

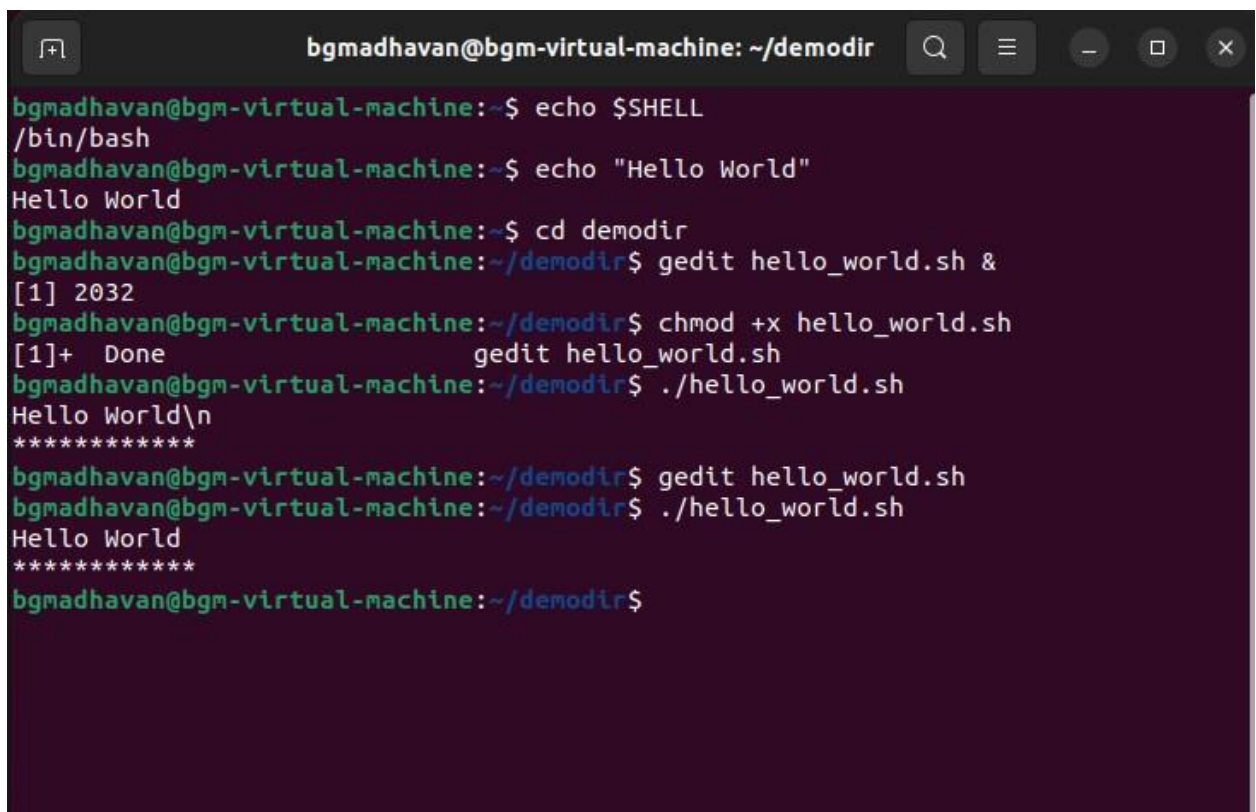
Module 3 – Bash assignment

GOKULA MADHAVAN B

SRI SAIRAM ENGINEERING COLLEGE

Introduction to Bash

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

A terminal window titled 'bgmadhavan@bgm-virtual-machine: ~/demodir' with standard window controls. The terminal shows a series of commands and their outputs. The user first checks the shell type with 'echo \$SHELL', then echoes 'Hello World'. They then navigate to the 'demodir' directory, create a file 'hello_world.sh' using 'gedit', set permissions with 'chmod +x', and execute it with './hello_world.sh'. The script outputs 'Hello World\n' followed by '*****'. The user repeats the creation and execution of the script, showing the same output again.

```
bgmadhavan@bgm-virtual-machine:~$ echo $SHELL
/bin/bash
bgmadhavan@bgm-virtual-machine:~$ echo "Hello World"
Hello World
bgmadhavan@bgm-virtual-machine:~$ cd demodir
bgmadhavan@bgm-virtual-machine:~/demodir$ gedit hello_world.sh &
[1] 2032
bgmadhavan@bgm-virtual-machine:~/demodir$ chmod +x hello_world.sh
[1]+  Done                  gedit hello_world.sh
bgmadhavan@bgm-virtual-machine:~/demodir$ ./hello_world.sh
Hello World\n
*****
bgmadhavan@bgm-virtual-machine:~/demodir$ gedit hello_world.sh
bgmadhavan@bgm-virtual-machine:~/demodir$ ./hello_world.sh
Hello World
*****
bgmadhavan@bgm-virtual-machine:~/demodir$
```

Basics of Shell Scripting

- 1) Write a simple Bash program to get the following system variables:
 - a. pwd
 - b. logname

```
bgmadhavan@bgm-virtual-machine: ~  
bgmadhavan@bgm-virtual-machine:~$ gedit bashprog1.sh &  
[1] 2447  
bgmadhavan@bgm-virtual-machine:~$ ./bashprog1.sh  
/home/bgmadhavan  
bgmadhavan  
[1]+  Done                  gedit bashprog1.sh  
bgmadhavan@bgm-virtual-machine:~$ ./bashprog1.sh  
/home/bgmadhavan  
bgmadhavan  
bgmadhavan@bgm-virtual-machine:~$ s
```

- 2) Write a simple Bash program:
- To ask username from user
 - Exit the program, if user does not enter anything within 10 seconds

Hint: `read -t 10 -p`

```
Open  bashprog2.sh  
1 #!/bin/bash  
2 read -t 10 -p "Enter Username: " username  
3 echo "Hello $username"
```

```
bgmadhavan@bgm-virtual-machine: ~  
bgmadhavan@bgm-virtual-machine:~$ gedit bashprog1.sh &  
[1] 2447  
bgmadhavan@bgm-virtual-machine:~$ ./bashprog1.sh  
/home/bgmadhavan  
bgmadhavan  
[1]+  Done                  gedit bashprog1.sh  
bgmadhavan@bgm-virtual-machine:~$ ./bashprog1.sh  
/home/bgmadhavan  
bgmadhavan  
bgmadhavan@bgm-virtual-machine:~$ gedit bashprog2.sh &  
[1] 2491  
bgmadhavan@bgm-virtual-machine:~$ chmod +x bashprog2.sh  
[1]+  Done                  gedit bashprog2.sh  
bgmadhavan@bgm-virtual-machine:~$ ./bashprog2.sh  
Enter Username: bgmadhavan  
Hello bgmadhavan  
bgmadhavan@bgm-virtual-machine:~$ ./bashprog2.sh  
Enter Username: Hello  
bgmadhavan@bgm-virtual-machine:~$
```

Command Line arguments and Quoting

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

```
cmdarg.sh  
~/  
1 #!/bin/bash  
2 x=`expr $1 + $2`  
3 echo "The sum of $1 and $2 = $x"  
  
bgmadhavan@bgm-virtual-machine:~$ gedit cmdarg.sh &  
[2] 3039  
bgmadhavan@bgm-virtual-machine:~$ ./cmdarg.sh 45 87  
The sum of 45 and 87 = 132  
[2]+  Done                  gedit cmdarg.sh  
bgmadhavan@bgm-virtual-machine:~$
```

Globbing and Export statement

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



```
Open  glob.sh
1 #!/bin/bash
2 ls *.sh
3 echo "*****"
4 ls c*.sh
5 echo "*****"
6 ls [a-c]*.sh
7 echo "*****"
8 ls [^a-c]*.sh
9 echo "*****"
10 ls aA*.sh
```



```
bgmadhavan@bgm-virtual-machine: ~
bgmadhavan@bgm-virtual-machine:~$ gedit glob.sh &
[1] 3258
bgmadhavan@bgm-virtual-machine:~$ chmod +x glob.sh
bgmadhavan@bgm-virtual-machine:~$ ./glob.sh
bashprog1.sh bashprog2.sh cmdarg.sh glob.sh
*****
cmdarg.sh
*****
bashprog1.sh bashprog2.sh cmdarg.sh
*****
glob.sh
*****
```

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

```
bgmadhavan@bgm-virtual-machine: ~  
bgmadhavan@bgm-virtual-machine:~$ declare -a array1=('IT' 'CS' 'ECE' 'EEE' 'CSBS'  
' 'AIDS' 'MECH')  
bgmadhavan@bgm-virtual-machine:~$ echo -e "${array1[@]}"  
>  
>  
bash: unexpected EOF while looking for matching `''  
bash: syntax error: unexpected end of file  
bgmadhavan@bgm-virtual-machine:~$ echo -e "${array1[@]}"  
IT CS ECE EEE CSBS AIDS MECH  
bgmadhavan@bgm-virtual-machine:~$ echo -e "Number of elements in array :${#array  
1[@]}"  
Number of elements in array :7  
bgmadhavan@bgm-virtual-machine:~$ echo -e "Number of elements in array :${#array  
1[@]}"  
Number of elements in array :7  
bgmadhavan@bgm-virtual-machine:~$ echo -e "Fifth element in the array :${array1[  
4]}"  
Fifth element in the array :CSBS  
bgmadhavan@bgm-virtual-machine:~$
```

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.

```
bgmadhavan@bgm-virtual-machine: ~  
bgmadhavan@bgm-virtual-machine:~$ echo -e "${array1[@]:2:3}"  
ECE EEE CSBS  
bgmadhavan@bgm-virtual-machine:~$ echo -e "${array1[@]}"  
IT CS ECE EEE CSBS AIDS MECH  
bgmadhavan@bgm-virtual-machine:~$ array1[2]='Debian'  
bgmadhavan@bgm-virtual-machine:~$ echo -e "${array1[@]}"  
IT CS Debian EEE CSBS AIDS MECH  
bgmadhavan@bgm-virtual-machine:~$ array1=("${array1[@]}" "Fedora" "ubuntu")  
bgmadhavan@bgm-virtual-machine:~$ echo -e "${array1[@]}"  
IT CS Debian EEE CSBS AIDS MECH Fedora ubuntu  
bgmadhavan@bgm-virtual-machine:~$
```

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"

```
Open  cond.sh ~/
1 #!/bin/bash
2 read -p "Enter your Username : " username
3 if [ $username == $USER ]; then
4     echo "Hello $username"
5 else
6     echo "Try again"
7 fi
```

```
bgmadhavan@bgm-virtual-machine: ~
bgmadhavan@bgm-virtual-machine:~$ ./cond.sh
Enter your Username : bgmadhavan
Hello bgmadhavan
bgmadhavan@bgm-virtual-machine:~$ ./cond.sh
Enter your Username : virat
Try again
bgmadhavan@bgm-virtual-machine:~$
```

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

```
Open  nestedcond.sh ~/
1 #!/bin/bash
2 read -p "Enter a number: " n
3 if [ $n -gt 3 ]; then
4     echo "Number is greater than 3"
5 elif [ $n -lt 3 ]; then
6     echo "Number is lesser than 3"
7 elif [ $n -eq 3 ]; then
8     echo "Number is equal to 3"
9 else
10     echo "Invalid Number"
11 fi
```



bgmadhavan@bgm-virtual-machine: ~

```
bgmadhavan@bgm-virtual-machine:~$ gedit nestedcond.sh &
[1] 5197
bgmadhavan@bgm-virtual-machine:~$ ./nestedcond.sh
Enter a number: 4
Number is greater than 3
bgmadhavan@bgm-virtual-machine:~$ ./nestedcond.sh
Enter a number: 3
Number is equal to 3
bgmadhavan@bgm-virtual-machine:~$ ./nestedcond.sh
Enter a number: 2
Number is lesser than 3
bgmadhavan@bgm-virtual-machine:~$ ./nestedcond.sh
Enter a number:
./nestedcond.sh: line 3: [: -gt: unary operator expected
./nestedcond.sh: line 5: [: -lt: unary operator expected
./nestedcond.sh: line 7: [: -eq: unary operator expected
Invalid Number
bgmadhavan@bgm-virtual-machine:~$
```