

# MODULE 3

## Introduction to Bash

1) Write a simple Bash shell script to display the message "Welcome to Bash learning" and "\*\*\*\*\*" on separate lines.

```
balamurugan@balamurugan-Lenovo-E41-25:~$ gedit bashdemo.sh &
[1] 9951
balamurugan@balamurugan-Lenovo-E41-25:~$ chmod +x bashdemo.sh
[1]+  Done                  gedit bashdemo.sh
balamurugan@balamurugan-Lenovo-E41-25:~$ ./bashdemo.sh
Welcome to Bash learning
*****
```

## Basics of Shell Scripting

1) Write a simple Bash program to get the following system variables:

- a. pwd
- b. Logname

Script:

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

```
current_directory=$(pwd)
```

```
echo "Current Working Directory (pwd): $current_directory"
```

```
login_name=$(logname)
```

```
echo "Login Name (logname): $login_name"
```

```
balamurugan@balamurugan-Lenovo-E41-25:~$ gedit bashdemo.sh &
[1] 10345
balamurugan@balamurugan-Lenovo-E41-25:~$ chmod +x bashdemo.sh
[1]+  Done                  gedit bashdemo.sh
balamurugan@balamurugan-Lenovo-E41-25:~$ ./bashdemo.sh
Current Working Directory (pwd): /home/balamurugan
Login Name (logname): balamurugan
```

2) Write a simple Bash program

- a. To ask username from user
- b. Exit the program, if user does not enter anything within 10 seconds

### Script:

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

```
read -t 10 -p "Enter username: "
```

```
balamurugan@balamurugan-Lenovo-E41-25:~$ gedit bashdemo2.sh &
[1] 10477
balamurugan@balamurugan-Lenovo-E41-25:~$ chmod +x bashdemo2.sh
[1]+  Done                  gedit bashdemo2.sh
balamurugan@balamurugan-Lenovo-E41-25:~$ ./bashdemo2.sh
Enter username: balamurugan@balamurugan-Lenovo-E41-25:~$
```

## Command Line arguments and Quoting

1) Write a bash program for addition using command line arguments.

### Script:

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

```
a="$1"
```

```
b="$2"
```

```
ans=$((a+b))
```

```
echo "$a + $b = $ans"
```

```
balamurugan@balamurugan-Lenovo-E41-25:~$ gedit add.sh &
[2] 11010
balamurugan@balamurugan-Lenovo-E41-25:~$ chmod +x add.sh
[2]+  Done                  gedit add.sh
balamurugan@balamurugan-Lenovo-E41-25:~$ ./add.sh 11 22
11 + 22 = 33
balamurugan@balamurugan-Lenovo-E41-25:~$
```

## Globbering and Export statement

1) Write a Bash script to do all operations discussed under Globbering

### Script:

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

```
ls b*.txt
```

```
echo "_____"
```

```
ls [a-g]*.txt
```

```
echo "_____"
```

```
ls [^a-g]*.txt
```

```
echo "_____"
```

```
ls [Bb]*.txt
```

```

balamurugan@balamurugan-Lenovo-E41-25:~$ gedit globbing.sh &
[1] 11670
balamurugan@balamurugan-Lenovo-E41-25:~$ chmod +x globbing.sh
balamurugan@balamurugan-Lenovo-E41-25:~$ ./globbing.sh
bala.txt

_____
alphabets.txt  bala.txt  grepdemo.txt  grep.txt

_____
Bala.txt  numbers.txt  seddemo.txt

_____
bala.txt  Bala.txt

```

## Array Operations in BASH

1) Declare an Array names of length 7 and find

- a. The total number of elements
- b. Print all the elements
- c. Print the 5th element

Script:

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

```
declare -a names=('bala' 'ravi' 'ram' 'varsha' 'catherine' 'kumar' 'ashish')
```

```
echo "The total no. of elements: ${#names[@]}"
```

```
echo "The whole array: ${names[@]}"
```

```
echo "The 5th element is ${names[4]}"
```

```

balamurugan@balamurugan-Lenovo-E41-25:~$ gedit array.sh &
[2] 11946
balamurugan@balamurugan-Lenovo-E41-25:~$ chmod +x array.sh
[2]+  Done                  gedit array.sh
balamurugan@balamurugan-Lenovo-E41-25:~$ ./array.sh
The total no. of elements: 7
The whole array: bala ravi ram varsha catherine kumar ashish
The 5th element is catherine

```

## More on Arrays

1) Declare an Array names2 of length 7 and perform following operations-

- a. Extract three elements starting from index two.
- b. Replace third element with 'Debian' and display.
- c. Append any new name at the end of Array.

### Script:

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

```
declare -a names=('bala' 'ravi' 'ram' 'varsha' 'catherine' 'kumar' 'ashish')
```

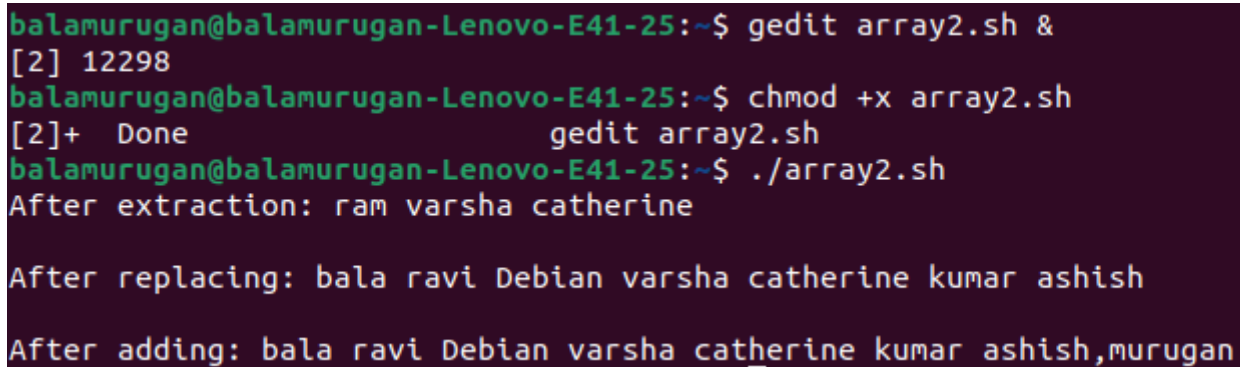
```
echo -e "After extraction: ${names[@]:2:3}\n"
```

```
names[2]='Debian'
```

```
echo -e "After replacing: ${names[@]}\n"
```

```
names=(" ${names[@]} " "murugan")
```

```
echo "After adding: ${names[@]}"
```

A terminal window with a dark purple background and green text. The user is at a prompt 'balamurugan@balamurugan-Lenovo-E41-25:~\$'. They run 'gedit array2.sh &', which opens a file editor. They then run 'chmod +x array2.sh', which returns '[2]+ Done'. Next, they run './array2.sh', which outputs 'After extraction: ram varsha catherine'. Pressing Enter shows 'After replacing: bala ravi Debian varsha catherine kumar ashish'. Pressing Enter again shows 'After adding: bala ravi Debian varsha catherine kumar ashish,murugan'.

```
balamurugan@balamurugan-Lenovo-E41-25:~$ gedit array2.sh &  
[2] 12298  
balamurugan@balamurugan-Lenovo-E41-25:~$ chmod +x array2.sh  
[2]+  Done                  gedit array2.sh  
balamurugan@balamurugan-Lenovo-E41-25:~$ ./array2.sh  
After extraction: ram varsha catherine  
  
After replacing: bala ravi Debian varsha catherine kumar ashish  
  
After adding: bala ravi Debian varsha catherine kumar ashish,murugan
```

### Conditional execution

- 1) Write a script which will take your name as an input.
- 2) It should check this name with your system's username.
- 3) If the username matches, it should greet you by displaying "Hello".
- 4) Else, it should display "Try again"

### Script:

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

```
# Get the user's name as input
```

```
read -p "Enter your name: " user_name
```

```
# Get the system's username
```

```
system_username="$USER"
```

```
# Check if the entered name matches the system's username
```

```
if [ "$user_name" = "$system_username" ]; then
```

```
    echo "Hello"
```

```
else
```

```
    echo "Try again"
```

```
fi
```

```

balamurugan@balamurugan-Lenovo-E41-25:~$ gedit ifdemo.sh &
[2] 12727
balamurugan@balamurugan-Lenovo-E41-25:~$ chmod +x ifdemo.sh
[2]+  Done                  gedit ifdemo.sh
balamurugan@balamurugan-Lenovo-E41-25:~$ ./ifdemo.sh
Enter your name: balamurugan
Hello
balamurugan@balamurugan-Lenovo-E41-25:~$ ./ifdemo.sh
Enter your name: bala
Try again

```

## Nested and multilevel if elsif statements

1) Write a program to output different messages when number is:

a. Greater than 3 b. Lesser than 3 c. Or equal to 3 d. Or when the user input is empty

Script:

```

#!/bin/bash
read -p "Enter a number: " user_input

if [ -z "$user_input" ]; then
    echo "Input is empty."
else
    # Check if the input is greater than 3
    if [ "$user_input" -gt 3 ]; then
        echo "$user_input is greater than 3."
    # Check if the input is less than 3
    elif [ "$user_input" -lt 3 ]; then
        echo "$user_input is less than 3."
    # Check if the input is equal to 3
    else
        echo "$user_input is equal to 3."
    fi
fi

```

```

balamurugan@balamurugan-Lenovo-E41-25:~$ gedit ifdemo2.sh &
[2] 12974
balamurugan@balamurugan-Lenovo-E41-25:~$ chmod +x ifdemo2.sh
[2]+  Done                  gedit ifdemo2.sh
balamurugan@balamurugan-Lenovo-E41-25:~$ ./ifdemo2.sh
Enter a number: 2
2 is less than 3.

```

