به نام ایزد یکتا

گزارش سوم آزمایشگاه سیستم‌عامل

­­

استاد: مهندس قاسمی

تهیه کننده: بردیا اردکانیان

۹۸۳۱۰۷۲

آزمایش اول:

آبتدا یک فایل با پسوند .sh می‌سازیم، کد مورد نظر را می‌نویسیم و با کمک دستور آن را اجرا می‌کنیم.

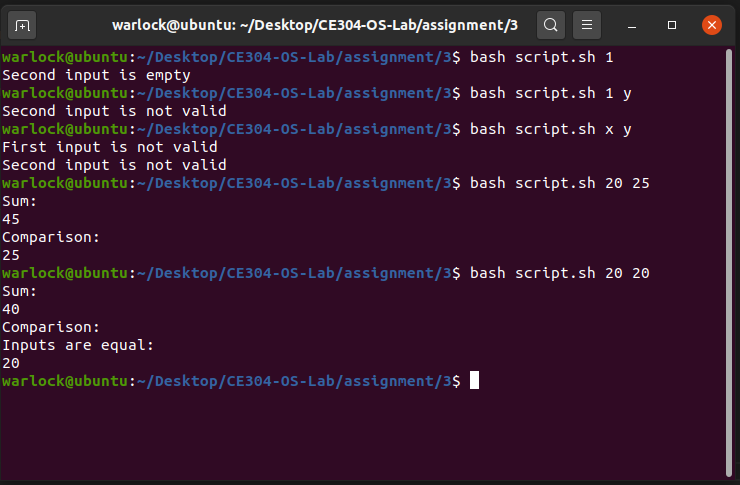
برنامه ما دو ارگومان را به عنوان ورودی دریافت می‌کند و با برسی صحت آرگومان‌ها به بخش بعدی برنامه وارد می‌شود. در صورتی که ارگومان‌های ورودی واجد شرایط نباشند خطایی را با کمک دستور در ترمینال چاپ می‌کند.

در صورت صحت ارگومان‌های ورودی جمع آنها را حساب می‌کند و به کمک دستور به نمایش می‌گذارد.

و در نهایت با انجام مقایسه ساده‌ای عدد بزرگ‌تر را پیدا کرده و آن را در ترمینال به نمایش می‌گذارد.

#!/bin/bash  
  
flag=0  
function **check**(){  
if [ -z $1 ]; then  
    echo "First input is empty"  
    flag=1  
elif ! [[ $1 =~ ^[+-]?[0-9]+$ ]]; then  
    echo "First input is not valid"  
    flag=1  
fi  
if [ -z $2 ]; then  
    echo "Second input is empty"  
    flag=1  
elif  ! [[ $2 =~ ^[+-]?[0-9]+$ ]]; then  
    echo "Second input is not valid"  
    flag=1  
fi  
}  
  
check $1 $2  
if [ $flag -eq 0 ]; then  
    echo "Sum:"  
    expr $1 + $2  
  
    echo "Comparison:"  
    if [ $1 -gt $2 ]  
    then  
        echo $1  
    elif [ $1 -lt $2 ]  
    then  
        echo $2  
    elif [ $1 -eq $2 ]  
    then  
        echo "Inputs are equal:"  
        echo $2  
    fi  
fi

*کد 1-1*

**

*عکس 1-1*

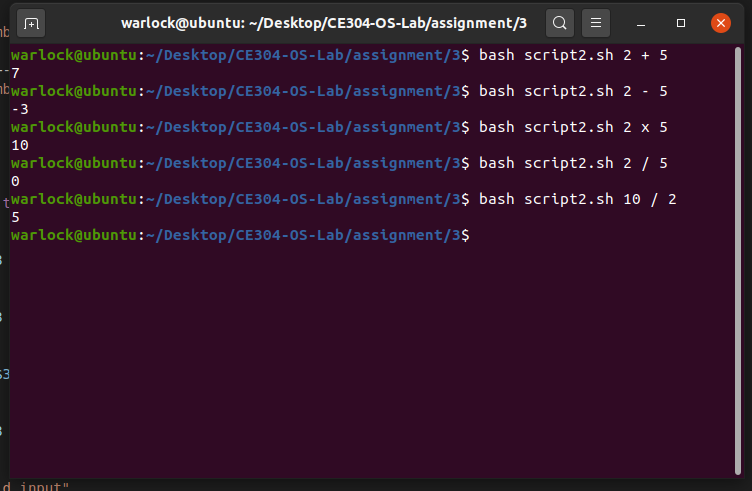
*آزمایش دوم:*

*مانند آزمایش اول بخش اول کد صحت ورودی‌ها را چک می‌کند و در صورت وجود مشکل در وردی، پیغامی مناسب چاپ می‌کند.*

*در نهایت با استفاده از (در اینجا) عملیات ریاضی مد نظر را به انجام می‌رساند. با تمام شدن کیس‌های مختلف با آن را به اتمام می‌رساند.*

#!/bin/bash  
  
flag=0  
function **check**(){  
if [ -z $1 ]; then  
    echo "First number is empty"  
    flag=1  
elif ! [[ $1 =~ ^[+-]?[0-9]+$ ]]; then  
    echo "First number is not valid"  
    flag=1  
fi  
if [ -z $3 ]; then  
    echo "Second number is empty"  
    flag=1  
elif  ! [[ $3 =~ ^[+-]?[0-9]+$ ]]; then  
    echo "Second number is not valid"  
    flag=1  
fi  
}  
  
check $1 $2 $3  
if [ $flag -eq 0 ]; then  
    case $2 in  
    "+")  
        expr $1 + $3  
        ;;  
    "-")  
        expr $1 - $3  
        ;;  
    "x")  
        expr $1 \\* $3  
        ;;  
    "/")  
        expr $1 / $3  
        ;;  
    \*)  
        echo "invalid input"  
        ;;  
    esac  
fi

*کد 2-1*

**

*شکل 2-1*

*آزمایش سوم:*

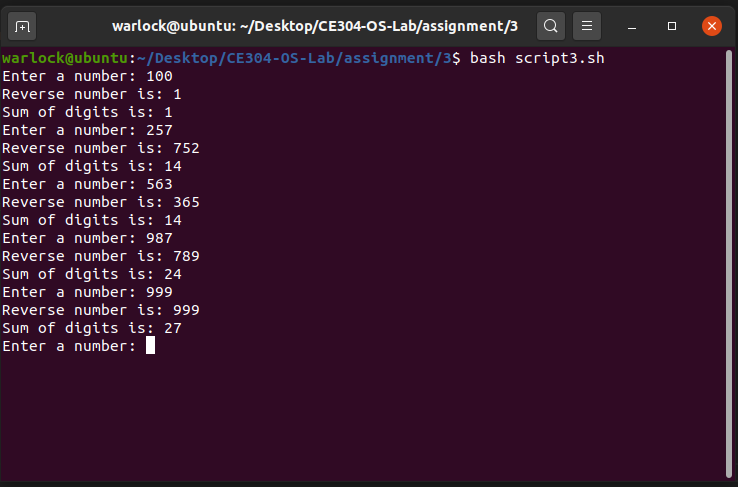
*در مرحله اول در یک که هیچگاه تمام نمی‌شود مگر اینکه را فشار دهیم بدنه برنامه را قرار می‌دهیم.*

*در مرحله دوم به کمک یک عدد را به عنوان ورودی دریافت می‌کنیم و با کمک الگوریتمی آن را به صورت برعکس شده چاپ می‌کنیم.*

*در این الگوریتم تا زمانی که عدد از 0 بزرگ تر باشد باقیمانده آن را بر 10 حساب می‌کنیم و به ابتدا عدد حدید برعکس شده اضافه می‌کنیم. با باقیمانده گیری بر 10 مدام در حال دریافتن یکان آن هستیم.*

#!/bin/bash  
  
while true; do  
    sd=0  
    rev=0  
    sum=0  
  
    read -p 'Enter a number: ' num  
    while [ $num -gt 0 ]  
    do  
        sd=$(( $num % 10 ))  
        rev=$(( $rev \* 10 + $sd ))  
        sum=$(( $sd + $sum ))  
        num=$(( $num / 10 ))  
    done  
  
    echo "Reverse number is: $rev"  
    echo "Sum of digits is: $sum"    
done

*کد 3-1*

**

*شکل 3-1*

آزمایش چهارم:

مانند آزمایش سوم با کمک دستور ورودی را گرفته و با کمک دستور خط مورد نظر را از فایل مورد نظر می‌خوانیم. حال برای خواندن تمامی خط‌های از استفاده می‌کنیم.

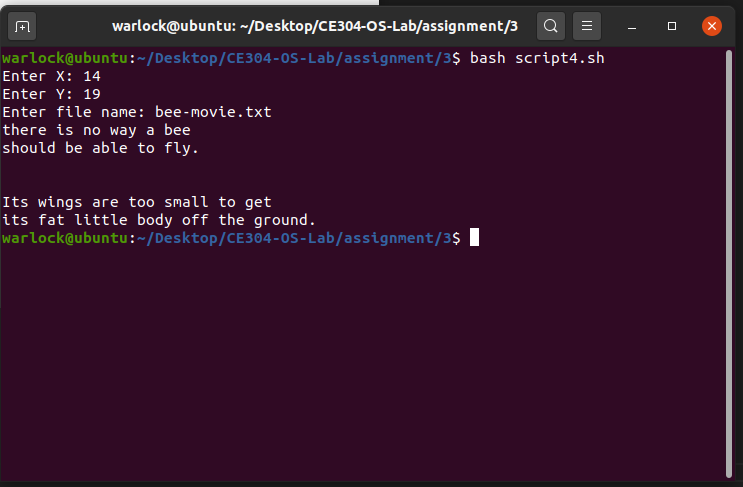
همچنین می‌شود با استفاده از (کد 4-2) نیز این کار را انجام داد و نیاز به پیمایش روی فایل ندارد.

#!/bin/bash  
  
read -p 'Enter X: ' x  
read -p 'Enter Y: ' y  
read -p 'Enter file name: ' fileName  
  
for (( i=x; i<=y; i++ ))  
do  
    sed -n $i'p' $filename  
done

کد 4-1

#!/bin/bash  
for arg in "$@"; do  
   grep -R "$arg" \*  
done

کد 4-2



شکل 4-1

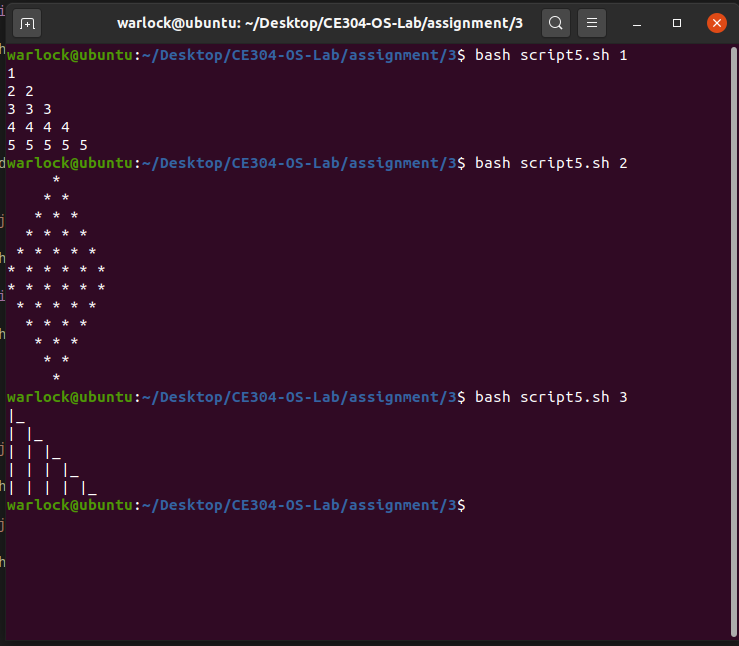
*آزمایش پنجم:*

*با گرفتن ورودی یکی از سه حالات مد نظر را در ورودی دریافت می‌کنیم. هر عدد یک شکل متمایز را به نمایش می‌گذارد.*

*الگوریتم استفاده شده نیز 3 تابع می‌باشند که با استفاده از شکل‌های مدنظر را به نمایش می‌گذارند.*

#!/bin/bash  
  
function **pattern1**(){  
    for ((i=1;i<=5;i++))  
    do  
        for j in $(seq 1 $i)  
        do  
            echo -n "$i "  
        done  
        echo  
    done  
}  
  
function **pattern2**(){  
    for ((i=1;i<=6;i++))  
    do  
        for ((j=5;j>=$i;j--))  
        do  
            echo -n " "  
        done  
        for j in $(seq 1 $i)  
        do  
            echo -n "\* "  
        done  
        echo  
    done  
    for ((i=1;i<=6;i++))  
    do  
        for ((j=1;j<$i;j++))  
        do  
            echo -n " "  
        done  
        for ((j=$i;j<=6;j++))  
        do  
            echo -n "\* "  
        done  
        echo  
    done  
     
}  
  
function **pattern3**(){  
    for ((i=1;i<=5;i++))  
    do  
         
        for ((j=1;j<$i;j++))  
        do  
            echo -n "| "  
        done  
        echo "|\_"  
    done  
}  
  
case $1 in  
    "1")  
        pattern1  
        ;;  
    "2")  
        pattern2  
        ;;  
    "3")  
        pattern3  
        ;;    
esac

*کد 5-1*

**

*شکل 5-1*

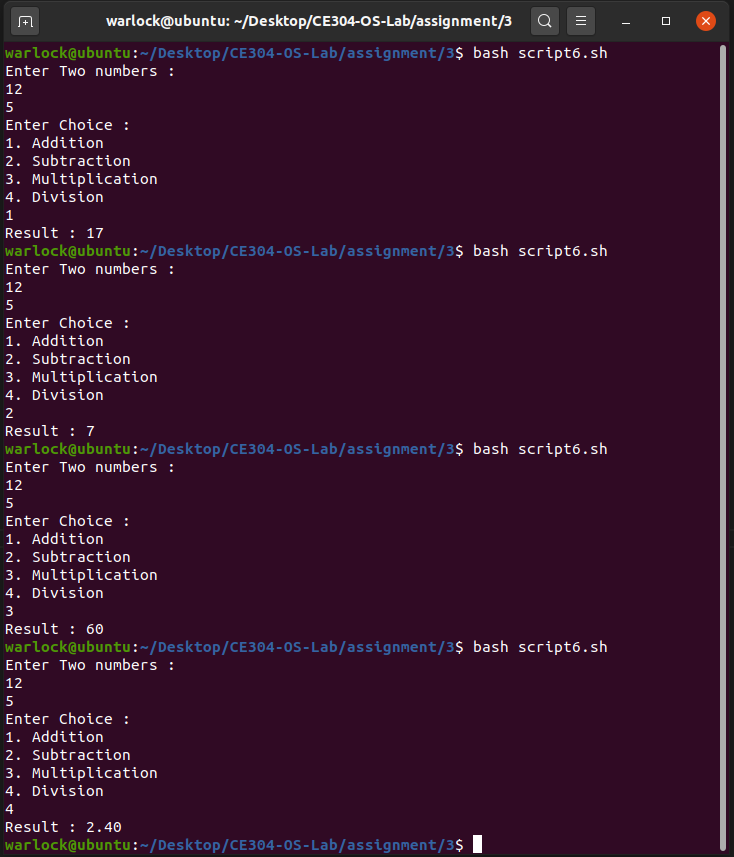
*بخش امتیازی:*

*برای این بخش به هدف نوشتن یک ماشین حسابی اعداد حقیقی ابتدا دو عدد ورودی گرفته، بعد یکی از 4 عملیات اصلی جمع، تفریق، ضرب و تقسیم را انتخاب می‌کنیم؛ و در نهایت نتیجه را چاپ می‌کنیم.*

*دقت کنید در تقسیم عددی اعشاری خروجی گرفته‌ایم که صحت کد را مشخص می‌کند.*

# !/bin/bash  
  
# Take user Input  
echo "Enter Two numbers : "  
read a  
read b  
  
# Input type of operation  
echo "Enter Choice :"  
echo "1. Addition"  
echo "2. Subtraction"  
echo "3. Multiplication"  
echo "4. Division"  
read ch  
  
# Switch Case to perform  
# calculator operations  
case $ch in  
1)res=`echo $a + $b | bc`  
;;  
2)res=`echo $a - $b | bc`  
;;  
3)res=`echo $a \\* $b | bc`  
;;  
4)res=`echo "scale=2; $a / $b" | bc`  
;;  
esac  
echo "Result : $res"

*کد 6-1*

**

*عکس 6-1*