

## Module # 14

### References:

- [1] S. Jain, *100 Shell Programs in Unix*. Pinnacle Technology, 2009.
- [2] M. Garrels, *Bash Guide for Beginners (Second Edition)*. Fultus Corporation, 2010.
- [3] Isrd, *Basics Of Os Unix And Shell Programming*. Tata McGraw-Hill Education, 2006.
- [4] P. Seebach, *Beginning Portable Shell Scripting: From Novice to Professional*. Apress, 2008.
- [5] E. Foster-Johnson, J. C. Welch, and M. Anderson, *Beginning Shell Scripting*. John Wiley & Sons, 2007.
- [6] A. Robbins and N. H. F. Beebe, *Classic Shell Scripting: Hidden Commands that Unlock the Power of Unix*. O'Reilly Media, Inc., 2005.
- [7] R. Peters, *Expert Shell Scripting*. Apress, 2009.
- [8] M. G. Venkateshmurthy, *Introduction to Unix and Shell Programming*. Pearson Education India, 2005.
- [9] G. Festari, *Learning Shell Scripting with Zsh*. Packt Publishing Ltd, 2014.
- [10] C. Newham, *Learning the bash Shell: Unix Shell Programming*. O'Reilly Media, Inc., 2005.
- [11] B. Rosenblatt and A. Robbins, *Learning the Korn Shell*. O'Reilly Media, Inc., 2002.
- [12] W. E. S. Jr, *Linux Command Line*. NO STARCH Press, 2012.
- [13] R. Blum and C. Bresnahan, *Linux Command Line and Shell Scripting Bible*. John Wiley & Sons, 2015.
- [14] R. Blum and C. Bresnahan, *Linux Command Line and Shell Scripting Bible*. John Wiley & Sons, 2015.
- [15] S. Lakshman, *Linux Shell Scripting Cookbook*. Packt Publishing Ltd, 2011.
- [16] S. Tushar, *Linux Shell Scripting Cookbook*. Packt Publishing Ltd, 2013.
- [17] K. O. Burtch, *Linux Shell Scripting with Bash*. Sams, 2004.
- [18] R. K. Michael, *Mastering Unix Shell Scripting: Bash, Bourne, and Korn Shell Scripting for Programmers, System Administrators, and UNIX Gurus*. John Wiley & Sons, 2011.
- [19] C. Johnson, *Pro Bash Programming: Scripting the Linux Shell*. Apress, 2009.
- [20] S. Veeraraghavan, *Sams Teach Yourself Shell Programming in 24 Hours*. Sams Publishing, 2002.
- [21] S. Parker, *Shell Scripting: Expert Recipes for Linux, Bash and more*. John Wiley & Sons, 2011.
- [22] C. F. A. Johnson, *Shell Scripting Recipes: A Problem Solution Approach*. Dreamtech Press, 2007.
- [23] A. Verma, *Unix and Shell Programming*. Laxmi Publications, 2006.
- [24] B. A. Forouzan and R. F. Gilberg, *UNIX and Shell Programming: A Textbook*. Brooks/Cole-Thomson Learning, 2003.
- [25] Y. P. Kanetkar, *Unix Shell Programming*. BPB Publications, 2002.
- [26] T. Sanchez-Clark, *Unix Shell Scripting Interview Questions, Answers, and Explanations: Unix Shell Certification Review*. Equity Press, 2007.
- [27] D. Taylor, *Wicked Cool Shell Scripts: 101 Scripts for Linux, Mac OS X, and Unix Systems*. No Starch Press, 2004.

## Few Shell Scripts for Reference:

### Script-1: String Processing

```
clear
echo "STRING MANIPULATION PROGRAM"
echo "1. COMPARE STRING"
echo "2. JOINT TWO STRING"
echo "3. LENGTH OF STRING"
echo "4. OCCOURANCE OF CHARACTER"
echo "5. OCCOURANCE OF WORD"
echo "6. REVERSE STRING"
echo "ENTER CHOICE:"
read ch
case $ch in
    1)
        echo "ENTER FIRST STRING:"
        read str1
        echo "ENTER SECOND STRING:"
        read str2

        len1=`echo $str1 | wc -c`
        len2=`echo $str2 | wc -c`

        if [ $len1 -eq $len2 ];then
            echo "BOTH STRINGS ARE OF SAME LENGTH"
        elif [ $len1 -gt $len2 ];then
            echo "$str1 is greater than $str2"
        else
            echo "$str2 is greater than $str1"
        fi

        ;;
    2)
        echo "ENTER FIRST STRING:"
        read str1
        echo "ENTER SECOND STRING:"
        read str2
        str1="$str1$str2"
        echo "COMBINED STRING : $str1"

        ;;
    3)
        echo "ENTER STRING:"
        read str
        len=`echo $str | wc -c`
        echo "Length of string is $len"

        ;;
    4)
        echo "ENTER STRING:"
        read str
        echo "ENTER character of word to search:"
        read search
        len=`echo $str | wc -c`
        i=1
        while [ $i -lt $len ]
```

```

do
    ch=`echo $str | cut -c $i`
    echo $ch
    i=`expr $i + 1`
done
;;
5)
echo "NOT THERE"
;;
6)
echo "ENTER STRING:"
read str
echo "REVERSE STRING IS : "`echo $str | rev`
;;
*)
echo "ENTER PROPER CHOICE"
echo "NOT THERE"
;;
esac

```

### **Script-2: Display of Greeting Message**

```

clear
HH=`date +%H`
time=`date +%S %p`
if [ $HH -ge 12 ];then
    HH=`expr $HH % 12`
    if [ $HH -lt 5 ];then
        msg="GOOD AFTERNOON"
    elif [ $HH -ge 5 ] && [ $HH -lt 9 ];then
        msg="GOOD EVENING"
    else
        msg="GOOD NIGHT"
    fi
    echo "$msg ,CURRENT TIME $HH:$time"
    exit 1
else
    if [ $HH -lt 5 ];then
        msg="GOOD NIGHT"
    else
        msg="GOOD MORNING"
    fi
    echo "$msg ,CURRENT TIME $HH:$time"
    exit 1
fi

```

### **Script-3: Display of All words of a file in ascending order**

```

clear
echo "Ascending order of word in File"
echo " ====="
echo "Enter File Name "
read filename
echo "\n\n\t\t\t Main File "

```

```

echo "\t\t\t ====="
echo `cat $filename`
for i in `cat $filename`
do
echo $i >> tempfile.txt
done
`sort tempfile.txt>$filename`
echo "\n\n\t\t\tSorted Word in File "
echo "\t\t\t =====\n"
echo `cat $filename`
rm tempfile.txt

```

#### **Script-4: Delete Zero sized file from current directory**

```

clear
`echo ls`>filelist
for filename in `cat filelist`
do
if [ ! -d $filename ];then
size=`ls -s $filename`
size=`echo $size|cut -d " " -f 1`
if [ $size -eq 0 ];then
echo "Want to Remove $filename:"
rm -i $filename
fi
fi
done

```

#### **Script-5: Calculate Gross Salary from basic, DA=20% and HRA=10%**

```

clear
echo "          GROSS SALARY CALCULATOR"
echo "          ====="
echo "ENTER BASIC SALARY:"
read basic
hra=`expr 10 \* $basic / 100`
da=`expr 20 \* $basic / 100`
gs=`expr $basic + $hra + $da`
echo "GROSS SALARY IS $gs"

```

#### **Script-6: A shell script illustrating the use of the case construct.**

```

echo"          MENU\n
1. Long List of files\n2. Today's Date\n3. Process status\n4.Users of the
System\n5. Display the present working directory and \n 6. Quit to
Unix\nEnter your option : \c"
read choice
case "$choice" in
    1) Is -l ;;
    2) date ;;
    3) ps -ef;;

```

```

        4) who -Hu;;
        5) pwd ;;
        6) exit;;
        *) echo "Invalid Choice"
           exit ;;
esac

```

#### **Script-7: Sample for loop**

```

for x in 1 3 5 7 # four members in the list
do
echo "The value of x is $x"
done

```

#### **Script-8: A shell script illustrating the use of the break and continue.**

```

ans=y
while [ "$ans" = "y" ]
echo"          MENU\n
1. List of files\n2. Today's Date\n3. Process Status\n4. Users of the
System\n5. present working directory\n6. Quit to Unix\n Enter your option :
\c"
read choice
case "$choice" in
1) ls -l ;;
2) date ;;
3) ps ;;
4) who ;;
5) pwd ;;
6) break ;;
*) echo "Invalid Choice"
   continue ;;
esac

```

#### **Script-9: A shell script to calculate area using bc and scale**

```
$cat triangle_area.sh
echo "\n Enter a value for base : \c"
read base
echo "\n Enter a value for height \c"
read height
area=`echo "scale=2 \n 1/2*$base*$height" | bc`
echo "The area of the tri4angle is $area"
```

**Script-10: A shell script that uses bc**

```
$cat degree_conv.sh
echo "\n Enter a Fahrenheit value : \c"
read fahr
cel=`echo "scale=2 \n 5/9*($fahr-32)" | bc`
echo "The equivalent degree Celsius = $cel"
```