

## CS241 : Assignment III-Part(2)

Prof. Hemangee K. Kapoor  
Dr. Aryabartta Sahu  
Department of CSE,  
IIT Guwahati

September 26, 2018

---

A shell script is a computer program designed to be run by the Unix/Linux shell which could be one of the following:

- The Bourne Shell
- The C Shell
- The Korn Shell
- The GNU Bourne-Again SHell (BASH)

A shell is a command-line interpreter and typical operations performed by shell scripts include file manipulation, program execution, and printing text. You may know the shell types that are supported by your operating system. You can type the command in your terminal under the root directory **cat etc/shells**. It will display the shell names supported by your operating system. The default shell of the system is BASH shell.

### **Follow up Tutorial**

–[https://www.tutorialspoint.com/unix/shell\\_scripting.htm](https://www.tutorialspoint.com/unix/shell_scripting.htm)

### **Problem-1: Use of echo, read the data from the terminal, comments in shell**

Write a shell program that will

- print a message on terminal **This is my first shell script**
- read from the terminal **Well done**
- Also show how to do single line and multiline comments in shell program

### **Problem-2: Uses of system commands in shell**

Write a Unix Shell Script which prints the following

- Current working directory.
- Current user name.

- The message “Today is :” with current date in MM/dd/yyyy format
- The message “No of users logged in :” with total no of current logged in users
- The message “Terminal :” With you own terminal number

**Problem-3: Uses of conditions**

- Write a shell script to perform like calculator. It should ask for the number and operand from the user. For conditions, you have to use if..elif
- Write a shell scripts to generate random number. For conditions, you have to use case...esac. Allow user to guess 6 times to get the no. Print appropriate messages after each guess. e.g.  
Random generated is 10  
Users enters 4 message : Too small  
User enters 14 then message : Too big

**Problem-4: how to store string in a shell program and how to use conditions**

- Write a script to find out string is palindrome or not. If we read a string from end to begin, it is same as begin to end. e.g.  
Input: ABCDCBA  
Output: Yes

**Problem-5: how to delete files and directory**

- Write a shell scripts that delete all files in current directory with 0 byte.

**Problem-6: how to read from the file and append in a file**

- Write a shell scripts to count number of vowels in file ignoring the case.
- Write a scripts which copies the content of file1 to file2 without using cp command It should check If file has a read permissions if not it should print an error message. If file2 exists then it should ask the user whether he wants to overwrite it.
- Write a shell script to accept two filenames and check if both exists. If second file exists then append the content of first file to second file.

**Problem-7: Write in a file**

- Write a shell script to display a directory listing as follows. Your home directory is <home directory name>
- | File name                          | date | time | permission |
|------------------------------------|------|------|------------|
| Filename1                          | date | time | permission |
| Filename2                          | date | time | permission |
| Filename3                          | date | time | permission |
| .....                              |      |      |            |
| .....                              |      |      |            |
| Total no. of files: <total number> |      |      |            |
| Total no of directory: <number>    |      |      |            |

- write the following output snippet in the file final\_output.txt  
Total no. of files: <total number>  
Total no of directory: <number>
- Write a shell script for accepting the following information and storing it in a file. CD No., Movie Name, Language, Price and Date of release

**Problem-8: Use the arrays, nested loops and functions**

- Write a shell script to examine all the number from 1 to 999 and display all those number whose sum of cube of the digit is equal to the number.  
e.g.  $371 = 3 \times 3 \times 3 + 7 \times 7 \times 7 + 1 \times 1 \times 1$
- Write a shell script to read two matrices from the user and add them & display their output on the terminal. You have to create a function add in the shell program that will perform the addition of matrices. You can use any of the loop among while, for, until.

**Problem-9: How to debug a shell script?**

- Debug the following debug.sh shell program using the command **bash -x ./shellprogram\_name**. Demonstrate how you can debug a shell program. (Note: line numbers are not the part of the program)
- Demonstrate how to debug the shell program by using debug options in the shell program instead of commandline.
- Also, demonstrate how to do debugging only a code snippet of shell program using **set -x** and **set +x** options for the shell program.
- What is the output of the program after debugging?

**Program: debug.sh**

```

1. #!/bin/sh
2. touch /home/user/Downloads/sample.txt
3. file=/home/user/Downloads/sample.txt
4. trap "rm -f $file && echo file deleted; exit" 0 2 15
5.
6. echo "pid is $$"
7. while (( COUNTER < 10 ))
8. do
9.     sleep 1
10.    (( COUNT ++ ))
11.    echo $COUNT
12. done
13. exit 0

```

**Problem-10: Few exercise on practical uses of shell**

Perform the following tasks using shell program.

- **Image resizing and format conversion.** You have to convert the jpg and jpeg images into pdf in the same directory. Remove all the jpg and jpeg files from the directory. Put all pdf of images in a single directory

**image** inside your home directory. Check the size of the image directory. Scale the each image by 50% to reduce their size. Now print the size of the image directory. Write a shell program to perform all these operation using shell programming.

- Split the `Linux.Shell.Scripting.Cookbook.pdf` file into 192 pdfs in the directory `multiple_pdf`. Each pdf should have two pages after splitting into multiple pdfs. Now move all the pdf into a home directory named as **Success** and remove the older directory `multiple_pdf`.
- Convert excel files `marks.xlsx` and `grades.xlsx` into `marks.csv` and `grades.csv` files of either comma separated or tab space separated. Add a column with name `grades` in the `marks.csv` file, the input for the eighth column grade should read through another file `grades.csv`. Delete the column four, five and six from the file `marks.csv` and create the `xlsx` file `final_marksheet.xlsx` from the file `marks.csv`.
- Read the `auth.log` file. Find the total time for the user `sonal` who spent on the system. You have to add all the times between session opened and session closed in `auth.log` file.

#### **Follow up Tutorial: `Linux.Shell.Scripting.Cookbook.pdf`**

During the evaluation, we may ask similar kind of other questions on shell, so you should go through the shell tutorial. For the doubts in the assignment, you can contact to Sonal.