Operating System

PORTFOLIO  
with

Lab Manual

NAME: Muhammad Iftikhar Uddin Khan Sami

ID: 022-16-113275

SECTION: IU- Gulshan Campus

DEGREE: BS - CS

**Operating System**

# List of Labs

***Lab #1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

* Introduction to OS and Fedora.
* Creation of Virtual Machine.

***Lab #2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Introduction to UNIX/LINUX Shell

***Lab # 3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

UNIX/LINUX Shell programming

* Variable
* Basic Variable

***Lab # 4\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

UNIX/LINUX Shell programming

* Using Array
* Basic Operator

***Lab # 5\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

UNIX/LINUX Shell programming

* Decision Making
* Shell Loop
* Loop Control
* Shell Function

***Lab # 6\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

UNIX/LINUX Shell programming

* Shell Substitution
* Quoting Mechanisms
* I/O Redirection
* Man\_page Help

***Lab # 7\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Linux File Management.

***Lab # 8\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Linux Directory Management

***Lab # 9\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Linux File Permission / Access Mode

***Lab # 10\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

CPU Scheduling Algorithms

***Lab # 11\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

File Allocation Strategies

***Lab # 12\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Memory Management Techniques

***Lab # 13\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Memory Management Techniques using Paging.

***Lab # 14\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Deadlock Management Techniques

***Lab # 15\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Page Replacement Algorithms

***Lab # 16\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Process Synchronization

# Lab Task:

## LAB # 01

### Task 01

Look for difference between VHD, VMDK and VDI

### Task 02

Create a Fedora Virtual machine also create a text file using fedora terminal

**Already completed**

## LAB # 02

Write Linux bash Shell Script, which will ask and display all information required to Student admission in Engineering and Sciences University

#!/bin/sh

echo "Enter your name"

read -r name

echo "Enter your Date of Birth"

read -r dob

echo "Enter your CNIC Number"

read -r cnic

echo "Enter your last qualification"

read -r qualification

echo "Enter your intermediate grade"

read -r grade

echo "Attached all documents mention their also 4 passport sizes photograph"

echo "YOUR DETAIL"

echo "Name $name"

echo "Date of Birth $dob"

echo "CNIC number $cnic"

echo "Qualification $qualification"

echo "Inter grade $grade"

## LAB # 03

Use following variables to write your CV in Linux bash script.

|  |  |  |
| --- | --- | --- |
| Variable | Script | Output |
| $0 |  |  |
| $n |  |  |
| $# |  |  |
| $\* |  |  |
| $@ |  |  |
| $? |  |  |
| $$ |  |  |
| $! |  |  |

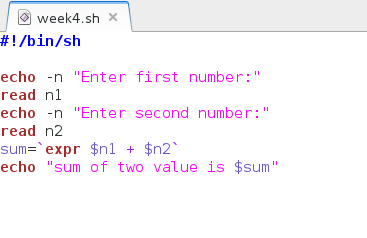
## LAB # 04

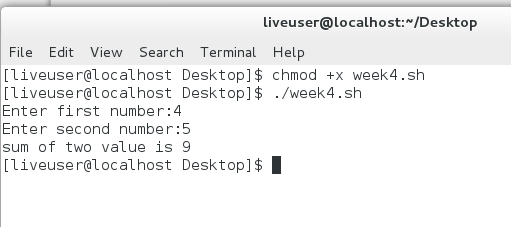
Write bash script to test all Athematic and logical operators in Linux

|  |  |  |
| --- | --- | --- |
| Operator: Task | Script | Output |
| + : add two number |  |  |
| - : make decrement operator |  |  |
| \* : Mutiply negative and positive number |  |  |
| /: divide any number by zero |  |  |
| Less than: Compare two value |  |  |
| Greater than: Compare two value |  |  |
| EquUniversityty: Compare two value |  |  |
| Not equal: Compare two value |  |  |
| And: make two simple and gate |  |  |
| Or: make two input or gate |  |  |
| Not: make 1 input not |  |  |

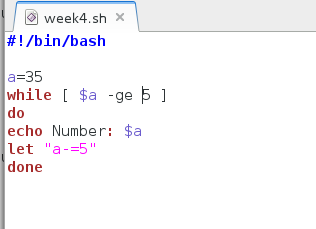
**Task: Write bash scrip to test all Athematic and logical operators in Linux**

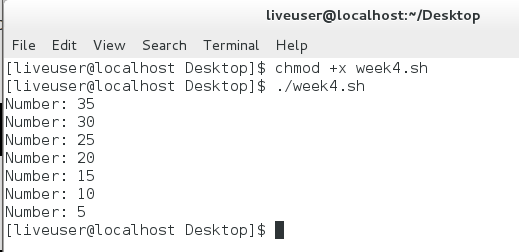
### + : add two number

****

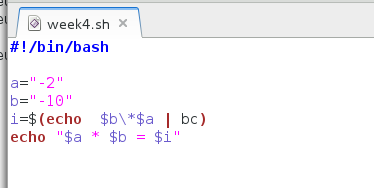
****

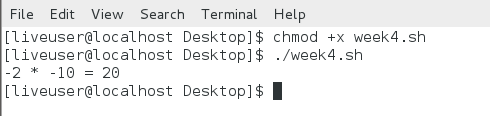
### - : make decrement operator

****

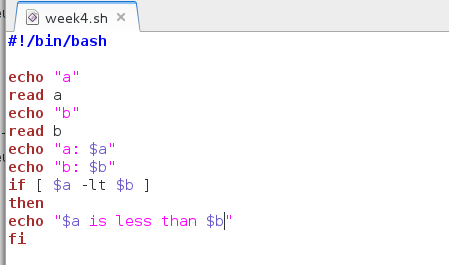
****

### \* : Mutiply negative and positive number

****

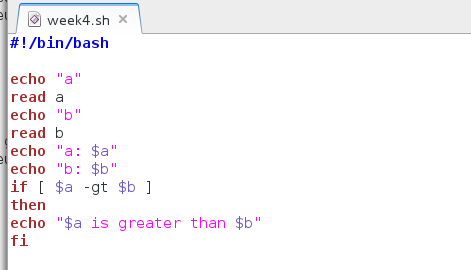
****

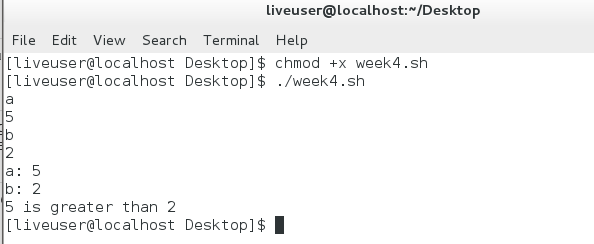
### Less than



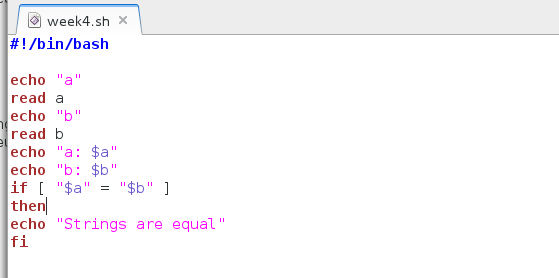


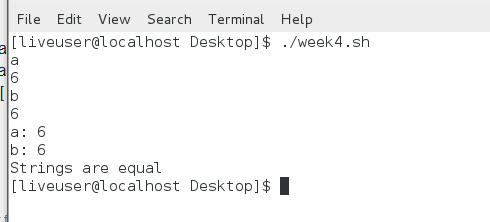
### Greater than



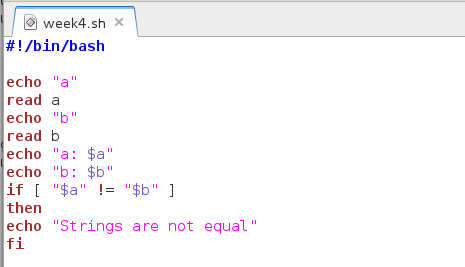


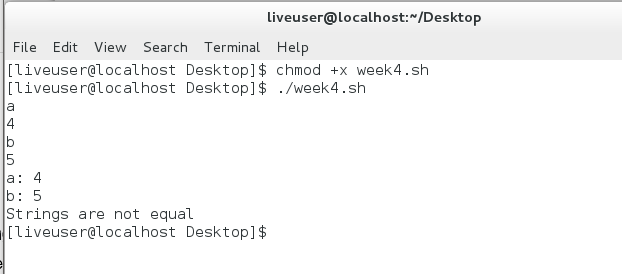
### Equals to:



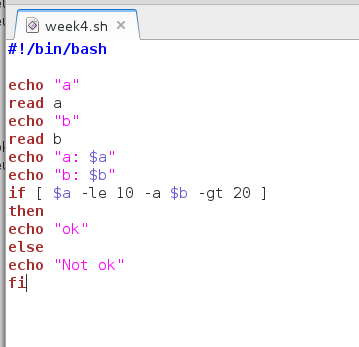


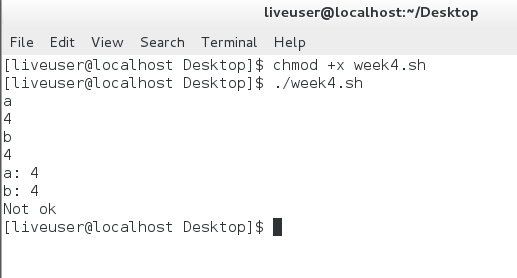
### Not equal



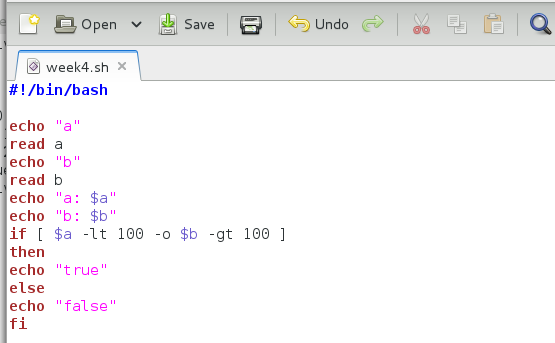


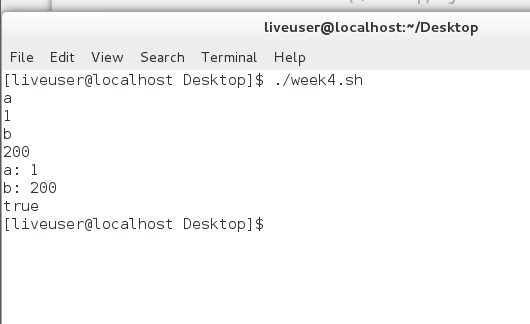
### AND operator:



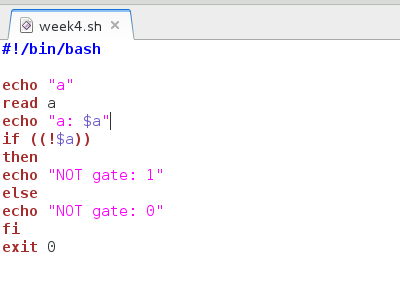


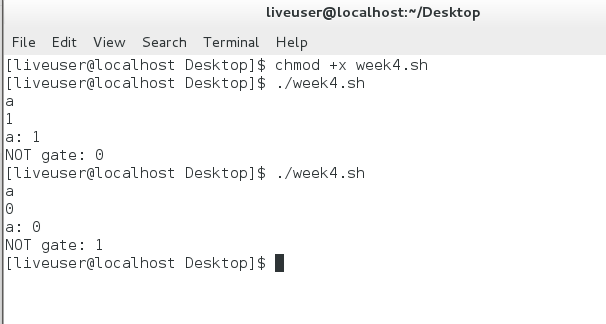
### **OR** operator**:**





### **NOT** operator**:**





## LAB # 05

Sort integer Array in ascending order.

### Code:

arr=(10 8 20 100 12)

echo "Array in original order"

echo "${arr[\*]}"

for ((i = 0; i<5; i++))

do

for((j = 0; j<5-i-1; j++))

do

if [ ${arr[j]} -gt ${arr[$((j+1))]} ]

then

# swap

temp=${arr[j]}

arr[$j]=${arr[$((j+1))]}

arr[$((j+1))]=$temp

fi

done

done

echo "Array in sorted order :"

echo "${arr[\*]}"

### Output:



Find shortest value in integer array.

### Code:

arrayName=(1 2 3 4 5 6 7)

min=${arrayName[0]}

# Loop through all elements in the array

for i in "${arrayName[@]}":

do

if [[ "$i" -lt "$min" ]]; then

min="$i"

fi

done

echo "Min is: $min"

### Output:

Make shell function which can find out that input number is prime or not

### Code:

#!/bin/sh

Check\_prime () {

num=29

for((i=2; i<=num/2; i++))

do

if [ $((num%i)) -eq 0 ]

then

echo "$num is not a prime number."

exit

fi

done

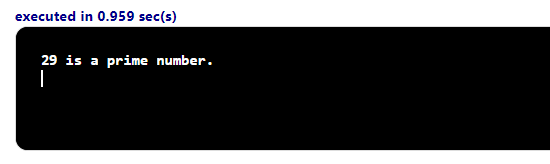
echo "$num is a prime number."

}

#function calling

Check\_prime

### Output:



Make shell function which can eliminate all odd numbers from integer array.

### Code:

### Output:

Make shell function which can find the last three prime number lesser than 100.

### Code:

### Output:

## LAB # 06 No Tasks

## LAB # 07 No Tasks

## LAB # 08 No Tasks

## LAB # 09 No Tasks

## LAB # 10 No Tasks

## LAB # 11 – Memory Management

* You are required to make changes in the above programs and introduce the use of compaction where required.
* Write code to simulate Worst-Fit Algorithm

### Code:

Also available on Github link

#include<stdio.h>

#include<conio.h>

#define max 25

void main()

{

int frag[max],b[max],f[max],i,j,nb,nf,temp;

static int bf[max],ff[max];

printf("\n\tMemory Management Scheme - First Fit");

printf("\nEnter the number of blocks:");

scanf("%d",&nb);

printf("Enter the number of files:");

scanf("%d",&nf);

printf("\nEnter the size of the blocks:-\n");

for(i=1;i<=nb;i++)

{

printf("Block %d:",i);

scanf("%d",&b[i]);

}

printf("Enter the size of the files :-\n");

for(i=1;i<=nf;i++)

{

printf("File %d:",i);

scanf("%d",&f[i]);

}

for(i=1;i<=nf;i++)

{

for(j=1;j<=nb;j++)

{

if(bf[j]!=1)

{

temp=b[j]-f[i];

if(temp>=0)

{

ff[i]=j;

break;

}

}

}

frag[i]=temp;

bf[ff[i]]=1;

}

printf("\nFile\_no:\tFile\_size :\tBlock\_no:\tBlock\_size:\tFragement");

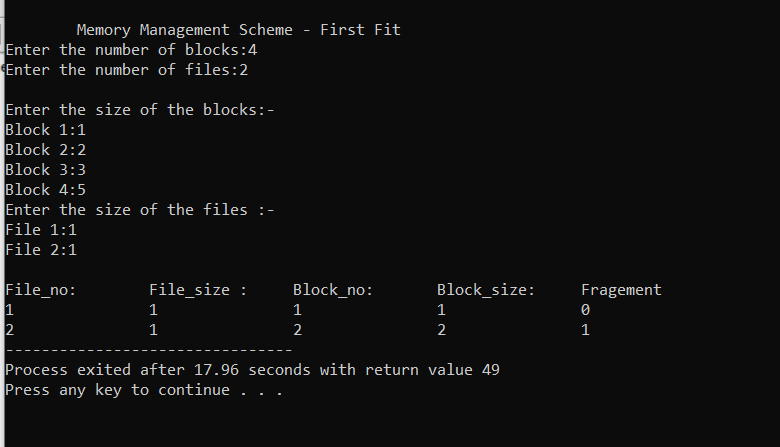
for(i=1;i<=nf;i++)

printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,f[i],ff[i],b[ff[i]],frag[i]);

getch();

}

### Output:



## LAB # 12 No Tasks

## LAB # 13

### Output 1:

