is used to tell the su command to start an environment similar to a login system. This entire command will use switch to the entered USERNAME. So, earlier when it was vagrant@localhost will become shellscripting@localhost
* So the username which you can set in shell needs to be 8 characters long. If its more than 8 characters, it wont break anything but will display the username upto 8 letters followed by a + sign. For eg: if username is stanfordacademy then, shell will display it as stanfor+
* Ps -ef: command to see the entire table of history of every process
* Usernames are case sensitive
* To create a user: useradd -c "${COMMENT}" -m "${USER\_NAME}"
* We are using "" - double quotes here to include the white spaces as well
* COMMENT usually specifies either the full name of the user or the name of the application/service for whom the account is being created.
* Why do we use the symbol | also called as pipe in command line?
  + The symbol says, the command mentioned before the pip should be considered a standard input to the command mentioned after the pipe.
  + For instance, man text | less -> this will take the output of "man text" and will give it as an input to "less" that is pagination. If the first command before the pip displays error messages, it will not be passed to the command following the pip symbol.