Shell Scripting programs:

1.Forloop program

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: One more example for for loop |
|  | #Version: |
|  | #Created Date: Wed May 16 19:31:50 IST 2018 |
|  | #Modified Date: |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | for i in 1 2 3 4 5 |
|  | do |
|  | echo $i |
|  | done |
|  | # END # |

==================================================================================

2.Function.sh

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: Function example. Taking Backup of Particular File |
|  | #Version:1.0 |
|  | #Created Date: Sat May 26 00:27:50 IST 2018 |
|  | #Modified Date: |
|  | #WebSite: https://arkit.co.in |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | function takebackup (){ |
|  | if [ -f $1 ]; then |
|  | BACKUP="/home/aravi/$(basename ${1}).$(date +%F).$$" |
|  | echo "Backing up $1 to ${BACKUP}" |
|  | cp $1 $BACKUP |
|  | fi |
|  | } |
|  |  |
|  | takebackup /etc/hosts |
|  | if [ $? -eq 0 ]; then |
|  | echo "BAckup Success" |
|  | fi |
|  | function testing (){ |
|  | echo "Just TEsting Function" |
|  | } |
|  |  |
|  | testing |
|  | # END #  ============================================================================= |

## 3.Arthemetic-Operators.sh

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: Arthemetic Operators |
|  | #Version:1.0 |
|  | #Created Date: Wed May 9 21:41:53 IST 2018 |
|  | #Modified Date: |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | echo -e "Please enter some value: \c" |
|  | read -r a |
|  | echo -e "Please enter another value: \c" |
|  | read -r b |
|  |  |
|  | echo "a+b value is $(($a+$b))" |
|  | echo "a-b value is $(($a-$b))" |
|  | echo "axb value is $(($a\*$b))" |
|  | echo "a/b value is $(($a/$b))" |
|  | echo "a%b value is $(($a%$b))" |
|  |  |
|  | echo "Completed successfully" |
|  | ============================================================================= |
|  |  |

## 4. **CpuMemDisk.sh**

|  |
| --- |
| #!/bin/bash |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | HOSTNAME=$(hostname) |
|  | DATET=$(date "+%Y-%m-%d %H:%M:%S") |
|  | CPUUSAGE=$(top -b -n 2 -d1 | grep "Cpu(s)" | tail -n1 | awk '{print $2}' |awk -F. '{print $1}') |
|  | MEMUSAGE=$(free | grep Mem | awk '{print $3/$2 \* 100.0}') |
|  | DISKUSAGE=$(df -h / | awk '{print $5}' |tail -n 1 |sed 's/%//g') |
|  |  |
| ~ | echo 'HostName, Date&Time, CPU(%), MEM(%), DISK(%)' >> /opt/usagereport |
|  | echo "$HOSTNAME, $DATET, $CPUUSAGE, $MEMUSAGE, $DISKUSAGE" >> /opt/usagereport |
|  |  |
|  | for i in `cat /scripts/hostlist` |
|  | do |
|  | RHOST=$(ssh $i hostname) |
|  | RDATET=$(ssh $i 'date "+%Y-%m-%d %H:%M:%S"') |
|  | RCPUUSAGE=$(ssh $i top -b -n 2 -d1 | grep "Cpu(s)" | tail -n1 | awk '{print $2}' |awk -F. '{print $1}') |
|  | RMEMUSAGE=$(ssh $i free | grep Mem | awk '{print $3/$2 \* 100.0}') |
|  | RDISKUSAGE=$(ssh $i df -P / |column -t | awk '{print $5}' |tail -n 1 |sed 's/%//g') |
|  |  |
|  | echo "$RHOST, $RDATET, $RCPUUSAGE, $RMEMUSAGE, $RDISKUSAGE" >> /opt/usagereport |
|  | Done  =========================================================================== |

## 5. **Logical-operators.sh**

|  |
| --- |
| #!/bin/bash |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | # START # |
|  | echo -e "Enter Your Maths Subject Marks: \c" |
|  | read -r m |
|  | echo -e "Enter Your Physics Subject Marks: \c" |
|  | read -r p |
|  | echo -e "Enter Your Chemistry Subject Marks: \c" |
|  | read -r c |
|  |  |
|  | if test $m -ge 35 -a $p -ge 35 -a $c -ge 35 |
|  | then |
|  | echo "Congratulations, You have passed in all subjects" |
|  | else |
|  | echo "Sorry You not upto mark in one of the subject" |
|  | fi |
|  | # END # |

## 6. **add.sh**

|  |
| --- |
| #!/bin/bash |
|  |  |
|  | echo "addition of X+Y" |
|  | echo "Enter X" |
|  | read X |
|  | echo "Enter Y" |
|  | read Y |
|  | echo "X+Y = $X+$Y = $[ X+Y ]" |
|  |  |
|  | # END # |

## 7. **arithmetic.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose:Arthmetic operators using expr command |
|  | #Version:1.0 |
|  | #Created Date: Wed May 9 21:47:04 IST 2018 |
|  | #Modified Date: |
|  | #website: https://arkit.co.in |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | echo -e "Enter value: \c" |
|  | read -r a |
|  | echo -e "Enter value: \c" |
|  | read -r b |
|  |  |
|  | echo "addition values `expr $a + $b`" |
|  | echo "minus values `expr $a - $b`" |
|  | echo "multiplied by values `expr $a \\* $b`" |
|  | echo "devided by values `expr $a / $b`" |
|  | echo "remainder values `expr $a % $b`" |
|  | echo "addition values `expr $a + $b`" |
|  |  |
|  | echo "Completed Sucessfully" |
|  | # END # |

8.**array.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: Array Example |
|  | #Version:1.0 |
|  | #Created Date: Mon May 28 22:59:22 IST 2018 |
|  | #Modified Date: |
|  | #WebSite: https://arkit.co.in |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | fruits=( "Apple" "Orange" "Banana" "Sapota" ) |
|  | fruits[3]='Green Apple' |
|  | for fruit in ${fruits[@]} |
|  | do |
|  | echo "Fruit Name is $fruit" |
|  | done |
|  |  |
|  | echo "Number of Fruits in Bucket is" ${#fruits[@]} |
|  | echo "All Fruits ${fruits[@]}" |
|  |  |
|  | # END # |

## 9. **casestatement.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: Example for Case Statement |
|  | #Version:1.0 |
|  | #WebSite: https://arkit.co.in |
|  | #Created Date: Mon May 21 20:37:59 IST 2018 |
|  | #Modified Date: |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | echo -c "Enter a number: \c" |
|  | read -r a |
|  | echo -c "Enter b number: \c" |
|  | read -r b |
|  |  |
|  | echo "1. Sum of values" |
|  | echo "2. Substraction" |
|  | echo "3. Multiplication" |
|  | echo "4. Division" |
|  | echo "5. Modulo division" |
|  | echo -c "Enter Your Choice from above menu: \c" |
|  | read -r ch |
|  | case $ch in |
|  | 1) echo "Sum of $a + $b = "`expr $a + $b`;; |
|  | 2) echo "Subsctraction = "`expr $a - $b`;; |
|  | 3) echo "Multiplication = "`expr $a \\* $b`;; |
|  | 4) echo "Division = "`expr $a / $b`;; |
|  | 5) echo "Modulo Division = "`expr $a % $b`;; |
|  | \*) echo "Invalid Option provided" |
|  | esac |
|  | # END # |

## 10. **continue.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: While loop Continue Statement |
|  | #Version:1.0 |
|  | #Website: https://arkit.co.in |
|  | #Created Date: Tue May 22 22:03:02 IST 2018 |
|  | #Modified Date: |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | opt=y |
|  | while [ $opt = y -o $opt = Y ] |
|  | do |
|  | echo -e "Please enter the number: \c" |
|  | read -r num |
|  | if [ $num -le 50 ]; then |
|  | sq=`expr $num \\* $num` |
|  | echo "Square of provided number $num: $sq" |
|  | else |
|  | echo "Number not in the given Range" |
|  | fi |
|  |  |
|  | echo -e "Do you want to continue [y/n]: \c" |
|  | read -r wish |
|  | if [ $wish = y -o $wish = Y ]; then |
|  | continue |
|  | else |
|  | echo "Thank You for Exiting.." |
|  | exit |
|  | fi |
|  | done |

## 11. **countargs.sh**

|  |
| --- |
| #!/bin/bash |
|  | ################################################## |
|  | # Purpose: Counting given postional parameters. |
|  | # Version:1.0 |
|  | # Created Date: Mon May 7 21:55:05 IST 2018 |
|  | # Modified Date: |
|  | # Author: Ankam Ravi Kumar |
|  | ################################################## |
|  |  |
|  | # START # |
|  | echo "Your current given parameters are $#" |
|  | if [ $# -lt 1 ];then |
|  | echo "Program Usage is './scriptname.sh' options" |
|  | else |
|  | echo "Program executed successfully" |
|  | fi |
|  | # END # |

## 12. **cpualert.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: Real time CPU utilization Monitoring |
|  | #Version:1.0 |
|  | #Created Date: Tue Jun 5 21:33:38 IST 2018 |
|  | #Modified Date: |
|  | #WebSite: https://arkit.co.in |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | PATHS="/" |
|  | HOSTNAME=$(hostname) |
|  | CRITICAL=98 |
|  | WARNING=90 |
|  | CRITICALMail="YOUREMAILaddresS@Domain.com" |
|  | MAILWAR="YOUREMAIL@Domain.in" |
|  | mkdir -p /var/log/cputilhist |
|  | LOGFILE=/var/log/cputilhist/cpusage-`date +%h%d%y`.log |
|  |  |
|  | touch $LOGFILE |
|  |  |
|  | for path in $PATHS |
|  | do |
|  | CPULOAD=`top -b -n 2 -d1 | grep "Cpu(s)" | tail -n1 | awk '{print $2}' |awk -F. '{print $1}'` |
|  | if [ -n $WARNING -a -n $CRITICAL ]; then |
|  | if [ "$CPULOAD" -ge "$WARNING" -a "$CPULOAD" -lt "$CRITICAL" ]; then |
|  | echo "`date "+%F %H:%M:%S"` WARNING - $CPULOAD on Host $HOSTNAME" >> $LOGFILE |
|  | echo "Warning Cpuload $CPULOAD Host is $HOSTNAME" | mail -s "CPULOAD is Warning" $MAILWAR |
|  | exit 1 |
|  | elif [ "$CPULOAD" -ge "$CRITICAL" ]; then |
|  | echo "`date "+%F %H:%M:%S"` CRITICAL - $CPULOAD on Host $HOSTNAME" >> $LOGFILE |
|  | echo "CRITICAL Cpuload $CPULOAD Host is $HOSTNAME" | mail -s "CPULOAD is CRITICAL" $CRITICALMail |
|  | exit 2 |
|  | else |
|  | echo "`date "+%F %H:%M:%S"` OK - $CPULOAD on $HOSTNAME" >> $LOGFILE |
|  | exit 0 |
|  | fi |
|  | fi |
|  | done |

## 13. **details.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: eval command Evaluating twice |
|  | #Version:1.0 |
|  | #Created Date: Wed Jun 13 22:09:59 IST 2018 |
|  | #Modified Date: |
|  | #WebSite: https://arkit.co.in |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  |  |
|  | echo "WEL COME TO $USER" |
|  | echo "Your present working directory is `pwd`" |
|  | echo "current logged in users are `who`" |
|  | echo "Today date is `date`" |
|  |  |
|  | # END # |

## 14. **diskspace.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: Monitoring Disk Space Utilization and Send Email Alert |
|  | #Version:1.0 |
|  | #Created Date: Wed Jun 6 22:38:01 IST 2018 |
|  | #Modified Date: |
|  | #WebSite: https://arkit.co.in |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | THRESHOULD=40 |
|  | mailto="root" |
|  | HOSTNAME=$(hostname) |
|  |  |
|  | for path in `/bin/df -h | grep -vE 'Filesystem|tmpfs' | awk '{print $5}' |sed 's/%//g'` |
|  | do |
|  | if [ $path -ge $THRESHOULD ]; then |
|  | df -h | grep $path% >> /tmp/temp |
|  | fi |
|  | done |
|  |  |
|  | VALUE=`cat /tmp/temp | wc -l` |
|  | if [ $VALUE -ge 1 ]; then |
|  | mail -s "$HOSTNAME Disk Usage is Critical" $mailto < /tmp/temp |
|  | fi |
|  |  |
|  | #rm -rf /tmp/temp |
|  |  |
|  |  |
|  |  |

## 15. **echo.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: eval command Evaluating twice |
|  | #Version:1.0 |
|  | #Created Date: Wed Jun 13 22:09:59 IST 2018 |
|  | #Modified Date: |
|  | #WebSite: https://arkit.co.in |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  |  |
|  | echo "current location files are `ls`" |
|  | echo "current working directory is `pwd`" |
|  |  |
|  | # END # |

## 16. **eval.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: eval command Evaluating twice |
|  | #Version:1.0 |
|  | #Created Date: Wed Jun 13 22:09:59 IST 2018 |
|  | #Modified Date: |
|  | #WebSite: https://arkit.co.in |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | COMMAND="ls -ltr /etc" |
|  | echo "$COMMAND" |
|  | eval $COMMAND |
|  | # END # |

## 17. **for-loop.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: For loop example |
|  | #Version:1.0 |
|  | #website: https://arkit.co.in |
|  | #Created Date: Wed May 16 19:26:02 IST 2018 |
|  | #Modified Date: |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | for i in `cat hostfile` |
|  | do |
|  | ping -c 1 $i > /tmp/pingresults |
|  | valid=`echo $?` |
|  | if [ $valid -gt 1 ]; then |
|  | echo "$i Host is not reachable" |
|  | else |
|  | echo "$i Host is Up" |
|  | fi |
|  | done |
|  | # END # |

## 18. **functions.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: Example for Functions |
|  | #Version:1.0 |
|  | #Created Date: Sat May 26 00:17:19 IST 2018 |
|  | #Modified Date: |
|  | #WebSite: https://arkit.co.in |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | function takebackup (){ |
|  | if [ -f $1 ]; then |
|  | BACKUP="/tmp/$(basename ${1}).$(date +%F).$$" |
|  | echo "Backing up $1 to ${BACKUP}" |
|  | cp $1 $BACKUP |
|  | fi |
|  | } |
|  |  |
|  | takebackup /etc/hosts |
|  | if [ $? -eq 0 ]; then |
|  | echo "Backup Success" |
|  | fi |
|  | # END # |

## 19. **getopts.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: Getopts Examples working with arguments |
|  | #Version:1.0 |
|  | #Created Date: Wed May 30 22:30:51 IST 2018 |
|  | #Modified Date: |
|  | #WebSite: https://arkit.co.in |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  |  |
|  | while getopts :a:b: options; do |
|  | case $options in |
|  | a) ap=$OPTARG;; |
|  | b) bo=$OPTARG;; |
|  | ?) echo "I Dont know What is $OPTARG is" |
|  | esac |
|  | done |
|  |  |
|  | echo "Option A = $ap and Option B = $bo" |
|  |  |
|  | # END # |

## 20. **helloworld.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: This is the Sample template File |
|  | #Version: 1.0 |
|  | #Created Date: Thu May 3 11:55:43 IST 2018 |
|  | #Modified Date: |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | echo "Testing template file" |
|  | # END # |

## 21. **heredoc.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: Here Document Example |
|  | #Version:1.0 |
|  | #Created Date: Tue Jun 12 22:50:23 IST 2018 |
|  | #Modified Date: |
|  | #WebSite: https://arkit.co.in |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | ftp -n <<- EOF 2> /dev/null |
|  | open ftp.server.com |
|  | user ftp ftp |
|  | ascii |
|  | cd uploadfolder |
|  | mput file1 file1 file2 |
|  | bye |
|  | EOF |

## 22. **hi.sh**

|  |
| --- |
|  |
| #!/bin/bash |
|  | #Purpose: eval command Evaluating twice |
|  | #Version:1.0 |
|  | #Created Date: Wed Jun 13 22:09:59 IST 2018 |
|  | #Modified Date: |
|  | #WebSite: https://arkit.co.in |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  |  |
|  | tmp=`date | cut -c12-13` |
|  | if [ $tmp -lt 11 ] ; then |
|  | echo "Good Mornind have a nice day $USER" |
|  | elif [ $tmp -gt 11 -a $tmp -lt 16 ] ; then |
|  | echo "Good Ofter noon $USER" |
|  | elif [ $tmp -gt 15 -a $tmp -lt 19 ] ; then |
|  | echo "Good Evening $USER" |
|  | else |
|  | echo "Good Night Sweet dreams $USER" |
|  | fi |
|  | echo "Now the time is `date |cut -c12-19`" |
|  |  |

## 23. **if-elif-if.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: Find biggest Number among 4 digits |
|  | #Version:1.0 |
|  | #Created Date: Wed May 16 18:45:58 IST 2018 |
|  | #Modified Date: |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | echo -e "Please Enter a Value: \c" |
|  | read -r a |
|  | echo -e "Please Enter b Value: \c" |
|  | read -r b |
|  | echo -e "Please Enter c Value: \c" |
|  | read -r c |
|  | echo -e "Please Enter d Value: \c" |
|  | read -r d |
|  |  |
|  | if [ $a -gt $b -a $a -gt $c -a $a -gt $d ]; then |
|  | echo "$a a is big" |
|  | elif [ $b -gt $c -a $b -gt $d ]; then |
|  | echo "$b b is big" |
|  | elif [ $c -gt $d ]; then |
|  | echo "$c c is big" |
|  | else |
|  | echo "$d d is big" |
|  | fi |
|  |  |
|  | # END # |

## 24**if-else-statement.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: If else statement example |
|  | #Version:1.0 |
|  | #Created Date: Sat May 12 23:49:15 IST 2018 |
|  | #Modified Date: |
|  | #Website: https://arkit.co.in |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | echo -e "Enter any value> \c" |
|  | read -r a |
|  | echo -e "Enter any value: \c" |
|  | read -r b |
|  |  |
|  | if [ $a -gt $b ]; then |
|  | echo "$a is greater than $b" |
|  | else |
|  | echo "$b is greater than $a" |
|  | fi |
|  | # END # |

## 25. **if-statement.sh**

|  |
| --- |
|  |
| #!/bin/bash |
|  | #Purpose: If statement example |
|  | #Version:1.0 |
|  | #Created Date: Sat May 12 23:41:50 IST 2018 |
|  | #Modified Date: |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | echo -e "Please provide Value below ten: \c" |
|  | read -r value |
|  |  |
|  | if [ $value -le 10 ] |
|  | then |
|  | echo "You provided value is $value" |
|  | touch /tmp/test{1..100}.txt |
|  | echo "Script completed successfully" |
|  | fi |
|  |  |
|  |  |

## 26. **morethanxdays.sh**

|  |
| --- |
| #!/bin/bash |
|  | ## Delete the Directories older than 2 days based on directory name validation |
|  | ## Refer YouTube Link for Explanation https://youtu.be/1Sh6PWcgXAA |
|  | ls -ltr /fullbackup/archive/ | awk '{print $9}' > /scripts/dirs |
|  | for i in `cat /scripts/dirs`; do |
|  | STARTTIME=$(date +%s -d"$i 00:00:00") |
|  | ENDTIME=$(date +%s) |
|  | echo $((ENDTIME-STARTTIME)) | awk '{print int($1/60)}' > /scripts/value |
|  | COUNT=`cat /scripts/value` |
|  | if [ $COUNT -gt 2880 ]; then |
|  | echo "Directories are older than 2days $i" >> /scripts/joblog |
|  | rm -rf /fullbackup/archive/$i |
|  | fi |
|  | done |

## 27. **multiplication.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: eval command Evaluating twice |
|  | #Version:1.0 |
|  | #Created Date: Wed Jun 13 22:09:59 IST 2018 |
|  | #Modified Date: |
|  | #WebSite: https://arkit.co.in |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  |  |
|  | echo "multification of X\*Y" |
|  | echo "Enter X" |
|  | read X |
|  | echo "Enter Y" |
|  | read Y |
|  | echo "X\*Y = $X\*$Y = $[ X\*Y ]" |
|  |  |
|  | # END # |

## 28. **myfirstscript.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: This is my first script in this shell scripting video tutorial |
|  | #Date: Wed May 2 17:10:13 IST 2018 |
|  | #Author: Ankam Ravi Kumar |
|  | #Version: 1.0 |
|  | #Modified Date: |
|  | #Modified by: |
|  |  |
|  | # START |
|  | echo "Welcome $USER" |
|  | echo "Your present working directory is `pwd`" |
|  | echo "Today date is `date`" |
|  | # END |

## 29. **nestedif.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: Validate and report Student subject marks |
|  | #Version:1.0 |
|  | #Created Date: Wed May 16 19:00:52 IST 2018 |
|  | #Modified Date: |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | echo -e "Please Enter Maths Marks: \c" |
|  | read -r m |
|  | echo -e "Please Enter Physics Marks: \c" |
|  | read -r p |
|  | echo -e "Please Enter Chemistry Marks: \c" |
|  | read -r c |
|  |  |
|  | if [ $m -ge 35 -a $p -ge 35 -a $c -ge 35 ]; then |
|  | total=`expr $m + $p + $c` |
|  | avg=`expr $total / 3` |
|  | echo "Total Marks = $total" |
|  | echo "Average Marks = $avg" |
|  | if [ $avg -ge 75 ]; then |
|  | echo "Congrats you got Distinction" |
|  | elif [ $avg -ge 60 -a $avg -lt 75 ]; then |
|  | echo "Congrats you got First Class" |
|  | elif [ $avg -ge 50 -a $avg -lt 60 ]; then |
|  | echo "You got second class" |
|  | elif [ $avg -ge 35 -a $avg -lt 50 ]; then |
|  | echo "You Got Third Class" |
|  | fi |
|  | else |
|  | echo "Sorry You Failed" |
|  | fi |

## 30. **spacialvariables.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: To learn special variables |
|  | #Version:1.0 |
|  | #Website: https://arkit.co.in |
|  | #Created Date: Sun May 6 15:23:12 IST 2018 |
|  | #Modified Date: |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | echo "'$\*' output is $\*" |
|  | echo "'$#' output is $#" |
|  | echo "'$1 & $2' output $1 and $2" |
|  | echo "'$@' output of $@" |
|  | echo "'$?' output is $?" |
|  | echo "'$$' output is $$" |
|  | sleep 400 & |
|  | echo "'$!' output is $!" |
|  |  |
|  | echo "'$0' your current program name is $0" |
|  |  |
|  | # END # |

## 31. **systemload.sh**

|  |
| --- |
| #!/bin/bash |
|  | ################################################## |
|  | # # |
|  | # Author: Ankam Ravi Kumar # |
|  | # Website: server-computer.com # |
|  | # Date: 23-02-2019 16:59:56 # |
|  | # Purpose: Capture and Store System Load Average # |
|  | # CPU Usage and Memory Usage # |
|  | ################################################## |
|  | # Log File Path |
|  | LOGFILE=/var/log/systemload.log |
|  |  |
|  | HOSTNAME=$(hostname) |
|  | DATE=$(date "+%d-%m-%Y %H:%M:%S") |
|  | SYSTEMLOAD=$(uptime | awk '{ print $8,$9,$10,$11,$12}') |
|  | CPULOAD=$(top -b -n 2 -d1 | grep "Cpu(s)" | tail -n1 |awk '{print $2}') |
|  | MEMORYUSAGE=$(free -m |grep Mem: |tail -n1 |awk '{print $2,$3}') |
|  |  |
|  | echo "$DATE $HOSTNAME LoadAverage: $SYSTEMLOAD CPU: $CPULOAD Memory: $MEMORYUSAGE" >> $LOGFILE |

## 32. **uadd.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: eval command Evaluating twice |
|  | #Version:1.0 |
|  | #Created Date: Wed Jun 13 22:09:59 IST 2018 |
|  | #Modified Date: |
|  | #WebSite: https://arkit.co.in |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  |  |
|  | echo "ecnter the user name $NM" |
|  | read NM |
|  | echo "`useradd -d /users/$NM $NM`" |
|  |  |
|  | # END # |

## 33. **useradd.sh**

|  |
| --- |
| #!/bin/bash |
|  | # Script to add a user to Linux system |
|  | if [ $(id -u) -eq 0 ]; then |
|  | read -p "Enter username : " username |
|  | read -s -p "Enter password : " password |
|  | egrep "^$username" /etc/passwd >/dev/null |
|  | if [ $? -eq 0 ]; then |
|  | echo "$username exists!" |
|  | exit 1 |
|  | else |
|  | pass=$(perl -e 'print crypt($ARGV[0], "password")' $password) |
|  | useradd -m -p $pass $username |
|  | [ $? -eq 0 ] && echo "User has been added to system!" || echo "Failed to add a user!" |
|  | fi |
|  | else |
|  | echo "Only root may add a user to the system" |
|  | exit 2 |
|  | fi |

## 34. **userexists.sh**

|  |
| --- |
| #!/bin/bash |
|  | ##Purpose: Check given user Exits Or Not |
|  | ##Date: 27th Oct 2016 |
|  | ##Author: Ankam Ravi Kumar |
|  | ##WebSite: https://arkit.co.in |
|  |  |
|  | ##Start |
|  | echo "Please Enter User name you want check:\c" |
|  | read user |
|  | grep $user /etc/passwd > /dev/null |
|  | if [ $? -eq 0 ]; then |
|  | grep $user /etc/passwd |
|  | echo "$user Exists in this Machine" |
|  | else |
|  | echo "$user does not exists" |
|  | fi |
|  |  |
|  | ##END |

## 35. **variables.sh**

|  |
| --- |
| #!/bin/bash |
|  | #Purpose: What is variable.? How is help us in writing shell scripts |
|  | #Version:1.0 |
|  | #Created Date: Sat May 5 20:25:21 IST 2018 |
|  | #Modified Date: |
|  | #website: https://arkit.co.in |
|  | #Author: Ankam Ravi Kumar |
|  | # START # |
|  | A=10 |
|  | Ba=23 |
|  | BA=45 |
|  | HOSTNAME=$(hostname) |
|  | DATE=`date` |
|  | 1value=333 |
|  | False@Var=False |
|  | Hyphen\_a=WrongValue |
|  |  |
|  | echo "Variable A Value: $A" |
|  | echo "Variable Ba Vaule: $Ba" |
|  | echo "Variable BA Vaule: $BA" |
|  | echo "Variable HOST value: $HOSTNAME" |
|  | echo "Variable DATE value: $DATE" |
|  | echo "Wrong Variable 1value $1value" |
|  | echo 'False @ Variable' $False@Var |
|  | echo "hyphen-a Variable Value: $Hyphen\_a" |
|  |  |
|  | # END # |

## 36. **while-loop.sh**

|  |
| --- |
|  |
|  |  |
|  | #!/bin/bash |
|  | ## While Loop Example with 2 table |
|  | echo -e "Please provide one value:\c" |
|  | read -r c |
|  | i=1 |
|  | while [ $i -le 10 ] |
|  | do |
|  | b=`expr $c \\* $i` |
|  | echo "$c \* $i = $b" |
|  | i=`expr $i + 1` |
|  | done |

|  |
| --- |
| 37. **process.sh** |
|  |  |
|  |  |
|  | |  | | --- | | #! /bin/bash | |  | echo "Hello $USER" | |  | echo "Hey i am" $USER "and will be telling you about the current processes" | |  | echo "Running processes List" | |  | ps | |

## 38. **interactive.sh**

|  |
| --- |
| #! /bin/bash |
|  | echo "Hey what's Your First Name?" |
|  | read a |
|  | echo "welcome Mr./Mrs. $a, would you like to tell us, Your Last Name" |
|  | read b |
|  | echo "Thanks Mr./Mrs. $a $b for telling us your name" |
|  | echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" |
|  | echo "Mr./Mrs. $b, it's time to say you good bye" |

## 39. **special-pattern.sh**

|  |
| --- |
| #!/bin/bash |
|  |  |
|  | MAX\_NO=0 |
|  |  |
|  | echo -n "Enter Number between (5 to 9) : " |
|  | read MAX\_NO |
|  |  |
|  | if ! [ $MAX\_NO -ge 5 -a $MAX\_NO -le 9 ]; then |
|  | echo "WTF... I ask to enter number between 5 and 9, Try Again" |
|  | exit 1 |
|  | fi |
|  |  |
|  | clear |
|  |  |
|  | for ((i = 1; i <= MAX\_NO; i++)); do |
|  | for ((s = MAX\_NO; s >= i; s--)); do |
|  | echo -n " " |
|  | done |
|  | for ((j = 1; j <= i; j++)); do |
|  | echo -n " ." |
|  | done |
|  | echo "" |
|  | done |
|  | ###### Second stage ###################### |
|  | for ((i = MAX\_NO; i >= 1; i--)); do |
|  | for ((s = i; s <= MAX\_NO; s++)); do |
|  | echo -n " " |
|  | done |
|  | for ((j = 1; j <= i; j++)); do |
|  | echo -n " ." |
|  | done |
|  | echo "" |
|  | done |
|  |  |
|  | echo -e "\n\n\t\t\t Whenever you need help, Tecmint.com is always there" |

## 40. **while-read.sh**

|  |
| --- |
| #!/bin/bash |
|  | # while-read: read lines from a file |
|  | count=0 |
|  | while read; do |
|  | printf "%d %s\n" $REPLY |
|  | count=$(expr $count + 1) |
|  | done <$1 |

## 41. **read-menu.sh**

|  |
| --- |
| #!/bin/bash |
|  | # read-menu: a menu driven system information program |
|  | clear |
|  | echo " |
|  | Please Select: |
|  |  |
|  | 1. Display System Information |
|  | 2. Display Disk Space |
|  | 3. Display Home Space Utilization |
|  | 0. Quit |
|  | " |
|  | read -p "Enter selection [0-3] > " |
|  |  |
|  | if [[ $REPLY =~ ^[0-3]$ ]]; then |
|  | if [[ $REPLY == 0 ]]; then |
|  | echo "Program terminated." |
|  | exit |
|  | fi |
|  | if [[ $REPLY == 1 ]]; then |
|  | echo "Hostname: $HOSTNAME" |
|  | uptime |
|  | exit |
|  | fi |
|  | if [[ $REPLY == 2 ]]; then |
|  | df -h |
|  | exit |
|  | fi |
|  | if [[ $REPLY == 3 ]]; then |
|  | if [[ $(id -u) -eq 0 ]]; then |
|  | echo "Home Space Utilization (All Users)" |
|  | du -sh /home/\* |
|  | else |
|  | echo "Home Space Utilization ($USER)" |
|  | du -sh $HOME |
|  | fi |
|  | exit |
|  | fi |
|  | else |
|  | echo "Invalid entry." >&2 |
|  | exit 1 |
|  | fi |

## 42. **while-menu.sh**

|  |
| --- |
| #!/bin/bash |
|  | # while-menu: a menu driven system information program |
|  | DELAY=1 # Number of seconds to display results |
|  | while true; do |
|  | clear |
|  | cat << EOF |
|  | Please Select: |
|  | 1. Display System Information |
|  | 2. Display Disk Space |
|  | 3. Display Home Space Utilization |
|  | 0. Quit |
|  | EOF |
|  | read -p "Enter selection [0-3] > " |
|  | case "$REPLY" in |
|  | 0) |
|  | break |
|  | ;; |
|  | 1) |
|  | echo "Hostname: $HOSTNAME" |
|  | uptime |
|  | ;; |
|  | 2) |
|  | df -h |
|  | ;; |
|  | 3) |
|  | if [[ $(id -u) -eq 0 ]]; then |
|  | echo "Home Space Utilization (All Users)" |
|  | du -sh /home/\* |
|  | else |
|  | echo "Home Space Utilization ($USER)" |
|  | du -sh $HOME |
|  | fi |
|  | ;; |
|  | \*) |
|  | echo "Invalid entry." |
|  | ;; |
|  | esac |
|  | sleep "$DELAY" |
|  | done |
|  | echo "Program terminated |

## 43. **affect.sh**

|  |
| --- |
| #!/bin/bash |
|  |  |
|  | arr=('-' '\' '|' '/') |
|  | while true; do |
|  | for c in "${arr[@]}"; do |
|  | printf "\r %c " $c |
|  | sleep .5 |
|  | done |
|  | done |

## 44. **color.sh**

|  |
| --- |
| #!/bin/bash |
|  |  |
|  | DARKGRAY='\033[1;30m' |
|  | RED='\033[0;31m' |
|  | LIGHTRED='\033[1;31m' |
|  | GREEN='\033[0;32m' |
|  | YELLOW='\033[1;33m' |
|  | BLUE='\033[0;34m' |
|  | PURPLE='\033[0;35m' |
|  | LIGHTPURPLE='\033[1;35m' |
|  | CYAN='\033[0;36m' |
|  | WHITE='\033[1;37m' |
|  | DEFAULT='\033[0m' |
|  |  |
|  | COLORS=($DARKGRAY $RED $LIGHTRED $GREEN $YELLOW $BLUE $PURPLE $LIGHTPURPLE $CYAN $WHITE ) |
|  |  |
|  | for c in "${COLORS[@]}";do |
|  | printf "\r $c LOVE $DEFAULT " |
|  | sleep 1 |
|  | done |

## 45. **convertlowercase.sh**

|  |
| --- |
| #!/bin/bash |
|  |  |
|  | echo -n "Enter File Name : " |
|  | read fileName |
|  |  |
|  | if [ ! -f $fileName ]; then |
|  | echo "Filename $fileName does not exists" |
|  | exit 1 |
|  | fi |
|  |  |
|  | tr '[A-Z]' '[a-z]' <$fileName >>small.txt |

## 46. **up.sh**

#!/bin/bash

LEVEL=$1

for ((i = 0; i < LEVEL; i++)); do

echo $i

CDIR=../$CDIR

done

cd $CDIR

echo "You are in: "$PWD

sh=$(which $SHELL)

exec $sh

|  |
| --- |
|  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## 47. **list-dir.sh**

#!/bin/bash

list=($(ls))

for f in "${list[@]}";do

echo $f

done

## 48. **count-lines.sh**

#!/usr/bin/env bash

for F in \*

do

if [[ -f $F ]]

then

echo $F: $(cat $F | wc -l)

fi

done

49. Random.sh

#!/bin/bash

echo "Hello $USER"

echo "$(uptime)" >>"$(date)".txt

echo "Your File is being saved to $(pwd)"

## 50. **random-emoji.sh**

#!/bin/bash

while true; do

rand=$(shuf -i 2600-2700 -n 1)

echo -n -e ' \u'$rand

sleep 1

done

## 51. **directorysize.sh**

|  |
| --- |
| #!/bin/bash |
|  |  |
|  | echo " Enter your directory: " |
|  | read x |
|  | du -sh "$x" |

## 52. **test-file.sh**

|  |
| --- |
| #!/bin/bash |
|  | # test-file: Evaluate the status of a file |
|  | echo "Hey what's the File/Directory name (using the absolute path)?" |
|  | read FILE |
|  |  |
|  | if [ -e "$FILE" ]; then |
|  | if [ -f "$FILE" ]; then |
|  | echo "$FILE is a regular file." |
|  | fi |
|  | if [ -d "$FILE" ]; then |
|  | echo "$FILE is a directory." |
|  | fi |
|  | if [ -r "$FILE" ]; then |
|  | echo "$FILE is readable." |
|  | fi |
|  | if [ -w "$FILE" ]; then |
|  | echo "$FILE is writable." |
|  | fi |
|  | if [ -x "$FILE" ]; then |
|  | echo "$FILE is executable/searchable." |
|  | fi |
|  | else |
|  | echo "$FILE does not exist" |
|  | exit 1 |
|  | fi |
|  | exit |

## 53. **server-health.sh**

|  |
| --- |
| #!/bin/bash |
|  | date |
|  | echo "uptime:" |
|  | uptime |
|  | echo "Currently connected:" |
|  | w |
|  | echo "--------------------" |
|  | echo "Last logins:" |
|  | last -a | head -3 |
|  | echo "--------------------" |
|  | echo "Disk and memory usage:" |
|  | df -h | xargs | awk '{print "Free/total disk: " $11 " / " $9}' |
|  | free -m | xargs | awk '{print "Free/total memory: " $17 " / " $8 " MB"}' |
|  | echo "--------------------" |
|  | start\_log=$(head -1 /var/log/messages | cut -c 1-12) |
|  | oom=$(grep -ci kill /var/log/messages) |
|  | echo -n "OOM errors since $start\_log :" $oom |
|  | echo "" |
|  | echo "--------------------" |
|  | echo "Utilization and most expensive processes:" |
|  | top -b | head -3 |
|  | echo |
|  | top -b | head -10 | tail -4 |
|  | echo "--------------------" |
|  | echo "Open TCP ports:" |
|  | nmap -p -T4 127.0.0.1 |
|  | echo "--------------------" |
|  | echo "Current connections:" |
|  | ss -s |
|  | echo "--------------------" |
|  | echo "processes:" |
|  | ps auxf --width=200 |
|  | echo "--------------------" |
|  | echo "vmstat:" |
|  | vmstat 1 5 |

## 54. **cpu.sh**

|  |
| --- |
| MAX=95 |
|  | EMAIL=server@127.0.0.1 |
|  |  |
|  | USE=$(grep 'cpu ' /proc/stat | awk '{usage=($2+$4)\*100/($2+$4+$5)} END {print usage ""}') |
|  | if [ $USE -gt $MAX ]; then |
|  | echo "Percent used: $USE" | mail -s "Running out of CPU power" $EMAIL |
|  | fi |

## 55. **disk-space.sh**

|  |
| --- |
| MAX=95 |
|  | EMAIL=server@127.0.0.1 |
|  | PART=sda1 |
|  |  |
|  | USE=$(df -h | grep $PART | awk '{ print $5 }' | cut -d'%' -f1) |
|  | if [ $USE -gt $MAX ]; then |
|  | echo "Percent used: $USE" | mail -s "Running out of disk space" $EMAIL |
|  | fi |

## 56. **remotebackup.sh**

|  |
| --- |
| #!/bin/bash |
|  |  |
|  | rsync -avz -e "ssh " /path/to/yourfile user@backupserver.com:/backup/ |
|  | echo "backup for $(date) " | mail -s "backup complete" user@youremail.com |

## 55. **get-temperature.sh**

|  |
| --- |
| #!/usr/bin/env bash |
|  |  |
|  | TEMP\_FILE=/sys/class/thermal/thermal\_zone0/temp |
|  |  |
|  | ORIGINAL\_TEMP=$(cat $TEMP\_FILE) |
|  | TEMP\_C=$((ORIGINAL\_TEMP/1000)) |
|  | TEMP\_F=$(($TEMP\_C \* 9/5 + 32)) |
|  |  |
|  | echo $TEMP\_F F |

## 56. **addition.sh**

|  |
| --- |
| #!/bin/bash |
|  |  |
|  | echo 'Enter the First Number :' |
|  | read a |
|  | echo 'Enter the Second Number :' |
|  | read b |
|  | x=$(expr "$a" + "$b") |
|  | echo $a + $b = $x |

## 57. **substraction.sh**

|  |
| --- |
| #!/bin/bash |
|  | echo .Enter the First Number: . |
|  | read a |
|  | echo .Enter the Second Number: . |
|  | read b |
|  | x=$(($a - $b)) |
|  | echo $a - $b = $x |

## 58. **multiplication.sh**

|  |
| --- |
| #!/bin/bash |
|  | echo .Enter the First Number: . |
|  | read a |
|  | echo .Enter the Second Number: . |
|  | read b |
|  | echo "$a \* $b = $(expr $a \\* $b)" |

## 59. **division.sh**

|  |
| --- |
| #!/bin/bash |
|  | echo .Enter the First Number: . |
|  | read a |
|  | echo .Enter the Second Number: . |
|  | read b |
|  | echo "$a / $b = $(expr $a / $b)" |

## 60. **simplecalc.sh**

|  |
| --- |
| #! /bin/bash |
|  | clear |
|  | sum=0 |
|  | i="y" |
|  |  |
|  | echo " Enter one no." |
|  | read n1 |
|  | echo "Enter second no." |
|  | read n2 |
|  | while [ $i = "y" ]; do |
|  | echo "1.Addition" |
|  | echo "2.Subtraction" |
|  | echo "3.Multiplication" |
|  | echo "4.Division" |
|  | echo "Enter your choice" |
|  | read ch |
|  | case $ch in |
|  | 1) |
|  | sum=$(expr $n1 + $n2) |
|  | echo "Sum ="$sum |
|  | ;; |
|  | 2) |
|  | sum=$(expr $n1 - $n2) |
|  | echo "Sub = "$sum |
|  | ;; |
|  | 3) |
|  | sum=$(expr $n1 \\* $n2) |
|  | echo "Mul = "$sum |
|  | ;; |
|  | 4) |
|  | sum=$(expr $n1 / $n2) |
|  | echo "Div = "$sum |
|  | ;; |
|  | \*) echo "Invalid choice" ;; |
|  | esac |
|  | echo "Do u want to continue (y/n)) ?" |
|  | read i |
|  | if [ $i != "y" ]; then |
|  | exit |
|  | fi |
|  | done |

## 61. **table.sh**

|  |
| --- |
| #!/bin/bash |
|  | echo .Enter The Number upto which you want to Print Table: . |
|  | read n |
|  | i=1 |
|  | while [ $i -ne 10 ]; do |
|  | i=$(expr $i + 1) |
|  | table=$(expr $i \\* $n) |
|  | echo $table |
|  | done |

62.**evenodd.sh**

|  |
| --- |
| #!/bin/bash |
|  | echo "Enter The Number" |
|  | read n |
|  | num=$(expr $n % 2) |
|  | if [ $num -eq 0 ]; then |
|  | echo "is a Even Number" |
|  | else |
|  | echo "is a Odd Number" |
|  | fi |

## 63. **factorial.sh**

|  |
| --- |
| #!/bin/bash |
|  | echo "Enter The Number" |
|  | read a |
|  | fact=1 |
|  | while [ $a -ne 0 ]; do |
|  | fact=$(expr $fact \\* $a) |
|  | a=$(expr $a - 1) |
|  | done |
|  | echo $fact |

## 64. **armstrong.sh**

|  |
| --- |
| #!/bin/bash |
|  | echo "Enter A Number" |
|  | read n |
|  | arm=0 |
|  | temp=$n |
|  | while [ $n -ne 0 ]; do |
|  | r=$(expr $n % 10) |
|  | arm=$(expr $arm + $r \\* $r \\* $r) |
|  | n=$(expr $n / 10) |
|  | done |
|  | echo $arm |
|  | if [ $arm -eq $temp ]; then |
|  | echo "Armstrong" |
|  | else |
|  | echo "Not Armstrong" |
|  | fi |

## 65. **prime.sh**

|  |
| --- |
| #!/bin/bash |
|  | echo “Enter Any Number” |
|  | read n |
|  | i=1 |
|  | c=1 |
|  | while [ $i -le $n ]; do |
|  | i=$(expr $i + 1) |
|  | r=$(expr $n % $i) |
|  | if [ $r -eq 0 ]; then |
|  | c=$(expr $c + 1) |
|  | fi |
|  | done |
|  | if [ $c -eq 2 ]; then |
|  | echo “Prime” |
|  | else |
|  | echo “Not Prime” |
|  | fi |

## 66. **fibonacci.sh**

|  |
| --- |
| !/bin/bash |
|  | x=0 |
|  | y=1 |
|  | i=2 |
|  | while true ; do |
|  | i=$(expr $i + 1) |
|  | z=$(expr $x + $y) |
|  | echo -n "$z " |
|  | x=$y |
|  | y=$z |
|  | sleep .5 |
|  | done |

## 67. **decimal2binary.sh**

|  |
| --- |
| #!/bin/bash |
|  |  |
|  | for ((i = 32; i >= 0; i--)); do |
|  | r=$((2 \*\* $i)) |
|  | Probablity+=($r) |
|  | done |
|  |  |
|  | [[ $# -eq 0 ]] && { |
|  | echo -e "Usage \n \t $0 numbers" |
|  | exit 1 |
|  | } |
|  |  |
|  | echo -en "Decimal\t\tBinary\n" |
|  | for input\_int in $@; do |
|  | s=0 |
|  | test ${#input\_int} -gt 11 && { |
|  | echo "Support Upto 10 Digit number :: skiping \"$input\_int\"" |
|  | continue |
|  | } |
|  |  |
|  | printf "%-10s\t" "$input\_int" |
|  |  |
|  | for n in ${Probablity[@]}; do |
|  |  |
|  | if [[ $input\_int -lt ${n} ]]; then |
|  | [[ $s == 1 ]] && printf "%d" 0 |
|  | else |
|  | printf "%d" 1 |
|  | s=1 |
|  | input\_int=$(($input\_int - ${n})) |
|  | fi |
|  | done |
|  | echo -e |
|  | done |

## 68. **binary2decimal.sh**

|  |
| --- |
| #!/bin/bash |
|  | echo "Enter a number :" | |
|  | read Binary | |
|  | if [ $Binary -eq 0 ]; then | |
|  | echo "Enter a valid number " | |
|  | return | |
|  | else | |
|  | while [ $Binary -ne 0 ]; do | |
|  | Bnumber=$Binary | |
|  | Decimal=0 | |
|  | power=1 | |
|  | while [ $Binary -ne 0 ]; do | |
|  | rem=$(expr $Binary % 10) | |
|  | Decimal=$((Decimal + (rem \* power))) | |
|  | power=$((power \* 2)) | |
|  | Binary=$(expr $Binary / 10) | |
|  | done | |
|  | echo " $Decimal" | |
|  | done | |
|  | fi | |
|  | |  |

## 69. **dec2hex.sh**

|  |
| --- |
| #!/bin/bash |
|  |  |
|  | printf "0x%x\n" $1 |

## 70. **hextodec.sh**

|  |
| --- |
| #!/bin/bash |
|  |  |
|  | printf "%d \n " $1 |