***­ AYUSH JINDAL***

***RA1911003010308***

***E1***

***Ex. No. 6***

***Date: 12/03/21 STUDY OF SHELL SCRIPTS***

***Aim:***

*To study about the types of shell scripting, execution procedure of shell*

*programs and various syntax on conditional branching and looping statements.*

***Steps to write Shell Script in Linux/Unix:***

1. *Create a file using a vi editor (or any other editor). Name script file with*
2. *extension .sh*
3. *Start the script with #! /bin/bash*
4. *Write some Program.*
5. *Save the script file as filename.sh*
6. *For executing the script type ./filename.sh*

***Syntax for if statement***

*if [ condition ]*

*then*

*...*

*elif [ condition ]*

*then*

*...*

*else*

*...*

*fi*

***Syntax for case structure***

*case value in*

*pat1) ...*

*statement;;*

*pat2) ...*

*statement;;*

*\*) ...*

*statement;;*

*Esac*

***Syntax for for-loop***

***for*** *var* ***in*** *list-of-values*

*do*

*...*

*...*

*done*

***Syntax for While loop***

*while [ condition ]*

*do*

*...*

*...*

*done*

***Syntax for until loop***

*until [ condition ]*

*do*

*...*

*...*

*done*

***Syntax for printf statement***

*printf “string and format” arg1 arg2 … …*

* *Break and continue statements functions similar to C programming*
* *Relational operators are –lt, -le, -gt, -ge, -eq,-ne*
* *Ex. (i>= 10) is written as [ $i -ge 10 ]*
* *Logical operators (and, or, not) are -o, -a, !*
* *Ex. (a>b) && (a>c) is written as [ $a –gt $b –a $a –gt $c ]*
* *Two strings can be compared using = operator*

***Result:***

*Learned various syntax and commands of Linux Shell Script successfully.*

***AYUSH JINDAL***

***RA1911003010308***

***E1***

***Ex. No. 7***

***Date: 12/03/21 SHELL PROGRAMMING***

***1) a) Present working Directory has list of files and directories. Write a shell***

***script to list only the name of sub directories in the present working***

***Directory – Simple if .. fi statement.***

***Aim:***

*To write a Bash Shell Script to list only the name of sub directories in the*

*present working Directory using Simple if .. fi statement*

***Procedure:***

***Step 1:*** *Read the file name from the user using read command*

***Step 2:*** *Use if .. else conditional branching check for Execute permission for*

*the given file name*

*if [ expression ]*

*then*

*statement1*

*fi*

***Step 3:*** *if [ -x filename ] then print file has Execute permission*

*else*

*print file has no Execute permission*

***Step 4:*** *use the chmod command to set execute permission for the inputted file*

*which has no execute permission*

***Program:***

*for i in \**

*do*

*if [ -d $i ]*

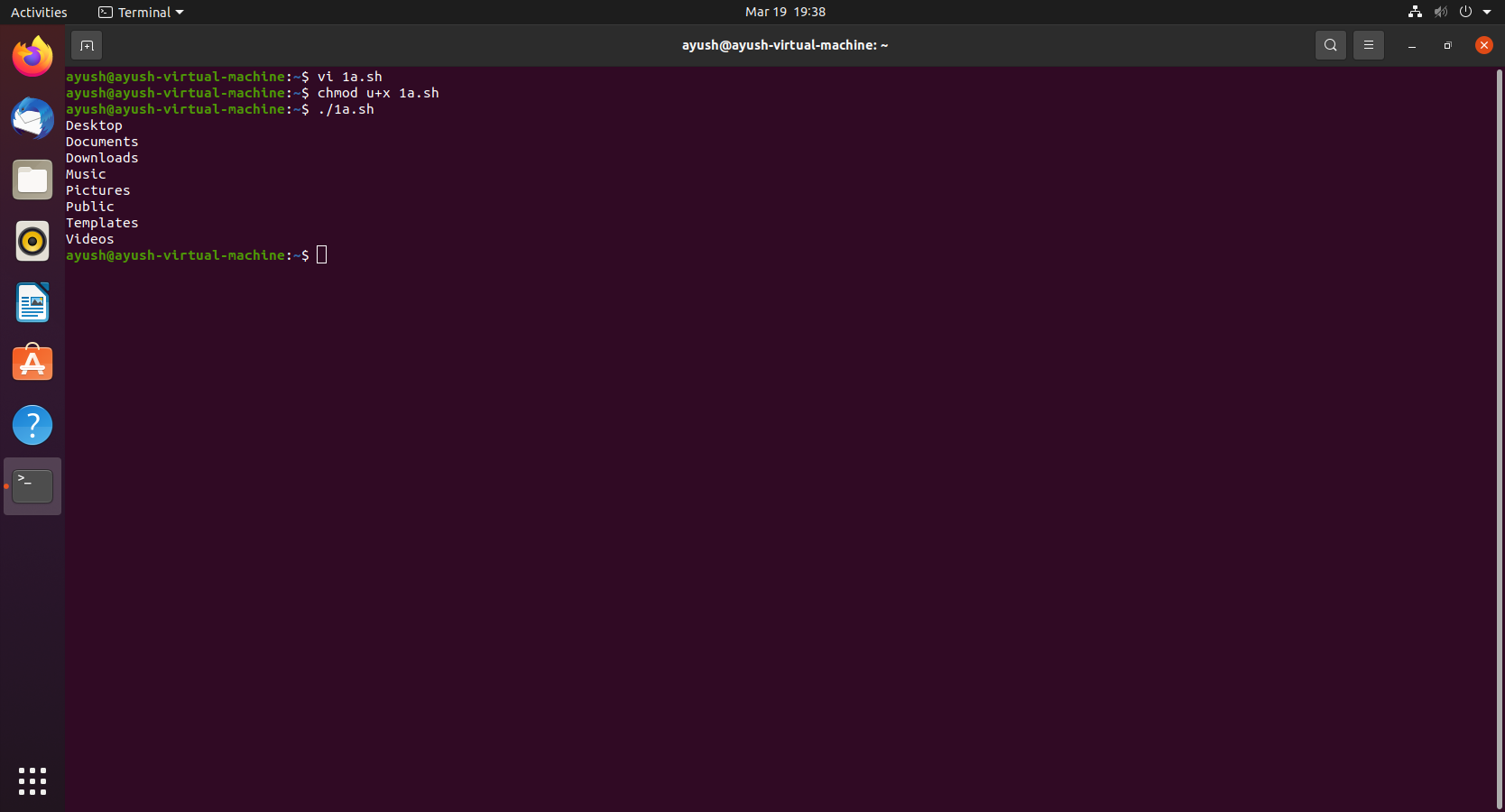
*then*

*echo $i*

*fi*

*done*

***Output:***

**

***Result:***

*Learned and practiced how to write bash script using simple if .. fi statement in*

*Linux.*

***1) b) Every directory and file created have a permission for directory, user***

***, group and others. Every user, group and others have all the three***

***permissions as read, write and execute. Write a bash shell script to read a***

***filename from the shell and check whether the file has execute permission***

***or not. If not, add the permission – if..else..fi statement.***

***Aim:***

*To write a Bash Shell Script to check whether the file has execute permission*

*or not. If not, add the permission using if .. else statement*

***Procedure:***

***Step 1:*** *Read the file name from the user using read command*

***Step 2:*** *Use if .. else conditional branching check for Execute permission for*

*the given file name*

*if [ expression ]*

*then*

*statement1*

*else*

*statement2*

*fi*

***Step 3:*** *if [ -x filename ] then print file has Execute permission*

*else*

*print file has no Execute permission*

***Step 4:*** *use the chmod command to set execute permission for the inputted file*

*which has no execute permission*

***Program:***

*for name in \**

*do*

*if [ -x $name ]*

*then*

*echo "$name has execute permission"*

*else*

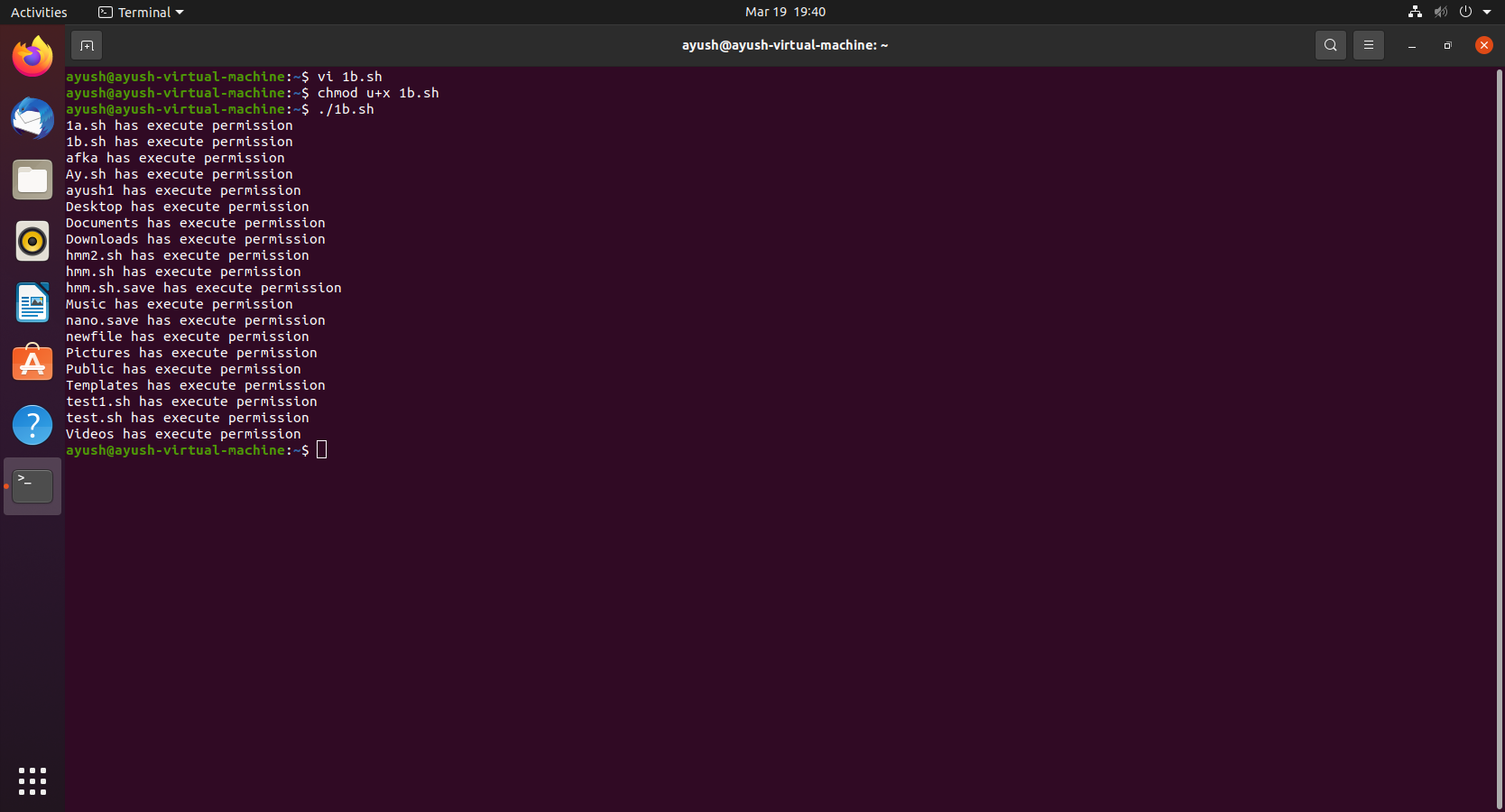
*echo "$name has no execute permission"*

*chmod +x $name*

*fi*

*done*

***Output:***

******

***Result:***

*Learned and practiced how to write bash script using if .. else statement in*

*Linux.*

***1) c)***

***Write a shell script to print a greeting as specified below.***

* ***If hour is greater than or equal to 0 (midnight) and less than or equal to 11 (up to 11:59:59), "Good morning" is displayed.***
* ***If hour is greater than or equal to 12 (noon) and less than or equal to 17 (up to 5:59:59 p.m.), "Good afternoon" is displayed.***
* ***If neither of the preceding two conditions is satisfied, "Good evening" is displayed.***

***Use if..elif..else.. fi statement***

***Aim:***

*To write a Bash Shell Script to print a greeting message using Simple if .. elif..*

*else .. fi statement*

***Procedure:***

***Step 1:*** *Read the file name from the user using read command*

***Step 2:*** *Use if .. elif.. else conditional branching statement, check for hour by*

*extracting the hour details from “date” command using “cut” command for the*

*given file name*

*if [ expression1 ]*

*then*

*statement1*

*statement2*

*.*

*.*

*elif [ expression2 ]*

*then*

*statement3*

*statement4*

*.*

*.*

*else*

*statement5*

*fi*

***Step 3:*** *Use “date” command and “cut” command*

***Program:***

*time=date | cut -d " " -f5 | cut -d ":" -f1-1*

*if [[ $date -ge 12 ]]; then*

*echo "Good Morning"*

*elif [[ $date -ge 18 ]]; then*

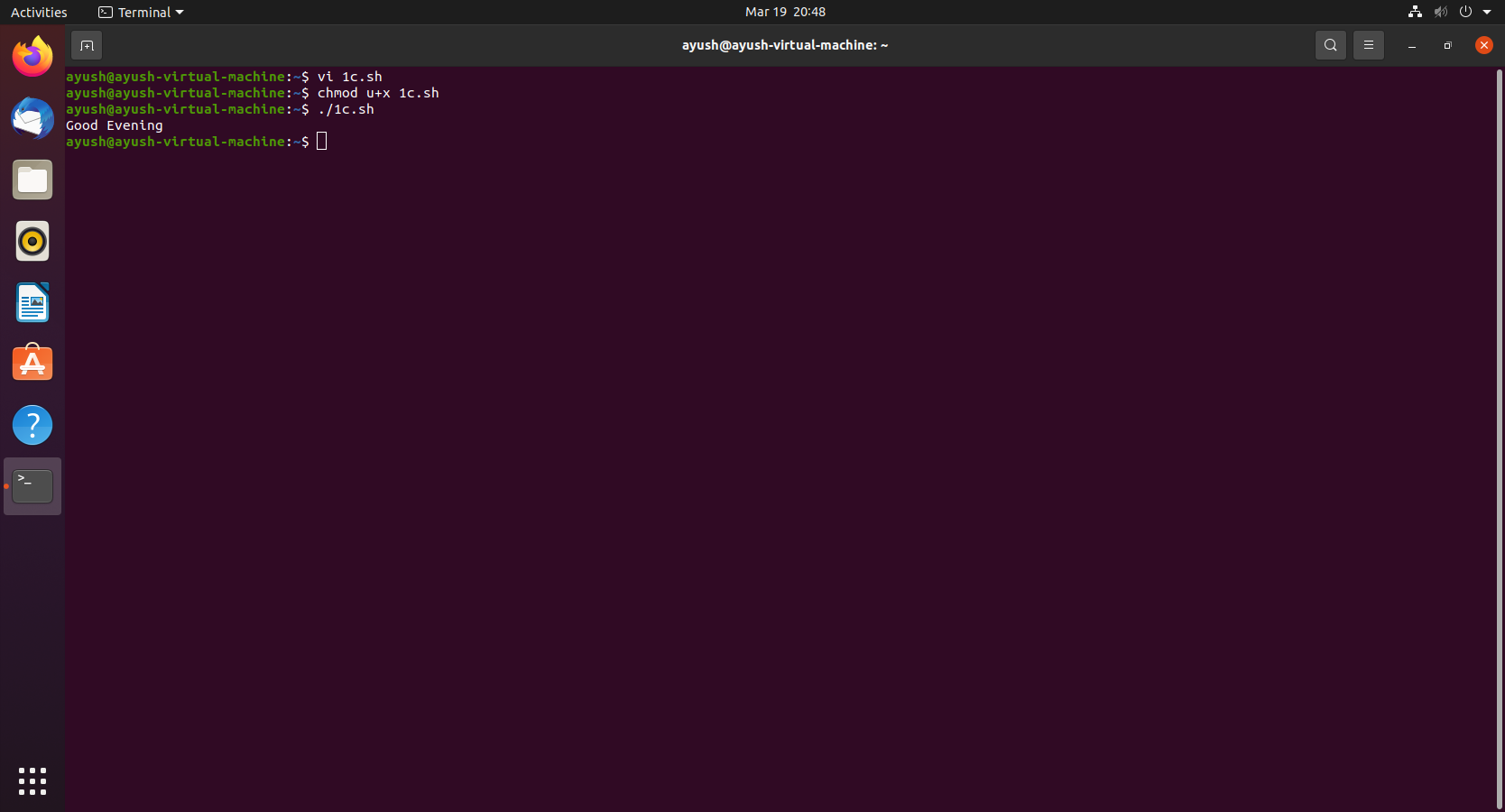
*echo "Good Afternoon"*

*else*

*echo "Good Evening"*

*fi*

***Output:***

******

***Result:***

*Learned and practiced how to write bash script using If ..elif.. else..fi statement*

*in Linux.*

***1) d)***

***Write a shell script to simulate a online personal assistant for banking as***

***specified below.***

***Use the following Program keywords to get online assistance from bank***

***1) If keyword is loan then ask for “Type of Loan”***

***2) If keyword is personalloan or personal\_loan or pl then print reply as***

***“Currently no offer for personal loan”***

***3) If keyword is carloan or car\_loan then print reply as “Currently best***

***offer of 7 % is going on!”***

***4) If keyword is home or housingloan or housing\_loan or homeloan or***

***home\_loan then print reply as “Currently best offer of 8 % is going***

***on!”***

***5) If keyword is card or credit or creditcard or debit or debitcard then print***

***reply as “Contact nearest branch!”***

***6) If keyword is bye print reply as “See you later!”***

***7) If keyword is other than above keywords then print reply as “Sorry, I***

***don't understand!”***

***Use case .. esac statement to implement the script***

***Aim:***

*To write a Bash Shell Script to simulate a online personal assistant for banking*

*using case .. esac statement.*

***Procedure:***

***Step 1:*** *Read the input keyword string*

*The keywords can be loan, credit, personalloan, homeloan, carloan,*

*debit*

***Step 2:*** *Use case … esac statement to get multiple options*

*case in*

*Pattern 1) Statement 1;;*

*Pattern n) Statement n;;*

*esac*

***Step 3:*** *Print the necessary output based on given inputs*

***Program:***

*echo "How may I help you,Please write your query :- "*

*while :*

*do*

*read query*

*case "$query" in*

*"loan") echo -e "Type of Loan.\nDo you have any other query Please write below :-";;*

*"personalloan"|"personal\_loan"|"pl") echo -e "Currently no offer for personal loan.\nDo you have any other query Please write below :-";;*

*"carloan"|"car\_loan") echo -e "Currently best offer of 7 % is going on!\nDo you have any other query Please write below :-";;*

*"home"|"housingloan"|"housing\_loan"|"homeloan"|"home\_loan") echo -e "Currently best offer of 8 % is going on!\nDo you have any other query Please write below :-";;*

*"card"|"credit"|"creditcard"|"debit"|"debitcard") echo -e "Contact nearest branch!\nDo you have any other query Please write below :-";;*

*"bye") echo "See you later!"*

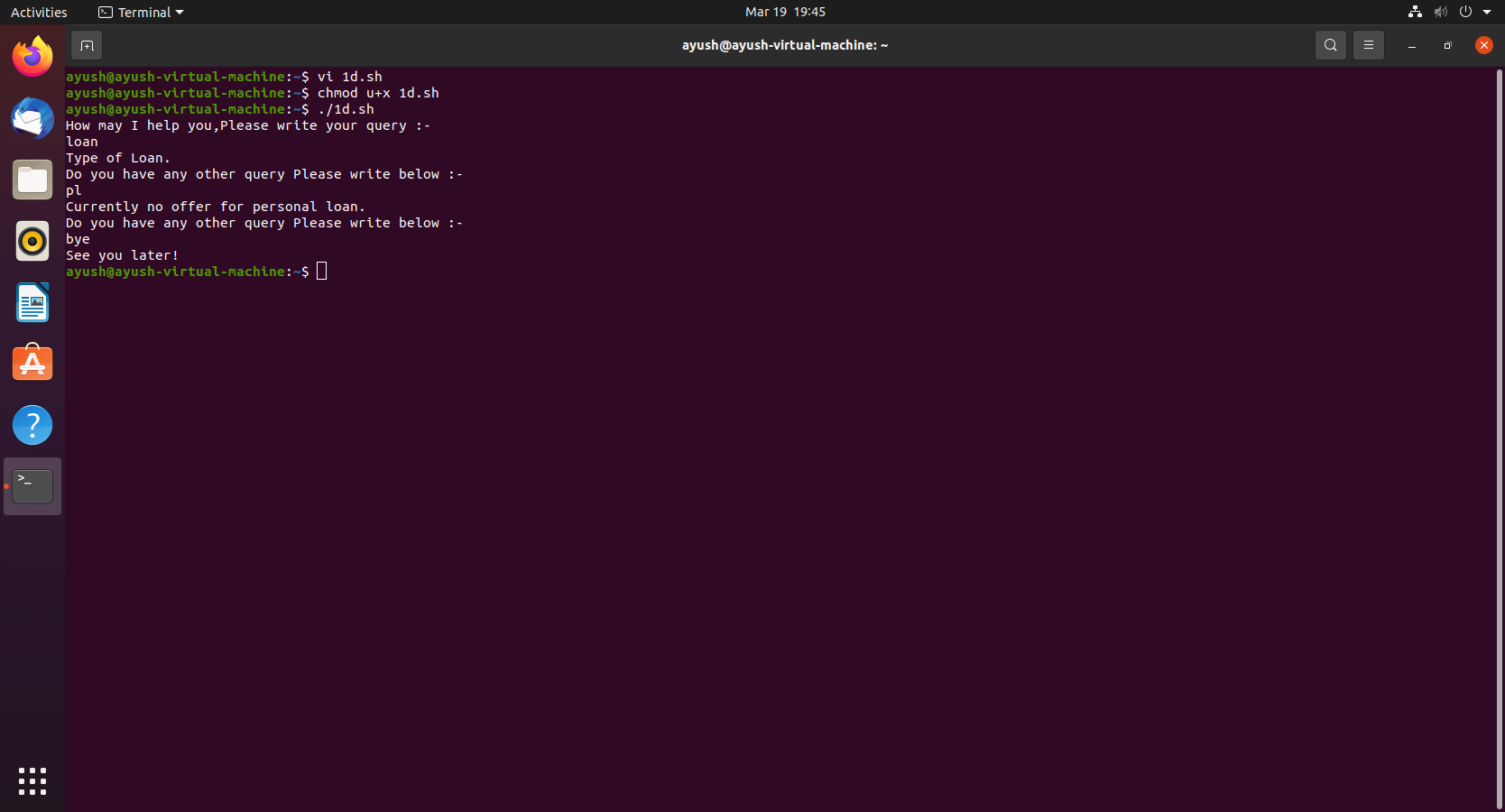
*break;;*

*\*) echo -e "Sorry, I don't understand!\nPlease write your query again :-";;*

*esac*

*done*

***Output:***

**

***Result:***

*Learned and practiced how to write bash script to simulate online banking*

*assistance using case .. esac statement in Linux.*

***2) a) Bash Shell Script to find whether the given number is Armstrong***

***number - While Loop***

***Aim:***

*To write a Bash Shell Script to find whether the given number is Armstrong*

*number or not using While Loop.*

***Procedure:***

*Step 1: Start the program with Vi Editor*

*Step 2: Read the input from the user*

*Step 3: Initialize the variables sum with 0 and temp with the given number*

*Step 4: Use WHILE loop to find the Armstrong number*

*WHILE Loop Syntax:*

*while [ condition]*

*do*

*...*

*...*

*done*

*Step 5: Find the remainder using Modulus operator*

*Step 6: Calculate the sum as*

*sum=sum+(reminder\*reminder\*reminder)*

*Step 7: Divide the number by 10*

*Step 8: Repeat steps 4 to 7 until number > 0*

*Step 9: Compare the sum and temp variables values*

*If values are same then*

*display number is Armstrong*

*else*

*display number is not Armstrong*

***Program:***

*echo "Enter input to check if its an armstrong number or not:-"*

*sum=0*

*temp=0*

*read number*

*n=$number*

*while (( $number > 0 ))*

*do*

*temp=$((number%10))*

*number=$((number/10))*

*temp=$((temp\*temp\*temp))*

*sum=$((sum + temp))*

*done*

*if (($sum == $n))*

*then*

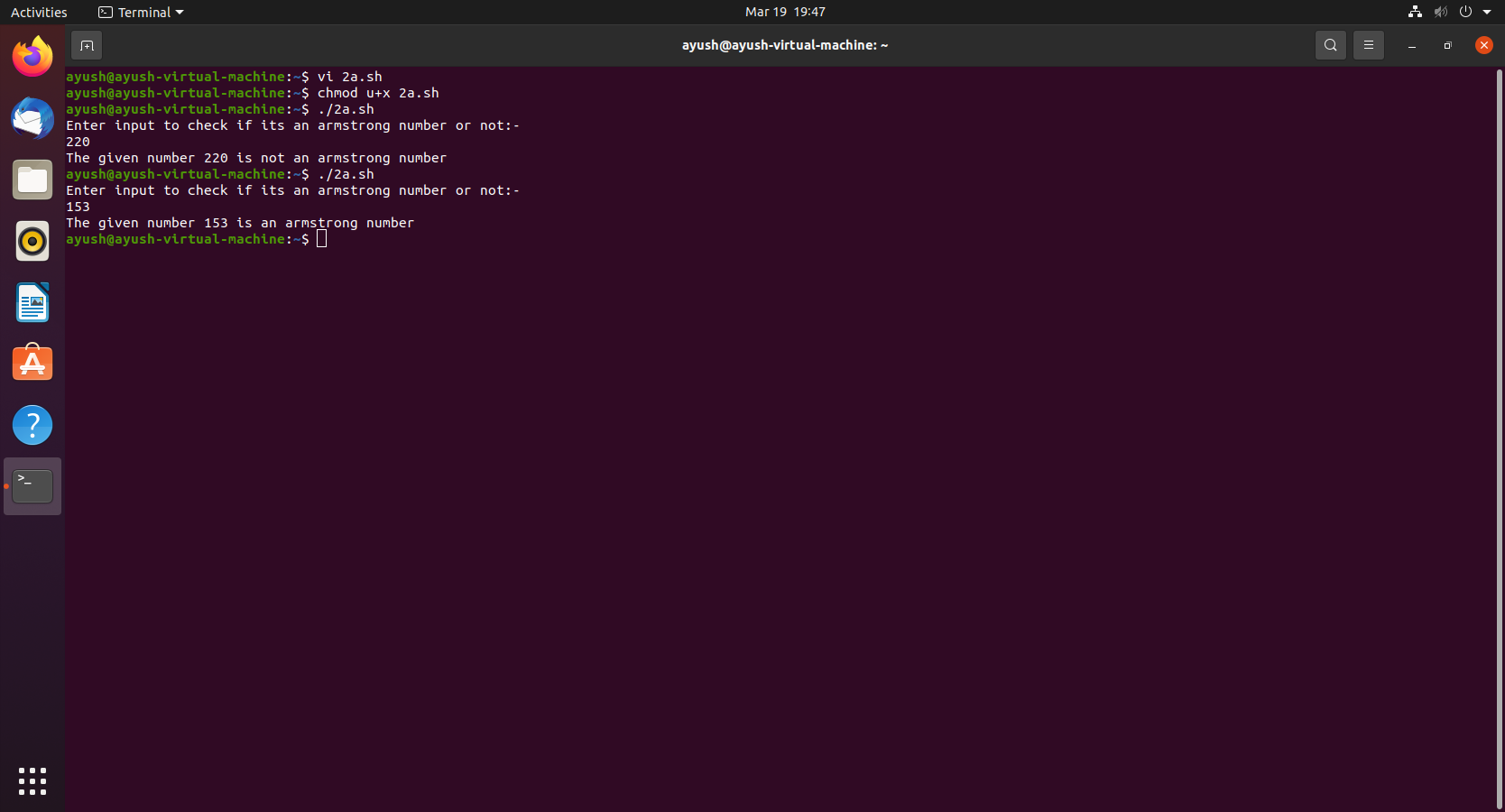
*echo "The given number $n is an armstrong number"*

*else*

*echo "The given number $n is not an armstrong number"*

*fi*

***Output:***

******

***Result:***

*Learned and practiced how to write bash script using WHILE loop in Linux.*

***2) b) Bash Shell Script to display the weekdays and weekends - FOR loop***

***Aim:***

*To write a Bash Shell Script to display the weekdays and weekends using For*

*Loop*

***Procedure:***

***Step 1****: Read the number from the user*

***Step 2****: Initialize the variable temp to 1*

***Step 3****: Use FOR loop to display the week day and week end*

*FOR Loop Syntax:*

*for var in list-of-values*

*do*

*...*

*...*

*done*

***Step 4****: Using Conditional branching statement check the day falls in*

*weekend or weekday*

***Step 5****: Use if [ $temp -eq 7 -o $temp -eq 8 ] For displaying Weekend*

*Else display the day number and name as weekday*

***Program:***

*echo "Please enter a number between 1-7"*

*read number*

*temp=1*

*day=0*

*temp=$number*

*for i in {1..7}*

*do*

*if (($i == 1))*

*then*

*echo "Day 1 : Mon (weekday)"*

*if (($number == $i))*

*then*

*day="Mon"*

*fi*

*elif (($i == 2))*

*then*

*echo "Day 2 : Tue (weekday)"*

*if (($number == $i))*

*then*

*day="Tue"*

*fi*

*elif (($i == 3))*

*then*

*echo "Day 3 : Wed (weekday)"*

*if (($number == $i))*

*then*

*day="Wed"*

*fi*

*elif (($i == 4))*

*then*

*echo "Day 4 : Thu (weekday)"*

*if (($number == $i))*

*then*

*day="Thu"*

*fi*

*elif (($i == 5))*

*then*

*echo "Day 5 : Fri (weekday)"*

*if (($number == $i))*

*then*

*day="Fri"*

*fi*

*elif (($i == 6))*

*then*

*echo "Day 6 : Sat (WEEKEND)"*

*if (($number == $i))*

*then*

*day="Sat"*

*fi*

*elif (($i == 7))*

*then*

*echo "Day 7 : Sun (WEEKEND)"*

*if (($number == $i))*

*then*

*day="Sun"*

*fi*

*fi*

*done*

*echo "---------------------"*

*if (($temp == 6 || $temp == 7));*

*then*

*echo "Day $number : $day is a WEEKEND"*

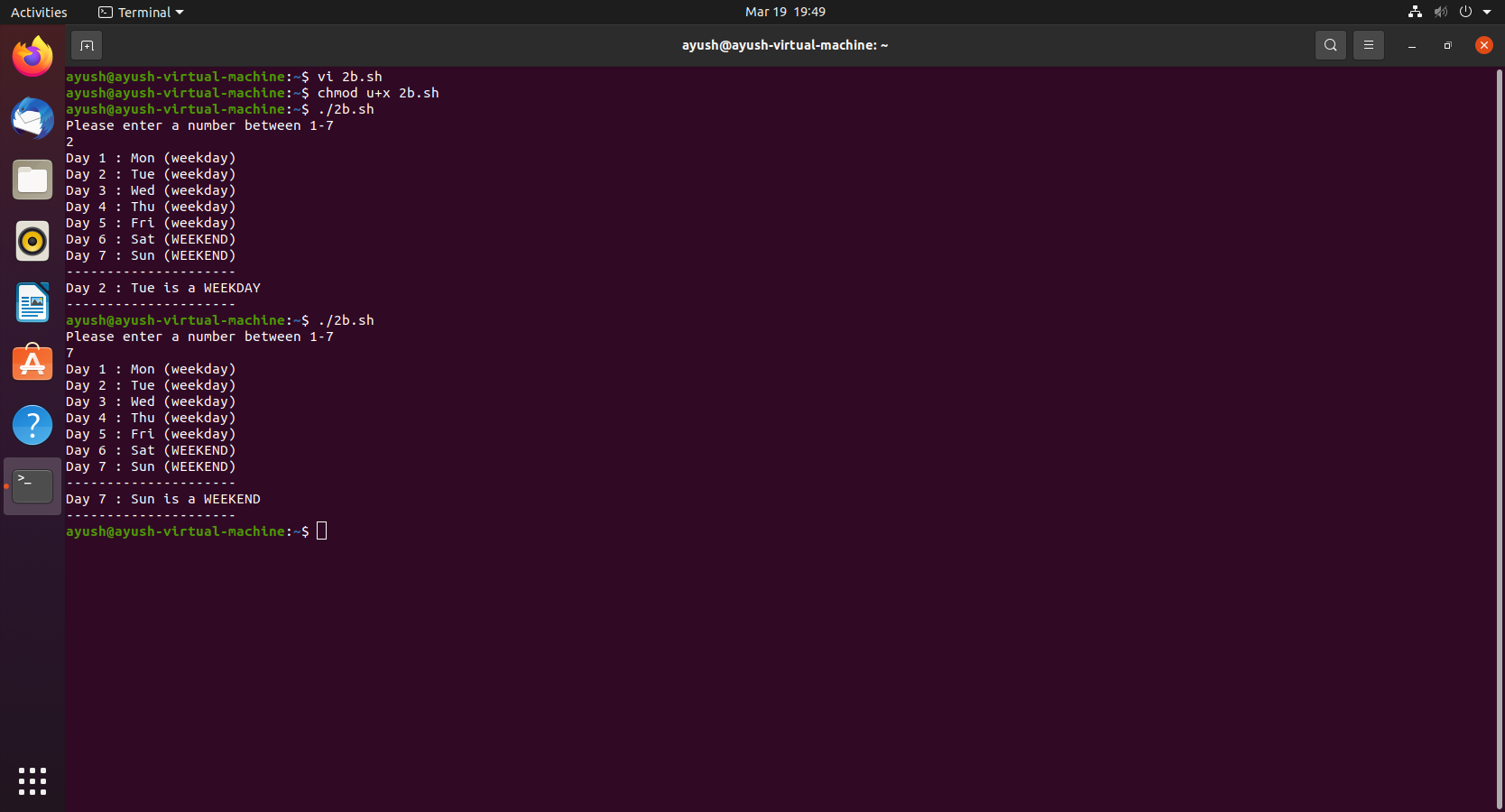
*else*

*echo "Day $number : $day is a WEEKDAY"*

*fi*

*echo "---------------------"*

***Output:***

******

***Result:***

*Learned and practiced how to write bash script using FOR loop in Linux.*

***2) c) Bash Shell Script to display the perfect numbers from a given range***

***of numbers - UNTIL loop***

***Aim:***

*To write a Bash Shell Script to display perfect numbers from a given range of*

*numbers using UNTIL Loop statements*

***Procedure:***

***Step 1:*** *Read the starting and ending range of numbers from the user*

***Step 2:*** *Use Outer UNTIL loop for iterating from starting till ending*

*range of numbers*

*until [ condition ]*

*do*

*...*

*...*

*done*

***Step 3:*** *Initialize the “sum” variable by 0*

***Step 4:*** *Iterate the” j” loop from 1 to” i “.*

***Step 5:*** *Divide” i “by” j “*

***Step 6:*** *If the remainder is 0, add the value of” j” to “sum” variable.*

***Step 7:*** *If the “sum” is equal to “i “, Number is perfect*

*else number is not perfect.*

***Step 8:*** *Repeat the steps from Step 2 to Step 7*

***Program:***

*echo "Enter a starting value:-"*

*read s*

*echo "Enter a ending value:-"*

*read e*

*echo "Perfect numbers are:"*

*until [ $s -gt $e ]*

*do*

*sum=0*

*j=1*

*until [ $j -ge $s ]*

*do*

*if (($s % $j == 0))*

*then*

*sum=$((sum + j))*

*fi*

*j=$(( $j+1 ))*

*done*

*if (($sum == $s))*

*then*

*echo -ne "$s\t"*

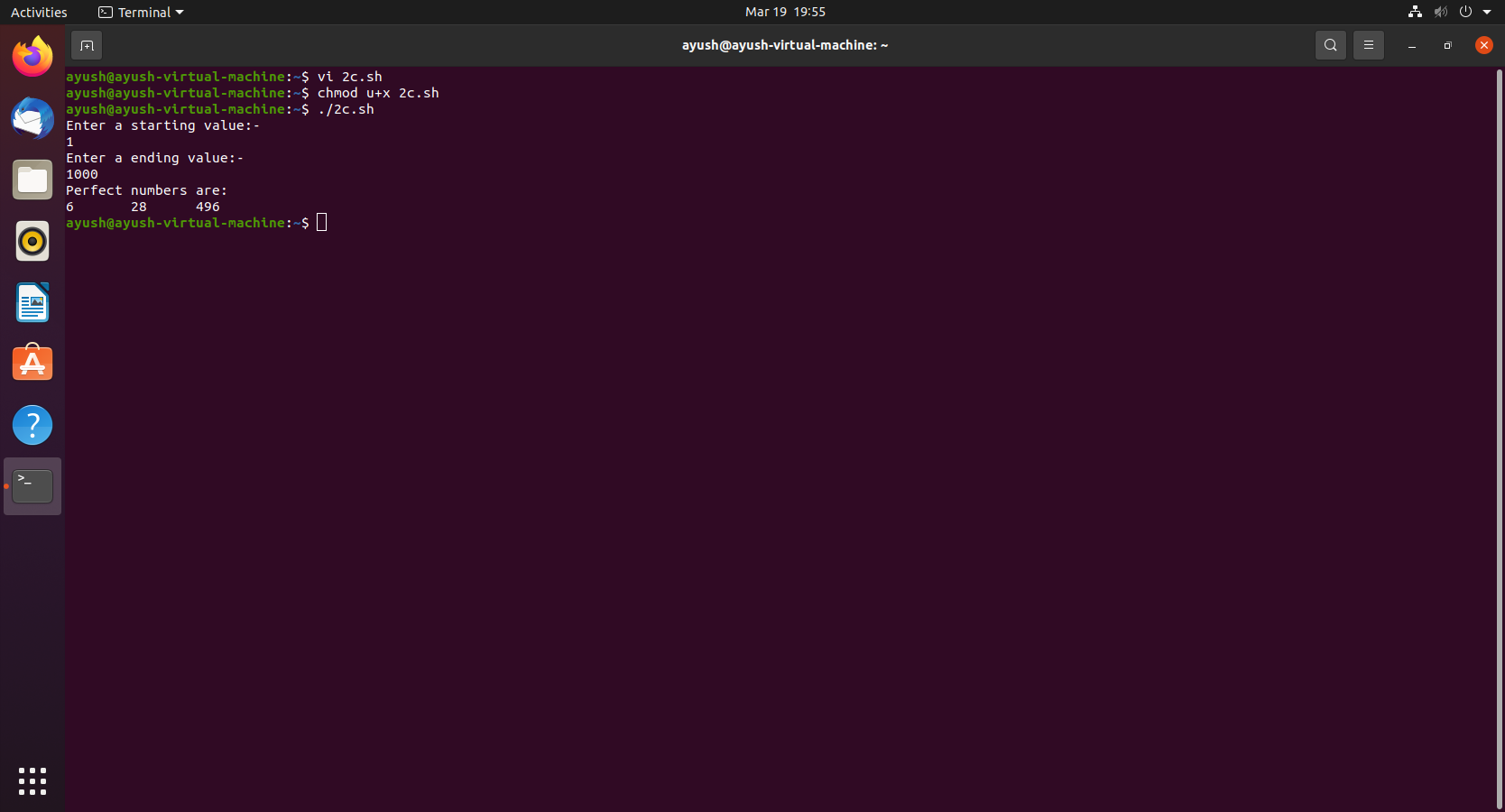
*fi*

*s=$(( s+1 ))*

*done*

*echo*

***Output:***

******

***Result:***

*Learned and practiced how to write bash script using UNTIL loop in Linux.*