**JAVA PROGRAMMING**

**PROGRAM NO 1:**

**AIM:**

**Simple program to create class for adding two numbers.**

**PROGRAM**

import java.io.\*;

class sum

{

public static void main(String args[])throws IOException

{

int s,n,m;

float a,b,c;

InputStreamReader isr=new InputStreamReader(System.in);

BufferedReader br=new BufferedReader(isr);

System.out.println("Enter two numbers:");

n=Integer.parseInt(br.readLine());

m=Integer.parseInt(br.readLine());

s=n+m;

System.out.println("Sum of " +n+" and "+m+" is:"+s);

System.out.println("\nEnter two floating value is:");

a=Float.valueOf(br.readLine());

b=Float.valueOf(br.readLine());

c=a+b;

System.out.println("\nSum of two floating point value is:"+c);

}

}

**OUTPUT**

Enter two numbers:

45

60

Sum of 45 and 60 is:105

Enter two floating value is:

22.56

55.14

Sum of two floating point value is:77.7

**PROGRAM NO 2:**

**AIM:**

**Simple program to create class for finding largest of three numbers.**

**PROGRAM**

import java.io.\*;

import java.util.Scanner;

class sum

{

public static void main(String args[])

{

int a,b,c;

Scanner s=new Scanner(System.in);

System.out.println("Enter the 3 numbers:");

a=s.nextInt();

b=s.nextInt();

c=s.nextInt();

if(a>b)

{

if(a>c)

{

System.out.println("large="+a);

}

else

{

System.out.println("large="+c);

}

}

else

{

if(b>c)

{

System.out.println("large="+b);

}

else

{

System.out.println("large="+c);

}

}

}

}

**OUTPUT**

Enter the 3 numbers:

10 15 26

large=26

**PROGRAM NO 3:**

**AIM:**

**To implement nested and inner class.**

**PROGRAM**

import java.io.\*;

import java.util.Scanner;

class outer

{

int x,y;

outer(int a,int b)

{

x=a;

y=b;

}

void read()

{

inner inr=new inner();

inr.add();

}

class inner

{

int s;

void add()

{

s=x+y;

System.out.println("Sum of two number is:"+s);

}

}

}

class nestedinner

{

public static void main(String args[])throws IOException

{

int a,b;

Scanner sc=new Scanner(System.in);

System.out.println("Enter 2 numbers");

a=sc.nextInt();

b=sc.nextInt();

outer otr=new outer(a,b);

otr.read();

}

}

**OUTPUT**

Enter 2 numbers

10

25

Sum of two number is:35

**PROGRAM NO 4:**

**AIM:**

**To implement string manipulation.**

**PROGRAM**

import java.io.\*;

import java.lang.\*;

class StringManipulation

{

public static void main(String args[]) throws IOException

{

String s1,s2;

BufferedReader ob=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter first string:");

s1=ob.readLine();

System.out.println("Enter second string:");

s2=ob.readLine();

System.out.println("STRING MANIPULATION OPERATIONS");

System.out.println("---------------------------------------------------------");

System.out.println("First string is: "+s1);

System.out.println("Second string is: "+s2);

System.out.println("After converting first string to uppercase: "+s1.toUpperCase());

System.out.println("After converting second string to lowercase: "+s2.toLowerCase());

char ch=s1.charAt(0);

System.out.println("First string character at position is: "+ch);

if(s1.compareTo(s2)==0)

{

System.out.println(s1+"Equals"+s2+"-> "+s1.equals(s2));

}

else

{

System.out.println(s1+" Not Equals "+s2+"-> "+s1.equals(s2));

}

String s4=s1.concat(s2);

System.out.println("After concatenating s1 and s2: "+s4);

String s3=s1.replace('B','S');

System.out.println("After replacing character 'B' into 'S' in first string is: "+s3);

s1=s1.trim();

System.out.println("After Trim: "+s1);

System.out.println("String Buffer\n----------------");

StringBuffer sb=new StringBuffer(s1);

System.out.println("Buffer: "+sb);

System.out.println("The character at position 1 is: "+sb.charAt(1));

sb.append("best").append("!").toString();

System.out.println("After appending: "+sb);

sb.insert(5,"is");

System.out.println("After inserting: "+sb);

sb.reverse();

System.out.println("After reversing: "+sb);

sb.replace(5,7,"iii");

System.out.println("After Replacing: "+sb);

}

}

**OUTPUT**

Enter first string:

Barun

Enter second string:

Sopthi

STRING MANIPULATION OPERATIONS

--------------------------------------------------------

First string is: Barun

Second string is: Sopthi

After converting first string to uppercase: BARUN

After converting second string to lowercase: sopthi

First string character at position is: B

Barun Not Equals Sopthi-> false

After concatenating s1 and s2: BarunSopthi

After replacing character 'B' into 'S' in first string is: Sarun

After Trim: Barun

String Buffer

----------------

Buffer: Barun

The character at position 1 is: a

After appending: Barunbest!

After inserting: Barunisbest!

After reversing: !tsebsinuraB

After Replacing: !tsebiiinuraB

**PROGRAMNO5**

**AIM:**

**Program to implement commandline arguments.**

**PROGRAM**

//import java.util.Scanner;

import java.io.\*;

class commandline

{

public static void main(String args[])throws IOException

{

int temp,n,sum=0;

n=Integer.parseInt(args[0]);

BufferedReader br= new BufferedReader(new InputStreamReader(System.in));

int a[]=new int[20];

System.out.println("Command line arguments:");

System.out.println("Enter the elements");

for(int i=0;i<n;i++)

{

a[i]=Integer.parseInt(br.readLine());

}

for(int i=0;i<n;i++)

{

for(int j=i+1;j<n;j++)

{

if(a[i] > a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

System.out.println("Assending order");

for(int i=0;i<n-1;i++)

{

System.out.println(a[i]+"");

}

System.out.println(a[n-1]);

for(int i=0;i<n;i++)

{

sum=sum+a[i];

}

System.out.println("Sum of array is="+sum);

}

}

**OUTPUT**

[Agnes@DBCPC21 Desktop]$ java comndline 5

Enter all the elements

34

12

11

32

10

Assending order

10

11

12

32

34

Sum of array is=99

**PROGRAM NO 6:**

**AIM:**

**To implement function overloading.**

**PROGRAM**

import java.io.\*;

class sample

{

int l,b,h;

void area(int l,int b)

{

int area1=l\*b;

System.out.println("Area of triangle="+area1);

}

void area(int l,int b,int h)

{

int area2=l\*b\*h;

System.out.println("Area of rectangle="+area2);

}

}

class overloading

{

public static void main(String args[])throws IOException

{

int l,b,h;

System.out.println("Enter l,b,h");

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

l=Integer.parseInt(br.readLine());

b=Integer.parseInt(br.readLine());

h=Integer.parseInt(br.readLine());

sample ob=new sample();

ob.area(l,b);

ob.area(l,b,h);

}

}

**OUTPUT**

Enter l,b,h

12

5

2

Area of triangle=60

Area of rectangle=120

**PROGRAM NO 7:**

**AIM:**

**To implement abstractbase class.**

**PROGRAM**

import java.io.\*;

abstract class shape

{

float l,b;

shape()

{

l=10;

b=20;

}

shape(float x,float y)

{

l=x;

b=y;

}

abstract public float area();

}

class rectangle extends shape

{

rectangle()

{

super();

}

rectangle(float x,float y)

{

super(x,y);

}

public float area()

{

float a=l\*b;

return a;

}

}

class triangle extends shape

{

triangle()

{

super();

}

triangle(float x,float y)

{

super(x,y);

}

public float area()

{

float a=(l\*b)/2;

return a;

}

}

class abstractclass

{

public static void main(String args[])throws IOException

{

float x,y,a1,a2,area1,area2;

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

rectangle rect=new rectangle();

a1=rect.area();

System.out.println("Area of rectangle using default constructor: "+a1);

triangle tri=new triangle();

a2=tri.area();

System.out.println("Area of triangle using default constructor: "+a2);

System.out.println("Enter the length:");

x=Float.parseFloat(br.readLine());

System.out.println("Enter the breadth:");

y=Float.parseFloat(br.readLine());

rectangle r=new rectangle(x,y);

area1=r.area();

System.out.println("Area of rectangle: "+area1);

triangle t=new triangle(x,y);

area2=t.area();

System.out.println("Area of Triangle: "+area2);

}

}

**OUTPUT**

Area of rectangle using default constructor: 200.0

Area of triangle using default constructor: 100.0

Enter the length:

12

Enter the breadth:

20

Area of rectangle: 240.0

Area of Triangle: 120.0

**PROGRAM NO 8:**

**AIM:**

**To implement multiple inheritance.**

**PROGRAM**

import java.io.\*;

interface exam

{

public void percent\_call();

}

class student

{

String name;

int rollno;

public int mark1,mark2;

student(String n,int r,int m1,int m2)

{

name=n;

rollno=r;

mark1=m1;

mark2=m2;

}

void display()

{

System.out.println("Name:"+name);

System.out.println("roll no:"+rollno);

System.out.println("mark1:"+mark1);

System.out.println("mark2:"+mark2);

}

}

class result extends student implements exam

{

float percentage,total;

result(String n1,int r1,int mk1,int mk2,float t)

{

super(n1,r1,mk1,mk2);

total=t;

}

public void percent\_call()

{

percentage=((mark1+mark2)/total)\*100;

}

void display()

{

super.display();

System.out.println("Percentage:"+percentage);

}

}

class inheritance

{

public static void main(String args[])throws IOException

{

DataInputStream ob=new DataInputStream(System.in);

int roll,mkr1,mkr2,n;

float tot1,tot2,total;

String name1;

System.out.println("percentage of mark using multiple inheritance");

System.out.println(".................................................");

System.out.println("Enter the limit:");

n=Integer.parseInt(ob.readLine());

result r[]=new result[n];

for(int i=0;i<n;i++)

{

System.out.println("\nEnter the name:");

name1=ob.readLine();

System.out.println("Enter the roll number:");

roll=Integer.parseInt(ob.readLine());

System.out.println("Enter the total mark in subject1:");

tot1=Integer.parseInt(ob.readLine());

System.out.println("Enter the mark obtained in subject1:");

mkr1=Integer.parseInt(ob.readLine());

System.out.println("Enter the total mark in subject2:");

tot2=Integer.parseInt(ob.readLine());

System.out.println("Enter the mark obtained in subject2:");

mkr2=Integer.parseInt(ob.readLine());

total=tot1+tot2;

r[i]=new result(name1,roll,mkr1,mkr2,total);

r[i].percent\_call();

System.out.println("STUDENT DETAILS");

System.out.println("---------------");

r[i].display();

}

}

}

**OUTPUT**

Percentage of mark using multiple inheritance

..........................................................................

Enter the limit:

2

Enter the name:

Barun

Enter the roll number:

1104

Enter the total mark in subject1:

100

Enter the mark obtained in subject1:

79

Enter the total mark in subject2:

100

Enter the mark obtained in subject2:

89

STUDENT DETAILS

----------------------------

Name:Barun

roll no:1104

mark1:79

mark2:89

Percentage:84.0

Enter the name:

Sanaya

Enter the roll number:

1001

Enter the total mark in subject1:

100

Enter the mark obtained in subject1:

90

Enter the total mark in subject2:

100

Enter the mark obtained in subject2:

88

STUDENT DETAILS

----------------------------

Name:Sanaya

roll no:1001

mark1:90

mark2:88

Percentage:89.0

**PROGRAM NO 9:**

**AIM:**

**To implement package-complex.**

**PROGRAM**

package complexpack;

import java.io.\*;

public class complex

{

int real,imag;

public complex()

{

real=0;

imag=0;

}

public complex(int r,int i)

{

real=r;

imag=i;

}

public void addcomplex(complex ob1,complex ob2)

{

real=ob1.real+ob2.real;

imag=ob1.imag+ob2.imag;

}

public void display()

{

System.out.println(real+"+"+imag+"i");

}

}

Main Class

import java.io.\*;

import complexpack.\*;

public class complexmain

{

public static void main(String args[])throws IOException

{

int r,r1,i,i1;

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter first complex number");

r=Integer.parseInt(br.readLine());

i=Integer.parseInt(br.readLine());

complex ob1=new complex(r,i);

System.out.println("Enter second complex number");

r1=Integer.parseInt(br.readLine());

i1=Integer.parseInt(br.readLine());

complex ob2=new complex(r1,i1);

complex ob3=new complex();

ob3.addcomplex(ob1,ob2);

System.out.println("Addition of complex numbers:");

ob3.display();

}

}

**OUTPUT**

Enter first complex number

10

6

Enter second complex number

4

4

Addition of complex numbers:

14+10i

**PROGRAM NO 10:**

**AIM:**

**To implement package-palindrome.**

**PROGRAM**

import palpack.Palindrome;

import java.io.\*;

class PalTest

{

public static void main(String args[])throws IOException

{

StringBuffer sb=new StringBuffer(args[0]);

Palindrome p=new Palindrome();

p.pal(sb);

}

}

package palpack;

public class Palindrome

{

public void pal(StringBuffer s1)

{

StringBuffer s=new StringBuffer();

String s2=s1.toString();

s=s1.reverse();

String s3;

s3=s.toString();

if(s2.equals(s3))

System.out.println("It is Palindrome");

else

System.out.println("Not a Palindrome");

}

}

**OUTPUT**

[Agnes@DBCPC21 Desktop]$ java PalTest hannah

It is Palindrome

[Agnes@DBCPC21 Desktop]$ java PalTest cris

Not a Palindrome

**PROGRAM NO 11:**

**AIM:**

**Program to implement exception handling.**

**PROGRAM**

import java.io.\*;

import java.lang.\*;

class MyException extends Exception

{

private int detail;

MyException(int a)

{

detail=a;

}

public String toString()

{

return"MyException["+detail+"]";

}

}

class ExceptionMain

{

static void check(int a)throws MyException

{

System.out.println("Called check("+a+")");

if(a<10)

{

System.out.println("Number "+a+" is less than 10");

}

else

{

if(a>10)

{

System.out.println("Number "+a+" is Greater than 10");

}

else

throw new MyException(a);

}

}

public static void main(String args[])

{

int x;

x=Integer.parseInt(args[0]);

System.out.println("Raise Exception,If Given Number is Equal to 10");

System.out.println(" Number is "+x);

try

{

check(x);

}

catch(MyException e)

{

System.out.println("Equal to 10 "+e);

}

}

}

**OUTPUT**

Raise Exception,If Given Number is Equal to 10

Number is 8

Called check(8)

Number 8 is less than 10

[Agnes@DBCPC21 Desktop]$ java ExceptionMain 18

Raise Exception,If Given Number is Equal to 10

Number is 18

Called check(18)

Number 18 is Greater than 10

[Agnes@DBCPC21 Desktop]$ java ExceptionMain 10

Raise Exception,If Given Number is Equal to 10

Number is 10

Called check(10)

Equal to 10 MyException[10]

**PROGRAM NO 12 :**

**AIM:**

**Program to implement multithreading.**

**PROGRAM**

import java.io.\*;

class A extends Thread

{

public void run()

{

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

System.out.println("From Thread A:i="+i);

System.out.println("Exit from A");

}

}

class B extends Thread

{

public void run()

{

for(int j=0;j<5;j++)

System.out.println("From Thread B:j="+j);

System.out.println("Exit from B");

}

}

class Multithreadtest

{

public static void main(String args[])

{

A oba=new A();

B obb=new B();

oba.start();

obb.start();

}

}

**OUTPUT**

From Thread B:j=0

From Thread B:j=1

From Thread B:j=2

From Thread B:j=3

From Thread B:j=4

Exit from B

From Thread A:i=0

From Thread A:i=1

From Thread A:i=2

From Thread A:i=3

From Thread A:i=4

Exit from A

**PROGRAM NO 13:**

**AIM:**

**Program to implement Thread- Synchronized.**

**PROGRAM**

class ClassA

{

void printvalue()

{

try

{

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

{

System.out.println(i+" ");

Thread.sleep(1000);

}

}

catch(InterruptedException e)

{

}

}

}

class Threadsync implements Runnable

{

ClassA ob1;

Thread t;

Threadsync(ClassA c)

{

this.ob1=c;

t=new Thread(this);

}

public void run()

{

synchronized(ob1)

{

ob1.printvalue();

}

}

public static void main(String args[])

{

ClassA ca=new ClassA();

Threadsync one=new Threadsync(ca);

one.t.start();

Threadsync two=new Threadsync(ca);

two.t.start();

Threadsync three=new Threadsync(ca);

three.t.start();

}

}

**OUTPUT**

[Agnes@DBCPC21 Desktop]$ vi Threadsync.java

[Agnes@DBCPC21 Desktop]$ javac Threadsync.java

[Agnes@DBCPC21 Desktop]$ java Threadsync

0

1

2

3

4

5

0

1

2

3

4

5

0

1

2

3

4

5

**PROGRAM NO 14:**

**AIM:**

**Program to implement threading .**

**PROGRAM**

import java.io.\*;

import java.lang.\*;

class Example implements Runnable

{

Thread t;

public Example(String threadname)

{

t=new Thread(this,threadname);

}

public void run()

{

System.out.println(Thread.currentThread());

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

{

System.out.println(i);

}

}

}

public class Threadtest

{

public static void main(String args[])

{

Example ob=new Example("First");

ob.t.start();

System.out.println("this is:"+Thread.currentThread());

}

}

**OUTPUT**

this is:Thread[main,5,main]

Thread[First,5,main]

0

1

2

3

4

5

**PROGRAM NO 15:**

**AIM:**

**Program to implement applet programming.**

**PROGRAM**

import java.awt.\*;

import java.applet.\*;

/\* <applet code="scroll" width=400 height=300>

</applet>

\*/

public class scroll extends Applet implements Runnable

{

int x=0;

Thread t;

public void init()

{

t=new Thread(this);

t.start();

}

public void run()

{

for(int i=0;i<500;i++,x++)

{

repaint();

try

{

t.sleep(50);

}

catch(InterruptedException e)

{

}

}

}

public void paint(Graphics g)

{

Font f=new Font("Arial",Font.BOLD,20);

g.setFont(f);

g.drawString("Welcome to Applets",x,25);

}

}

**OUTPUT**



**PROGRAM NO 16:**

**AIM:**

**Program to implement applet programming .**

**PROGRAM**

import java.awt.\*;

import java.applet.\*;

/\* <applet code="smily" width=300 height=100>

\*

\* </applet>

\* \*/

public class smily extends Applet

{

public void paint(Graphics g)

{

Font f=new Font ("Helvetica",Font.BOLD,20);

g.setFont(f);

g.drawString("Keep smiling",50,30);

g.drawOval(60,60,200,200);

g.fillOval(90,120,50,20);

g.fillOval(190,120,50,20);

g.drawLine(165,135,165,175);

g.drawArc(115,130,95,95,0,-180);

}

}

**OUTPUT**



**PROGRAM NO 17:**

**AIM:**

**Program to implement AWT-form .**

**PROGRAM**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\*<applet code=Frm height=250 width=250>

</applet>\*/

public class Frm extends Applet implements ActionListener

{

Button b1;

String n,p;

Frwnd f1;

public void init()

{

b1=new Button("window");

add(b1);

b1.addActionListener(this);

}

class Frwnd extends Frame implements ActionListener

{

Label l1,l2;

TextField t1,t2;

Button chk;

public Frwnd(String ss)

{

super(ss);

setLayout(new FlowLayout());

l1=new Label("username");

l2=new Label("password");

t1=new TextField(20);

t2=new TextField(20);

t2.setEchoChar('\*');

chk=new Button("check");

add(l1);

add(t1);

add(l2);

add(t2);

chk.addActionListener(this);

add(chk);

}

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource()==chk)

{

n="Name:"+t1.getText();

p="Password:"+t2.getText();

repaint();

f1.setVisible(false);

}

}

}

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource()==b1)

{

f1=new Frwnd("Login");

f1.setSize(300,300);

f1.show();

}

}

public void paint(Graphics g)

{

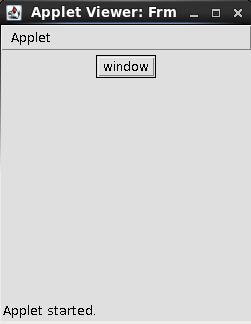
g.drawString(n,100,50);

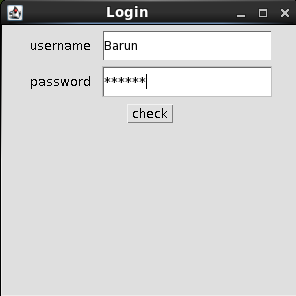
g.drawString(p,100,70);

}

}

**OUTPUT**





****

**PROGRAM NO 18:**

**AIM:**

**Program to implement AWT-list.**

**PROGRAM**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\*

<applet code="Listclr" width=300 height=180>

</applet>

\*/

public class Listclr extends Applet implements ActionListener

{

Dimension d;

List col;

String msg = "";

String s="";

int f;

public void init()

{

col = new List(4, false);

col.add("Green");

col.add("Yellow");

col.add("Blue");

col.select(1);

add(col);

col.addActionListener(this);

}

public void actionPerformed(ActionEvent ae)

{

String cmd=ae.getActionCommand();

if(cmd.equals("Yellow"))

{

f=0;

}

else if(cmd.equals("Blue"))

{

f=1;

}

else

{

f=2;

}

repaint();

}

public void paint(Graphics g)

{

int idx[];

d = getSize();

if(f==0)

setBackground(Color.yellow);

else if(f==1)

setBackground(Color.blue);

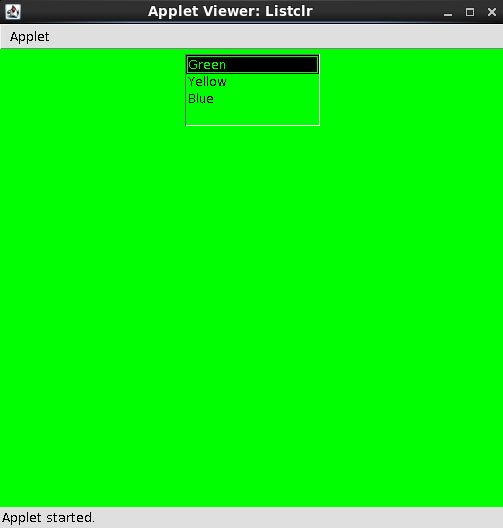
else

setBackground(Color.green);

}

}

**OUTPUT**



**PROGRAM NO 19:**

**AIM:**

**To implement event-driven program-keyevent.**

**PROGRAM**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\*

\*

\* <applet code="keyevent" width=300 height=100>

\*

\* </applet>

\*

\* \*/

public class keyevent extends Applet implements KeyListener

{

String msg = "";

int X = 10, Y = 20; // output coordinates

public void init()

{

addKeyListener(this);

}

public void keyPressed(KeyEvent ke)

{

showStatus("Key Down");

int key = ke.getKeyCode();

switch(key)

{

case KeyEvent.VK\_F1:

msg += "<F1>";

break;

case KeyEvent.VK\_F2:

msg += "<F2>";

break;

case KeyEvent.VK\_F3:

msg += "<F3>";

break;

case KeyEvent.VK\_PAGE\_DOWN:

msg += "<PgDn>";

break;

case KeyEvent.VK\_PAGE\_UP:

msg += "<PgUp>";

break;

case KeyEvent.VK\_LEFT:

msg += "<Left Arrow>";

break;

case KeyEvent.VK\_RIGHT:

msg += "<Right Arrow>";

break;

}

repaint();

}

public void keyReleased(KeyEvent ke)

{

showStatus("Key Up");

}

public void keyTyped(KeyEvent ke)

{

msg += ke.getKeyChar();

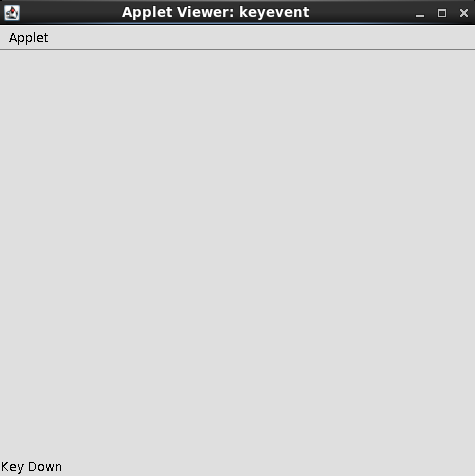
repaint();

}

}

**OUTPUT**





**PROGRAM NO 20:**

**AIM:**

**To implement event-driven program-mouse event.**

**PROGRAM**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\* <applet code="MouseEvents" width=500 height=300>

</applet>

\*/

public class MouseEvents extends Applet implements MouseListener, MouseMotionListener

{

String msg = "";

int mouseX = 0, mouseY = 0;

public void init()

{

addMouseListener(this);

addMouseMotionListener(this);

}

public void mouseClicked(MouseEvent me)

{

mouseX = 0;

mouseY = 10;

msg = "Mouse clicked.";

repaint();

}

public void mouseEntered(MouseEvent me)

{

mouseX = 0;

mouseY = 10;

msg = "Mouse entered.";

repaint();

}

public void mouseExited(MouseEvent me)

{

mouseX = 0;

mouseY = 10;

msg = "Mouse exited.";

repaint();

}

public void mousePressed(MouseEvent me)

{

mouseX = me.getX();

mouseY = me.getY();

msg = "Down";

repaint();

}

public void mouseReleased(MouseEvent me)

{

mouseX = me.getX();

mouseY = me.getY();

msg = "Up";

repaint();

}

public void mouseDragged(MouseEvent me)

{

mouseX = me.getX();

mouseY = me.getY();

msg = "\*";

showStatus("Dragging mouse at " + mouseX + ", " + mouseY);

repaint();

}

public void mouseMoved(MouseEvent me)

{

showStatus("Moving mouse at " + me.getX() + ", " + me.getY());

}

public void paint(Graphics g)

{

g.drawString(msg, mouseX, mouseY);

}

}

**OUTPUT**





**PROGRAM NO 21:**

**AIM:**

**Program for implementing JDBC .**

**PROGRAM**

import java.sql.\*;

class jdbc

{

public static void main (String[] args)

{

try

{

Class.forName("org.postgresql.Driver");

String url = "jdbc2:postgresql:jdbc2";

Connection con = DriverManager.getConnection(url,"postgres","");

Statement stm = con.createStatement();

ResultSet rs = stm.executeQuery("select name,place from RecordDetails");

System.out.println("");

System.out.println("");

System.out.println(" Name Place");

System.out.println("------- -------------");

while( rs.next())

{

System.out.print(rs.getString("name")+"\t");

System.out.println(""+rs.getString("place"));

}

}

catch(java.lang.ClassNotFoundException e)

{

System.err.print("ClassNotFoundException: ");

System.err.println(e.getMessage());

}

catch (SQLException e)

{

System.out.println("Exception!");

System.out.println(e.toString());

}

}

}

**OUTPUT**

[root@DBCPC21 Desktop]# vi jdbc.java

[root@DBCPC21 Desktop]# javac jdbc.java

[root@DBCPC21 Desktop]# java -classpath .:postgresql-9.0-801.jdbc4.jar jdbc

Name Place

------- -------------

Agnes Karuvanchal

[root@DBCPC21 Desktop]# service postgresql start

Starting postgresql service: [ OK ]

[root@DBCPC21 Desktop]# su - postgres

-bash-4.1$ createdb jdbc2

-bash-4.1$ psql jdbc2

psql (8.4.13)

Type "help" for help.

jdbc2=# create table RecordDetails(name varchar(20),place varchar(20));

CREATE TABLE

jdbc2=# insert into RecordDetails values('Agnes','Karuvanchal');

INSERT 0 1

jdbc2=# select \* from RecordDetails;

name | place

-------+-------------

Agnes | Karuvanchal

(1 row)

[root@DBCPC21 Desktop]# vi jdbc.java

[root@DBCPC21 Desktop]# javac jdbc.java

[root@DBCPC21 Desktop]# java -classpath .:postgresql-9.0-801.jdbc4.jar jdbc

Name Place

------- -------------

Agnes Karuvanchal

**PROGRAM NO 22:**

**AIM:**

**Program to demonstrate jdbc .**

**PROGRAM**

import java.sql.\*;

class PostgreSQLTest1

{

public static void main (String[]args)

{

try

{

Class.forName ("org.postgresql.Driver");

String url = "testdb:postgresql:testdb";

Connection con = DriverManager.getConnection (url, "postgres", "");

Statement stm = con.createStatement ();

ResultSet rs = stm.executeQuery ("select name,place,age from RDetails");

System.out.println ("");

System.out.println ("");

System.out.println (" Name Place Age");

System.out.println ("------ ------- ------ ");

while (rs.next ())

{

System.out.print (rs.getString ("name")+"\t");

System.out.print ("" + rs.getString ("place")+"\t");

System.out.println ("" + rs.getString ("age")+"\t");

}

}

catch (java.lang.ClassNotFoundException e)

{

System.err.print ("ClassNotFoundException: ");

System.err.println (e.getMessage ());

}

catch (SQLException e)

{

System.out.println ("Exception!");

System.out.println (e.toString ());

}

}

}

**OUTPUT**

[root@DBCPC25 Desktop]# vi PostgreSQLTest1.java

[root@DBCPC25 Desktop]# javac PostgreSQLTest1.java

[root@DBCPC25 Desktop]# java -classpath .:postgresql-9.0-801.jdbc4.jar PostgreSQLTest1

Name Place Age

------ ------- ------

Aleesha Vallithode 22

Subina Vilamana 23

import java.sql.\*;

class PostgreSQLTest1

{

public static void main (String[]args)

{

try

{

Class.forName ("org.postgresql.Driver");

String url = "testdb:postgresql:testdb";

Connection con = DriverManager.getConnection (url, "postgres", "");

Statement stm = con.createStatement ();

ResultSet rs = stm.executeQuery ("select name,place,age from RDetails");

System.out.println ("");

System.out.println ("");

System.out.println (" Name Place Age");

System.out.println ("------ ------- ------ ");

while (rs.next ())

{

System.out.print (rs.getString ("name")+"\t");

System.out.print ("" + rs.getString ("place")+"\t");

System.out.println ("" + rs.getString ("age")+"\t");

}

}

catch (java.lang.ClassNotFoundException e)

{

System.err.print ("ClassNotFoundException: ");

System.err.println (e.getMessage ());

}

catch (SQLException e)

{

System.out.println ("Exception!");

System.out.println (e.toString ());

}

}

}

[root@DBCPC25 Desktop]# service postgresql start

Starting postgresql service: [ OK ]

[root@DBCPC25 Desktop]# su - postgres

-bash-4.1$ createdb testdb

-bash-4.1$ psql testdb

psql (8.4.13)

Type "help" for help

testdb=# create table RDetails(name varchar(20),place varchar(20),age int);CREATE TABLE

testdb=# insert into RDetails values('Aleesha','Vallithode',22);

INSERT 0 1

testdb=# insert into RDetails values('Subina','Vilamana',23);

INSERT 0 1

testdb=# select \* from RDetails;

name | place | age

---------+------------+-----

Aleesha | Vallithode | 22

Subina | Vilamana | 23

(2 rows)

**SYSTEM AND NETWORK ADMINISTRATION**

# 1.LILO CONFIGURATION

Using LILO to boot Linux

LILO is a general- purpose boot manager that can boot whatever os installed on the system including Linux. There are many ways to configure LILO. The two common methods are:

1. Install LILO on the MBR of the hard disk.
2. Install LILO as the secondary boot loader for Linux only.

LILO is the general purpose boot manager that can boot whatever OS installed on the system including LINUX. There are many ways to configure LILO on the MPR; it is the first code to run after the stat up. LILO then allows you to choose between os.

/sbin/lilo

Using –v option with the lilo prints more information should something gets wrong.

Using –c option allows you to specify configuration file other than /etc/lilo.conf.

LILO has two aspects

1.The boot loader

2.The lilo command

Boot loader is the code that execute the boot time and boots either linux or oter os.

LILO is the command that cofigure and executes the boot loader and updates if it nessary.

Configuring LILO

The first step in configuring lilo is setting up the LILO configuring file. This file is often stored in:

1./etc/lilo.conf

2./boot/lilo.conf

3./etc/lilo/config

The first section of this file is called the global section. It setup some basic parameters:

boot=/dev/hda1 compact install=/boot/boot.b

map=/boot/map

boot-Name of the device where LILO should install itself. /dev/hda1 is MBR.

Compact – tells lilo to perform some optimization

Install – file containing the boot sector to use on MBR Map – map file that lilo creates when lilo installed.

After the globel section, there are seprate sections for each OS. Section for LINUX that is installed on /dev/hda2.

Image= /boot/vmlinuz

Label=linux

Root= /dev/hda2

Vga=ask

Section for MS-DOS /win 95/win98

Other= /dev/hda1

Table= /dev/had

Label=MS-DOS

After editing the lilo.conf file run the /sbin/lilo command. If the configuration file is other than lilo.conf file –c option should be used. After running the lilo command reboots the system. The first line of lilo.conf will be booted by default. To choose one of rhe other os listed in /etc/lilo.conf hold down shit ctrl key or press the scroll lock key while the system boots. Then boot prompt will appear. Press tab key to get the list of available option.

Removing LILO

If lilo is istalled on your MBR the easiest way is to remove it is to use windows fdisk. Runs fdisk and overwrites MBR with a valid windows boot record. Lilo saves back up copies of original boot record in the file /boot/boot.0300 and /boot/boot0800

We can use the dd command to replace the boot record on the drive with this back up copy.

dd if=/boot/boot.0300 of =/dev/had bs=446 count=1

Even though the files 512 bytes in size, only the first 446 bytes should be copied back to the MBR.

Boot parameters 1.root=partition the root parameter is used to specify the root partition..

2.ro=

It is the default mode. If ro is specified the root file system is mounted as read only, when specified as ro, an automatic file system checks up is done for integrity..

3.rw=

This parameter is used to mount the root file system in the read/write mode. The disadvantage of rw is that the file system check up is not done. This is dangerous as the file system can crash.

4.mem=size

Some BOIS is won’t recognize above 64 MB of RAM. Since needs the BIOS to tell it how much memory is installed systems with more than 64 MB of RAM will be incorrectly reported. We can use this option to tell how much memory is there. We can use either a hex address or a number followed by K or M.

5.debug=

Kernal messages are sending to the screen instead of being sent to the disk or to the syslog facility. When the system is booted, the kernel messages are sending to the log file in the disk. This parameter can be used to print the message on to the console as the system runs.

6. init=

It is possible to specify the program to be stored as the first program that is run after bootup. Usually the process specified is /sbin/init.

7. panic=”integer value”

panic=” “ is a future error signaled by the kernel itself. It occurs when the kernel is terminally confused and can’t continue with execution. When panic occurs the system waits for the administrator to restart. If panic=”integer” specified the kernal automatically restarts after a specified number of seconds.

8. vga=mode

This parameter will set the vga display mode. This setting overrides the entry /etc/lilo.conf. Valid modes are normal, extended and ask. Lilo has the ability to change the vga video modes from the standard 80\*24 to other modes, such as 80\*50 or 132\*42. for setting of vga=ask will provide a list of video modes and allow you to choose when boots.

9. single

System will be boot in the single user mode. Under single user mode, the system is nearly useless, very little configuration is done; file systems are unmounted and so on.

# 2.RUN-LEVEL CONFIGURATION

Linux can run in various modes based on the capabilities provided. A certain mode or state at which all operations are activated by running the system. These states are known as run levels. A run level is a particular set of processes ranging from a bare minimum for system administration only to a full operation supporting all configured devices. A number from zero to six defines run levels.

For most versions of Linux, the run levels have the following meanings.

* 0 – halt system.
* 1 - single run level mode.
* 2 – multi user without NFS.
* 3 – multi user with networking.
* 4 - not used.
* 5 – multi user with X-window system support.

.

The /etc/inittab file

The /etc/inittab file is tied closed to the init daemon. The lines in the /etc/inittab file follow a specific format. The format follows this pattern:

ID : run level : action : process.

id : 3 : intit default : si : sysinit : /etc/rc.d/rc.sysinit. 10 : 0 : wait : /etc/rc.d/rc 0

* 1. : 1 : wait : /etc/rc.d/rc 1
  2. : 2 : wait : /etc/rc.d/rc 2
  3. : 3 : wait : /etc/rc.d/rc 3
  4. : 4 : wait : /etc/rc.d/rc 4
  5. : 5 : wait : /etc/rc.d/rc 5 16 : 6 : wait : /etc/rc.d/rc 6 ca :: ctrlaltdel : /sbin/shutdown –t3 –r now

pf :: power fail : /sbin/ shutdown –f –h +2 “power failure system shutting down”

pr :: 12345 : power wait : /sbin/shutdown –c “power restored shut down cancelled”

1. : 2345 : respawn : /sbin/mingetty tty 1
2. : 2345 : respawn : /sbin/mingetty tty 2
3. : 2345 : respawn : /sbin/mingetty tty 3
4. : 2345 : respawn : /sbin/mingetty tty 4
5. : 2345 : respawn : /sbin/mingetty tty 5 6 : 2345 : respawn : /sbin/mingetty tty 6

x : 5 : respawn : /etc/x11/prefdm –nodaemon

Starting run level scripts

The /etc/ rc.d/rc script is a script that is integral to the concept of run levels. Any change of run level causes the script to be executed, with the new run level as an argument. The /etc/rc.d/rc script does

* Checks that run level script are correct.
* Determine current and previous run levels.
* Decides whether to enter interactive start up.
* Kill and start run level script.

A software package that has a service to start at boot time can add a script to the /etc/init.d directory. That script can then be linked to an appropriate run level directory and either be started or stopped. All the programs in the /etc/rcX.d directories where X is replaced by a run level number are symbolic links, usually to a file in /etc/ init.d. The /etc/rcX.d directories include:

* /etc/rc0.d : run level 0 directory.
* /etc/rc1.d : run level 1 directory. • /etc/rc2.d : run level 2 directory. • /etc/rc3.d : run level 3 directory. • /etc/rc4.d : run level 4 directory. • /etc/rc5.d : run level 5 directory.
* /etc/rc6.d : run level 6 directory.

Recognizing or removing run level scripts

There are several ways to deal with removing programs from the system start-up directories, adding them to particular run level or changing when they are executed. From a terminal window use the chk config command. From a GUI use the service configuration window.

Adding run level scripts

We can add the script to the /etc/init.d directory, then use the chk config command to configure it.

Manipulating run levels

Aside from the run level chosen at boot time and shut down or at reboot levels. We can change the run level at any time while we are logged in (as root user).

The telinit command enables us to specify a desired run level, causing the termination of all system process that shouldn’t exist in that run level and starting all processes that should be running.

# telinit 5

Determining current run level

We can determine the machines current run level with the aptly named run level command

# runlevel s 5

This means that the previous run level was ‘s’ and the current run level is 5. If the machine had booted properly, the previous run level would be listed as N to indicate that there really wasn’t a previous run level.

Changing to shut down run level

Shutting down the machine is simply a change in run level. Other ways to change the run level includes the reboot, halt, power off and shut down commands. The reboot command, which is a symbolic link to the halt command, executes a shutdown –r now, terminating all processes and rebooting the machine. The halt command executes shut down –h now, terminating all processes and leaving the machine in an idle state. Similarly the power off command, which is also a link to the halt command, executes a change to run level 0.

# 3.MOUNTING AND UNMOUNTING USING

# FSTAB

Linux uses a single directory structure regardless of how many disks and disk partitions are involved. Each partition file system must be part of layer directory structure. The entire tree has only one root directory and other file system are attached at lower layers.

The linux file system starts with the root partitions at the top: the root partition is on a partition of the first hard disk. Usually the disk also has other directories on it , such as /dev,/lib, /etc and so on.

Mounting

To mount a file system use the mount command:

mount device\_name mount\_point

Where device\_name is the name of the device such as hard disk, CD-ROM and so on and mount point is the name of the directory to which to want to mount the device.

Eg: To mount the cdrom file system in the directory in the directory /cdrom uses the following command:

mount /dev/cdrom /cdrom

To mount a file system read only use –r options as shown below

mount –r /dev/cdrom /cdrom

Unmounting

Where one of the mounted file system is disconnected the file system has been unmounted. Any mounted file system has been unmounted except for the root file system. To unmount the file system use the unmount command. To unmount cd-rom the command given below

umount /dev/cdrom umount /cdrom

Linux can also handle the unmounting as part of the shut down process.

Mounting the file system automatically

Any mounted file systems are not necessary mounted automatically when the system restarts. When the Linux boots, it must know where to find the file system to be mounted. Linux uses the /etc/rc initialization file to execute the command:

mount –av

When the linux execute this command, it knows to record the file /etc/fstab to find out which file system have to be mounted and where they should be mounted. Each line in the /etc/fstab file follows the format:

Device mount-location filesystem\_type options dump\_frequency pass\_number

Sample /etc/fstab is:

/dev/cdrom /cdrom iso9660 ro 1 1

# 4.USER AND GROUP MANAGEMENT

An operating system allows many users to use the resources of the system. For this each user should have an account, which is the entire file, resources and information belongs to each other.

Creating a user

Each user is identified by a unique integer. The userid or uid. Normally to create a user you need to add information about the user database and create home directory for him. Two command line alternatives: adduser and useradd for creating users. Various options for adduser command.

adduser [options] login-name

-r - Specifies the login name

-P - specifies the passwd for the user

Login

-d - allows specifying home directory

-g - specifies initial

group name or id

-s - specifies the default shell for the

user

-c - allows us to specify

comment

-e - expiring data for user account

The basic user interface in a Unix system is the /etc/passwd file, lets all the valid user names and associated information. The file has one line for user name and is divided into seven colondelimited fields.

User name

Passwd (in an encrypted form)

Numeric userid

Numeric group-id

Comments

*Home directory*

Default login name

Many Unix systems have shadow passwords. This is an alternative way of storing the password. The encrypted password is stored in a separate file: /etc/shadow, which only the root user can read. By command when the home directory for a new user is created, it is initialized with the files from the /etc/skel directory. File in this directory will provide a nice environment for these users.

Creating user manually

Edit /etc/passwd in any text editor and add a new line for the new account. You should make the passwd fields be (\*) so

that it is impossible to login. Similarly edit

/etc/ group in a text editor if you need to create a new group as well.

Create the home directory for the new user with mkdir. Copy the files from /etc/skel to the new home directory. Fix ownership and permissions with chown and chmod. The -R option is most useful. Set the passwd with the password command. After you set the password, the account will work.

Removing a user manually

To remove a user, you first removes all his files, mailboxes, mail aliases, print jobs, cron and at jobs and all other references to the user. Then you remove the relevant lines from /etc/passwd and /etc/group. Some Linux distribution comes with special commands to do this look for deluser or userdel. It is some times necessary to temporarily disable an account. For this we have to change his shell into a special program that just print a message userdel ren. su command enables us to login as different users.

Eg: su user name

Creating a group

Groups may be different in two ways. Implicitly, by GID, whenever a new GID appears in the fourth field of the password file, a new group is defined. Explicitly, by name and GID, via an entry in the /etc/group Each entry in the /etc/group consist of single line with the following form.

Name: \*: GID: additional-users

The meanings of these fields are as follows:

Name - A name identifying the groups usually restricted to 8 characters

\* Or! - The traditional group password field

GID - Groups identification number, usually start from 100

Additional users - this fields holds a list of users who are member of the group. In addition to those users belonging to the group by virtue of /etc/passwd. Commas must separate names.

Eg: chem.!! : 200: root, Williams. John. Jack

Command for managing groups:

The groupadd and groupmode commands can be used to set up and modify new groups.

groupadd group-name [options]

-g - Specify group id

-o - Override the existing GID if specified one exists

-f - Force flag exist with error message, if

Eg: groupadd conquers

The following command renames the group conquers to winners

groupmod-n conquers winners

Linux system provides grouppasswd commands for adding and removing members of group and for specifying group administrators the following command adds user rent to winner groups.

gppasswd-a ren winners

# 5. JOB SCHEDULING PROCESS

Job scheduling using cron and at utilities

Automating task is one of the best ways to keep a system running smoothly. the two utilities cron and at helps us to do our jobs easier. Both will execute commands at specified time

• **cron utility**

cron is a command that is designed to execute a particular job at specified time without anyone directly initiating them.Process to be executed by cron can be send through one of the two different files:

/usr/lib/cron/cron.allow and /usr/lib/cron/cron.deny. Many system uses /etc/cron.d/cron.allow and /etc/cron.d/cron.deny.

/etc/lib/cron/cron.allow contains the list of all users that are allowed to access cron and /usr/lib/cron/cron.deny contains the list of all users that are not allowed to access cron. If there two files does not exist, only root can submit process to cron

This file consists of one complete line for each to be submitted that specifies when to run the command and which command is to be executed. Each statement takes the following form

minute hour day\_of\_month month\_of\_year day\_of\_week command

Each line in crontab file has 6 columns separated with white space

* the minute of the hour (0-59)
* the hour of the day (0-23)
* the day of month(1-23)
* the days of week(sun=0,mon=1….sat=6)

Absolute pathname for the command to be executed are given. You must have executed permission for the utility or script. If you are submitting crontab file as a user, you must have file permissions or ownership set to allow your normal access.

Eg: 20 1 \* \* \* /usr/bin/calendar will execute 20 minutes per 1 in the morning everyday of the week; everyday of month, every month. The asterisk (\*) means all values.

if you have any errors in crontab file, cron mails you a no5tice of the problem when it processes your file

* **at utility**

The at command executes the command only once, at specified time the format is as follows:

at time date<file

We can specify most of the at command parameters in several different ways. It can also specify absolute time or as two digits that are taken as hours. We can add am or pm to time.

The file to be read as input to at command can be any file with commands in it. Suppose we have files called reorg.data with the following command on it. /usr/tparker/set\_perms

/usr/tparker/set\_database

/usr/tparker/index\_database

/usr/tparker/cleanup

if you want to execute this file at 8.30 pm, we use anyone of the following commands. at 20.30>reorg.data at 8.30pm<reorg.data at 20.30today,reorg.data

To remove a job you need the job id and the command, at-l.

All jobs are queued to at that kept in the directory

/usr/spool/cron/at jobs with the job id number as the filename

• **batch utility**

Here we are specifying that our commands be executed at the precise moment. The way to achieve this is through a command called batch. When we submit our jobs using these commands, Linux executes our job when it is relatively free and system load is light. Since the time of execution of our commands is left for the system to decide we don’t specify the time while executing the batch command.

$batch

sort employ.dat | grep palai>address.out

ctrl d

job 692322435.b at fri jun 14 17:00:00 IST 2006

‘b’ extension given to our job id signifies that it has been submitted using batch command

• **Logrotate**

NAME

logrotate - rotates, compresses, and mails system logs

SYNOPSIS

logrotate [-dv] [-f|--force] [-s|--state file] config\_file ..

DESCRIPTION

logrotate is designed to ease administration of systems that generate large numbers of log files. It allows automatic rotation, compression, removal, and mailing of log files. Each log file may be handled daily, weekly, monthly, or when it grows too large.

OPTIONS

-d, --debug

Turns on debug mode and implies -v. In debug mode, no changes will be made to the logs or to the logrotate state file.

-f, --force

Tells logrotate to force the rotation, even if it doesn’t think this is necessary. Sometimes this is useful after adding new entries to a logrotate.

-m, --mail <command>

Tells logrotate which command to use when mailing logs. This command should accept two arguments: 1) the subject of the message, and 2) the recipient. The command must then read a message on standard input and mail it to the recipient. The default mail command is /bin/mail -s.

-s, --state <statefile>

Tells logrotate to use an alternate state file. This is useful if logrotate is being run as a different user for various sets of log files. The

default state file is /var/lib/logrotate.status.

--usage

Prints a short usage message.

--?, --help

-Prints help message.

-v, --verbose

Turns on verbose mode.

CONFIGURATION FILE

logrotate reads everything about the log files it should be handling from the series of configuration files specified on the command line. Each config-

uration file can set global options (local definitions override global ones, and later definitions override earlier ones) and specify logfiles to rotate. A simple configuration file looks like this:

# sample logrotate configuration file compress

/var/log/messages { rotate 5 weekly postrotate

/usr/bin/killall -HUP syslogd

endscript

}

"/var/log/httpd/access.log" /var/log/httpd/error.log { rotate 5

mail www@my.org size 100k sharedscripts postrotate

/usr/bin/killall -HUP httpd

endscript

}

/var/log/news/\* { monthly rotate 2 olddir /var/log/news/old missingok postrotate

kill -HUP ‘cat /var/run/inn.pid‘

endscript

nocompress

}

• **Tmpwatch**

tmpwatch - removes files which haven’t been accessed for a period of time

SYNOPSIS

tmpwatch [-u|-m|-c] [-MUadfqstvx] [--verbose] [--force] [--all] [--nodirs] [--nosymlinks] [--test] [--fuser] [--quiet]

[--atime|--mtime|--ctime] [--dirmtime] [--exclude path]

[--exclude-user user] time dirs

DESCRIPTION

tmpwatch recursively removes files which haven’t been accessed for a given time. Normally, it’s used to clean up directories which are used for temporary holding space such as /tmp.

OPTIONS

-u, --atime

Make the decision about deleting a file based on the file’s atime (access time). This is the default.

Note that the periodic updatedb file system scans keep the atime of directories recent.

-m, --mtime

Make the decision about deleting a file based on the file’s mtime (modification time) instead of the atime.

-c, --ctime

Make the decision about deleting a file based on the file’s ctime (inode change time) instead of the atime; for directories, make the decision based on the mtime.

-M, --dirmtime

Make the decision about deleting a directory based on the directory’s mtime (modification time) instead of the atime; completely ignore atime for directories.

-a, --all

Remove all file types, not just regular files, symbolic links and directories.

-d, --nodirs

Do not attempt to remove directories, even if they are empty.

-f, --force

Remove files even if root doesn’t have write access (akin to rm -f).

-l, --nosymlinks

Do not attempt to remove symbolic links.

-q, --quiet

Report only fatal errors.

-s, --fuser

Attempt to use the "fuser" command to see if a file is already open before removing it. Not enabled by default.

Does help in some circumstances, but not all. Dependent on fuser being installed in /sbin. Not supported on HP- UX or Solaris.

-t, --test Don’t remove files, but go through the motions of removing them.

# 6.BACKUP(tar, cpio, dd etc.)

• **tar**

NAME

tar - manual page for tar 1.23

SYNOPSIS

tar [OPTION...] [FILE]...

DESCRIPTION

GNU ‘tar’ saves many files together into a single tape or disk archive, and can restore individual files from the archive.

EXAMPLES

tar -cf archive.tar foo bar

# Create archive.tar from files foo and bar.

tar -tvf archive.tar

# List all files in archive.tar verbosely.

tar -xf archive.tar

# Extract all files from archive.tar.

Main operation mode:

-A, --catenate, --concatenate append tar files to an archive

-c, --create

create a new archive

-d, --diff, --compare

find differences between archive and file system

--delete

delete from the archive (not on mag tapes!)

-r, --append append files to the end of an archive

-t, --list

list the contents of an archive

--test-label

test the archive volume label and exit

-u, --update

only append files newer than copy in archive

Operation modifiers: --check-device

check device numbers when creating incremental archives

(default)

-g, --listed-incremental=FILE

handle new GNU-format incremental backup

-G, --incremental

handle old GNU-format incremental backup

--ignore-failed-read

do not exit with nonzero on unreadable files

--level=NUMBER

dump level for created listed-incremental archive

-n, --seek

archive is seekable

--sparse-version=MAJOR[.MINOR]

set version of the sparse format to use (implies --sparse)

-S, --sparse

handle sparse files efficiently

Overwrite control:

-k, --keep-old-files

don’t replace existing files when extracting, treat them as errors

--skip-old-files

don’t replace existing files when extracting, silently skip over

them

--keep-newer-files

don’t replace existing files that are newer than their archive copies

• **cpio**

NAME

cpio - copy files to and from archives

SYNOPSIS

Copy-out mode

In copy-out mode, cpio copies files into an archive. It reads a list of filenames, one per line, on the standard input, and writes the archive onto the standard output.

cpio {-o|--create} [-0acvABLV] [-C bytes] [-H format] [-M message] [-O [[user@]host:]archive] [-F [[user@]host:]archive] [-file=[[user@]host:]archive]

[--format=format] [--message=message][--null] [--reset-accesstime] [--verbose] [--dot] [--append] [--block-size=blocks] [-dereference]

[--io-size=bytes] [--rsh-command=command] [--help] [--version]

< name-list [> archive]

Copy-in mode

In copy-in mode, cpio copies files out of an archive or lists the archive contents. It reads the archive from the standard input.

cpio {-i|--extract} [-bcdfmnrtsuvBSV] [-C bytes] [-E file] [-H format] [-M message] [-R [user][:.][group]] [-I

[[user@]host:]archive] [-F

[[user@]host:]archive] [--file=[[user@]host:]archive] [--makedirectories] [--nonmatching] [--preserve-modification-time] [-numeric-uid-gid] [--rename]

[-t|--list] [--swap-bytes] [--swap] [--dot] [--unconditional] [-verbose] [--block-size=blocks] [--swap-halfwords] [--io-size=bytes]

[--pattern- file=file] [--format=format] [--owner=[user][:.][group]] [--nopreserve-owner] [--message=message] [--force-local] [--no-absolutefilenames] [--abso- lute-filenames] [--sparse] [--only-verify-crc] [--to-stdout] [--quiet]

[--rsh-command=command] [--help] [--version] [pattern...] [< archive]

Copy-pass mode

In copy-pass mode, cpio copies files from one directory tree to another, combining the copy-out and copy-in steps without actually using an archive. cpio {-p|--pass-through} [-0adlmuvLV] [-R [user][:.][group]] [-null] [--reset-access-time] [--make-directories] [--link] [--quiet] [-preserve-modifi- cation-time] [--unconditional] [--verbose] [--dot] [--dereference] [-owner=[user][:.][group]] [--no-preserve-owner] [--sparse] [--help]

[--version]

destination-directory < name-list

DESCRIPTION

GNU cpio is a tool for creating and extracting archives, or copying files from one place to another. It handles a number of cpio formats as well as reading and writing tar files.

OPTIONS

‘-0, --null’

Read a list of filenames terminated by a null character, instead of a newline, so that files whose names contain newlines can be archived. GNU find is one way to produce a list of null-terminated filenames.

This option may be used in copy-out and copy-pass modes.

‘-a, --reset-access-time’

Reset the access times of files after reading them, so that it does not look like they have just been read.

‘-A, --append’

Append to an existing archive. Only works in copy-out mode. The archive must be a disk file specified with the -O or -F (-file) option.

‘-b, --swap’

Swap both halfwords of words and bytes of halfwords in the data

‘-B’ Set the I/O block size to 5120 bytes. Initially the block size is 512 bytes.

‘--block-size=BLOCK-SIZE’

Set the I/O block size to BLOCK-SIZE \* 512 bytes.

‘-c’ Identical to “-H newc”, use the new (SVR4) portable format. If you wish the old portable (ASCII) archive format, use “-H odc” instead.

‘-C IO-SIZE, --io-size=IO-SIZE’

Set the I/O block size to IO-SIZE bytes.

‘-d, --make-directories’

Create leading directories where needed.

‘-E FILE, --pattern-file=FILE’

Read additional patterns specifying filenames to extract or list from FILE.

‘-H FORMAT, --format=FORMAT’

Use archive format FORMAT. The valid formats are listed below; the same names are also recognized in all-caps.

‘odc’ The old (POSIX.1) portable format.

‘newc’ The new (SVR4) portable format, which supports file systems having more than 65536 i-nodes.

‘crc’ The new (SVR4) portable format with a checksum added.

‘tar’ The old tar format.

‘-k’ Ignored; for compatibility with other versions of cpio.

‘-l, --link’

Link files instead of copying them, when possible.

‘-L, --dereference’

Copy the file that a symbolic link points to, rather than the symbolic link itself.

‘-t, --list’

Print a table of contents of the input.

‘--to-stdout’

Extract files to standard output. This option may be used in copy-in mode.

‘--version’

Print the cpio program version number and exit.

• **dd**

NAME

dd - convert and copy a file

SYNOPSIS

dd [OPERAND]...

dd OPTION

DESCRIPTION

Copy a file, converting and formatting according to the operands.

bs=BYTES

read and write BYTES bytes at a time (also see ibs=,obs=)

cbs=BYTES

convert BYTES bytes at a time

conv=CONVS

convert the file as per the comma separated symbol list

count=BLOCKS

copy only BLOCKS input blocks

ibs=BYTES

read BYTES bytes at a time (default: 512)

if=FILE

read from FILE instead of stdin

iflag=FLAGS

read as per the comma separated symbol list

obs=BYTES

write BYTES bytes at a time (default: 512)

of=FILE

write to FILE instead of stdout

oflag=FLAGS

write as per the comma separated symbol list

BLOCKS and BYTES may be followed by the following multiplicative suf-

fixes: c =1, w =2, b =512, kB =1000, K =1024, MB

=1000\*1000, M

=1024\*1024, xM =M GB =1000\*1000\*1000, G

=1024\*1024\*1024, and so on for

T, P, E, Z, Y.

Each CONV symbol may be:

ascii from EBCDIC to ASCII

ebcdic from ASCII to EBCDIC

ibm from ASCII to alternate EBCDIC

block pad newline-terminated records with spaces to cbs-size

unblock

replace trailing spaces in cbs-size records with newline

lcase change upper case to lower case

physically write output file data before finishing

directory

fail unless a directory sync likewise, but also for metadata

nonblock

use non-blocking I/O

Sending a USR1 signal to a running ‘dd’ process makes it print I/O statistics to standard error and then resume copying.

$ dd if=/dev/zero of=/dev/null& pid=$!

$ kill -USR1 $pid; sleep 1; kill $pid

18335302+0 records in 18335302+0 records out

9387674624 bytes

(9.4 GB) copied, 34.6279 seconds, 271 MB/s

# 7. PRINTER SETUP

In the case of Linux there are two types of printing systems are: BSD printing system and system5. Printing services are handled by a daemon called lpd (line printer daemon). Print jobs are kept into print spooler.

Spooling system has following parts:

* User commands: To start printing
* Queues: store and process print jobs
* Spooling directory: Temporary storage to hold pending jobs. Spool directory must be equal to number of devices connected.
* Server processes: Transfer print jobs from spooling

directory to device. Lpd: Transfer spooler to actual device. Number of lpd must be equal to the number of spool directory plus one.

This one for the master lpd.

Lpdtest: Testing the printer to work correct or not.

* Administrative commands: Starts and stop print jobs To start lpd daemon we use syntax

Lpd[-l] port

-l: Start a logging processes that copy. One of the important parts of the daemon start up procedure is to read the printer configuration file /etc/printcap.

/etc/printcap

Each line represents a printer. This file is used to identify instructions for communication with all printers that are configured and attached to the system. An entry in the printcap file is shown below,

-ljet|lp|ps|postscript|600 dpi 20 ms memory|local|LPT1:||

:lp=/dev/lp0:rw||

:sd=/var/spool/lpd/ljet 4:mx # 0:mc # 0:pl # 72:|

:pw # 85|

:if=/var/log/lpd-errs:if=/usr/local/cap/ljet 4

/etc/printcap contains

1. #(bash mark or pound signs): Identifies comments anywhere in the information in the first column.
2. end with ‘/’: represents line continue with next line.
3. ‘:’:Separate each field.
4. First field contains the name of the variable
5. There must be at least one printer entry in printcap
6. Printer with lp as treated as default printer
7. Printer attached to local system there must be a field

lp=/dev/lp0/:rw

1. Printcap includes so many variables

Printcap variables

Mainly three types of variables are used in printcap.

1. variable assign value using “=”.
2. Assign value using # (numeric) 3. Boolean variable only set Mainly used variables are:
3. sd: Spool directory

Specify spool directory.eg:sd=/var/spool/lpd/ljet.

1. lp: local printer assign local printer device. Contain in /dev/lp0
2. lf:log file: to store error messages
3. rw: read/write open lpd as read/write mode
4. mx: Maximum size of a print job assign zero indicates job at any size
5. if: input filter used to format prints. Default location is /usr/sbin/lpt
6. of: output filter used to format outputs
7. br: baud rate:

exists in the case where local printer connected to serial device

1. pl: page length:

specify the number of lines in a page. Eg: pl # 66 means 66 line/page

1. pw: page width:

specify number of characters in a line. Eg: pw # 85 means 85 characters /line

1. px: number of pixels used in the X-axis
2. py: number of pixels used in the Y-axis
3. sh: suppress header-suppress printing of header
4. rp: remote printer-specify name of the remote printer to be used
5. rm: remote machines-specify ID of the remote machine.
6. rs: restricted access to local printers to those users with account on the system
7. rg: specify restricted group to access printer
8. mc:specifies maximum number of copies
9. sc: single copy

Manage Print Queue

We can manage print queue using three commands:

* Lpq
* Lprm
* Lpc

1. lpq

Display the contents of a queue and status of the printer.

Syntax

Lpq[l][-p printer name][jobid…][user name]

Lpq-Display information about default printer

Lpq –l- more information about default printer such as user, jobid, postion in job queue, title bring printed and status of the file.

Lpq[-l][-p printer name]- Display information about printer specified

Lpq user name – Display the jobs present for a particular user.

1. Lprm

* Remove files from the queue
* User or administrator can remove Syntax

Lprm[-p printer name][-][jobid] user name

Remove all files which is issued by user

Delete all files of all users if no option is specified

3. Lpc

* Line printer control command
* Activate and deactivate printer queue
* Rearrange queue
* Status of any printer
* Lpc can be run on command line or interactive link

# 8.SETTING HOTNAME AND IP ADDRESS

Setting the hostname

NAME

hostname - show or set the system’s host name domainname - show or set the system’s NIS/YP domain name dnsdomainname - show the system’s DNS domain name nisdomainname - show or set system’s NIS/YP domain name ypdomainname - show or set the system’s NIS/YP domain name

SYNOPSIS

hostname [-v] [-a] [--alias] [-d] [--domain] [-f] [--fqdn] [-A] [--all- fqdns] [-i] [--ip-address] [-I] [--all-ip-addresses] [--long] [-s]

[--short] [-y] [--yp] [--nis]

hostname [-v] [-F filename] [--file filename] [hostname]

domainname [-v] [-F filename] [--file filename] [name]

nodename [-v] [-F filename] [--file filename] [name]

hostname [-v] [-h] [--help] [-V] [--version]

dnsdomainname [-v] nisdomainname [-v]

ypdomainname [-v]

DESCRIPTION

Hostname is the program that is used to either set or display the cur-

rent host, domain or node name of the system. These names are used by many of the networking programs to identify the machine. The domain name is also used by NIS/YP.

OPTIONS

-a, --alias

Display the alias name of the host (if used).

-d, --domain

Display the name of the DNS domain. Don’t use the command

domainname to get the DNS domain name because it will show the

NIS domain name and not the DNS domain name. Use dnsdomainname instead.

-F, --file filename

Read the host name from the specified file. Comments

(lines starting with a ‘#’) are ignored. -f, --fqdn, --long

Display the FQDN (Fully Qualified Domain Name).

-A, --all-fqdns

Displays all FQDNs of the machine.

-h, --help

Print a usage message and exit.

-i, --ip-address

Display the IP address(es) of the host.

-I, --all-ip-addresses

Display all network addresses of the host.

-s, --short

Display the short host name. This is the host name cut at the first dot.

-V, --version

Print version information on standard output and exit success- fully.

-v, --verbose

Be verbose and tell what’s going on.

-y, --yp, --nis

Display the NIS domain name. If a parameter is given (or -file name ) then root can also set a new NIS domain.

FILES

/etc/hosts /etc/sysconfig/network

NOTE

Note that hostname doesn’t change anything permanently. After reboot original names from /etc/hosts are used again.

Setting the IP address

NAME

ifconfig - configure a network interface

SYNOPSIS

ifconfig [interface] ifconfig interface [aftype] options | address ...

NOTE

This program is obsolete! For replacement check ip addr and ip link. For statistics use ip -s link.

DESCRIPTION

Ifconfig is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed.

If no arguments are given, ifconfig displays the status of the currently active interfaces. If a single interface argument is given, it displays the status of the given interface only; if a single -a argument is given, it displays the status of all interfaces, even those that are down. Otherwise, it configures an interface.

OPTIONS

interface

The name of the interface up

This flag causes the interface to be activated. down

This flag causes the driver for this interface to be shut down. [-]arp

Enable or disable the use of the ARP protocol on this interface.

[-]promisc

Enable or disable the promiscuous mode of the interface.

[-]allmulti

Enable or disable all-multicast mode metric N

This parameter sets the interface metric. It is not available under GNU/Linux.

mtu N

This parameter sets the Maximum Transfer Unit (MTU) of an interface.

dstaddr addr

Set the remote IP address for a point-to-point link (such as PPP).

netmask addr

Set the IP network mask for this interface.

add addr/prefixlen

Add an IPv6 address to an interface.

del addr/prefixlen

Remove an IPv6 address from an interface. tunnel ::aa.bb.cc.dd

Create a new SIT (IPv6-in-IPv4) device, tunnelling to the given destination.

irq addr

Set the interrupt line used by this device. io\_addr addr

Set the start address in I/O space for this device. mem\_start addr

Set the start address for shared memory used by this device.

# 9.TCP/ IP CONFIGURAION

* Setting the Hostname

[root@DBCPC Desktop]# hostname

DBCPC

[root@DBCPC Desktop]# hostname DBCPC22

[root@DBCPC Desktop]# hostname

DBCPC22

* Setting the IP Address

[root@DBCPC Desktop]# ifconfig eth0 192.168.10.22

[root@DBCPC Desktop]# ifconfig

eth0 Link encap:Ethernet HWaddr 00:27:0E:20:EF:23 inet addr:192.168.10.22 Bcast:192.168.10.255 Mask:255.255.255.0

inet6 addr: fe80::227:eff:fe20:ef23/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500

Metric:1

RX packets:18561 errors:0 dropped:0 overruns:0 frame:0 TX packets:5334 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000

RX bytes:8833618 (8.4 MiB) TX bytes:828078 (808.6 KiB)

lo Link encap:Local Loopback inet addr:127.0.0.1 Mask:255.0.0.0

inet6 addr: ::1/128 Scope:Host

UP LOOPBACK RUNNING MTU:16436 Metric:1

RX packets:1127 errors:0 dropped:0 overruns:0 frame:0

TX packets:1127 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:0

RX bytes:71752 (70.0 KiB) TX bytes:71752 (70.0 KiB)

OR

By Editing /etc/sysconfig/network-scripts/ifcfg-eth0 file.

[root@DBCPC Desktop]# vi /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0

HWADDR=00:27:0E:20:EF:23

TYPE=Ethernet

#UUID=4a89e6f5-d233-4e30-a91c-6dc54e401e95

ONBOOT=yes

#DNS1=192.168.10.24

#DOMAIN=vipi.edu

NM\_CONTROLLED=yes

BOOTPROTO=static

IPADDR=192.168.10.22

NETWORK=192.168.10.0

NETMASK=255.255.255.0

BROADCAST=192.168.10.255

#GATEWAY=192.168.10.29

USERCTL=no

PEERDNS=yes

IPV6INIT=no

• Setting the route

[root@DBCPC Desktop]# route add -net 192.168.10.0 netmask

255.255.255.0 dev eth0

[root@DBCPC Desktop]# service network restart

Shutting down interface eth0: Device state: 3 (disconnected)

[ OK ]

Shutting down loopback interface: [ OK ] Bringing up loopback interface: [ OK ] Bringing up interface eth0:

# 10.TELNET CONFIGURATIONS

NAME

telnet - user interface to the TELNET protocol

SYNOPSIS

telnet [-8EFKLacdfrx] [-X authtype] [-b hostalias] [-e escapechar]

[-k realm] [-l user] [-n tracefile] [host [port]]

DESCRIPTION

The telnet command is used to communicate with another host using the TELNET protocol. If telnet is invoked without the host argument, it enters command mode, indicated by its prompt (telnet>).

The options are as follows:

-7

Strip 8th bit on input and output. Telnet is 8-bit clean by default but doesn’t send the TELNET BINARY option unless forced.

-8

Specifies an 8-bit data path. This causes an attempt to negotiate the TELNET BINARY option on both input and output.

-E

Stops any character from being recognized as an escape character.

-F

If Kerberos V5 authentication is being used, the -F option allows the local credentials to be forwarded to the remote system, including any credentials that have already been forwarded into the local environment.

-K

Specifies no automatic login to the remote system

-L

Specifies an 8-bit data path on output. This causes the BINARY option to be negotiated on output.

-X atype

Disables the atype type of authentication.

-a

Attempt automatic login. Currently, this sends the user name via the USER variable of the ENVIRON option if supported by the remote system.

-f

If Kerberos V5 authentication is being used, the -f option allows the local credentials to be forwarded to the remote system.

-l user

When connecting to the remote system, if the remote system understands the ENVIRON option, then user will be sent to the remote system as the value for the variable USER.

-n tracefile

Opens tracefile for recording trace information. See the set tracefile command below.

-r

Specifies a user interface similar to rlogin(1). In this mode, the escape character is set to the tilde (~) character, unless modified by the

-e option.

-x

Turns on encryption of the data stream if possible.

disable type

Disables the specified type of authentication enable type

Enables the specified type of authentication status

Lists the current status of the various types of authentication.

close

Close a TELNET session and return to command mode. display argument [...]

Displays all, or some, of the set and toggle values (see below).

encrypt argument [...]

The encrypt command manipulates the information sent through the TELNET ENCRYPT option.

logout

Sends the TELNET LOGOUT option to the remote side. This command is similar to a close command.

**OUTPUT**

[root@DBCPC22 Desktop]# chkconfig telnet on [root@DBCPC22 Desktop]# telnet 192.168.10.11 Trying 192.168.10.11...

Connected to 192.168.10.11.

Escape character is '^]'.

Red Hat Enterprise Linux Server release 6.4 (Santiago) Kernel 2.6.32-358.el6.x86\_64 on an x86\_64

login: shilpa

Password:

Last login: Wed Dec 2 15:24:30 from DBCPC12

[shilpa@DBCPC11 ~]$ ls

Desktop Downloads Music Public Untitled document

Documents h Pictures Templates Videos

[shilpa@DBCPC11 ~]$ logout

Connection closed by foreign host.

[root@DBCPC22 Desktop]#

**11.CONFIGURING NETWORK INFORMATION SERVICE(NIS)**

The NIS is any administrative database that provides central control and automatic dissemination of important administrative files. NIS converts several standard UNIX files into databases that can be queried over the network. The databases are called NIS maps. Some maps are created from files that are used for system adminstrations, such as the password file *(/etc/passwd*) and the group file *(/etc/group*). Others are derived from files related to network administrations.

/etc/ethers

Creates the NIS maps ethers. byaddr and ethers. byname .

/etc/hosts

Produces the maps hosts. byname and hosts.byaddr.

/etc/networks

Produces the maps networks. byname and host.byaddr.

/etc/protocols

Produces a single maps called service. byname.

/etc/aliases

Defines electronic mail aliases and produces the maps mail. aliases and mail.byaddr.The advantage of using NIS is that these important administrative files can be maintained on a central server and yet completely accessible to every workstation on the network. All of the maps are stored on a master server that runs the NIS server processes *ypserv*. The maps are queried remotely by client systems. Clients run *ypbind* to locate the server.

The NIS server and its client are a NIS domain–a term NIS shares with DNS. The NIS domain is identified by a domain name. The only requirements or the name is that different NIS domains accessible through the same local networks must have different names.

NIS uses its domain name to create a directory within */var/yp* where the NIS maps are stored.

*ypinit* builds the domain sub directory of */var/yp* for the current default domain after building the domain sub directory, *ypinit* builds a complete set of the syntax of *ypinit* is

/usr/lib/yp/ypinit [ -m ] [ -s master\_name ]

Options

-m - If the local host is the NIS master:

-s - Set up a slave server with the database from masre\_name

NIS Server Configurations

Configured NFS

1. Setup
   1. Disable named

Activate portmap and ypserv

1. run/bin/domainname mca
2. run/usr/lib/yp/ypinit-m

[root @ localhost yp]# /usr/lib/yp/ypinit-m

At this point, we have to construct a list of the hosts, which will run NIS servers. localhost.localdomain is in the list of NIS server hosts. Please continue to add the names for the other hosts, one per line. When you are done with the list, type < control D>.

next host to add: localhost.localdomain

next host to add:

The current list of NIS servers look like this:

localhost.localdomain

is this correct? [y/n: y ] y

We need a few minutes to build the database….

Building /var/yp/mca/ypservers…

Running /var/yp/Makefile…

gmake [1]: Entering directory ‘/var/yp/mca’

Updating passwd.byname…

Updating passwd.byuid….

Updating group.byname…

Updating group.bygid…

Updating hosts.byname…

gmake[1]: Leaving directory ‘/var/yp/mcaa’

localhost.localdomain has been set up as a NIS master server.

Now you can run ypinit –s locahost.localdomain on all slave server.

1. [root@localhost yp]# service portmap restart
2. [root@localhost yp]# service nfs restart
3. [root@localhost yp]# service ypserv restart

Clients configuration

1. Setup

Authentication Configuration

Enter the domain name and server address

[\*] Use NIS

Domain: mca

Server: 192.168.1.100

1. Mount the server home directory.

Edit /etc/fstab

192.168.1.100:/home nfs defaults 00

1. mount –a
2. Service ypbind restart
3. Login with new username and password to the server system.

RESULT:

The network was configured with NIS and was able to login to the system.

**12.CONFIGURING DHCP**

DHCP stands for Dynamic Host Configuration Protocol .It is used to control running clients with the help of server .It dynamically distributes IP address and configuration to clients with the help of server. DHCP was developed to alleviate many of the following shortcomings of BOOTP and to accomplish the following:

* DHCP allows administrators to control configuration parameters on their network.
* Clients using DHCP can be dynamically configured. This allows additions and changes to networks without the need to visit each individual host or workstation.
* DHCP provides a dynamic database for IP address allocation. These IP addresses, when no longer in use, can be reclaimed via lease.
* For fault tolerance, multiple DHCP servers can service one or more subnet.

DHCP provides three important features:

1. Backward compatibility:

A DHCP server can support BOOTP clients. Properly configured; a DHCP server can support all the clients.

2. Full configurations:

A DHCP server provides complete set of TCP/IP configuration parameters. The network administrator can handle the entire configurations.

3. Dynamic address assignments

A DHCP server can provide permanent addresses manually, permanent automatically, and temporary address dynamically. The network administrator can tailor the type of address to the needs of the network and the client system

DHCP server setup

Linux provides DHCP server daemon called dhcpd.In setup enable dhcpd daemon .The configuration information of DHCP contained in the file

/etc/dhcpd.conf. Normally it has to be created in the /etc directory.

The contents of /etc/dhcpd.conf file will look like

default – lease – time 600;

max – lease – time 7200;

get – lease – hostnames true;

Option – subnet – mask 255.255.255.0;

Option – domain – name – server 172.16.12.1,172.16.3.5;

Option – lpr – servers 172.16.12.1;

Option – interface – mtu 1500

Subnet 172.16.13.0 net mask 255.255.0.0

{

Option routers 172.16.3.25;

Option broadcast - address 172.16.3.255;

Range 172.16.3.50 172.16.3.250;

}

Subnet 172.16.12.0 net mask 255.255.0.0

{

Range 172.16.12.64 172.16.12.192;

Range 172.16.12.200 172.16.12.250;

}

Host peanut

{

Hardware Ethernet 08:80:20:01:59:c3;

Fixed address 172.16.12.2;

}

ILLUSTRATION: DHCP SERVER SETUP

First,we have to create DHCP daemon configuration file. So we edit /etc/dhcp.conf file.The main thing we have to do here is assign an IP address to a client randomly. This done setting the /etc/dhcp.conf as:

ddns-update-style ad-hoc;

default-lease-time 600;

max-lease-time 7200;

option subnet-mask 255.255.255.0;

option broadcast-address 192.168.1.255;

Subnet 192.168.1.0 netmask 255.255.255.0

{

Range 192.168.1.50 192.168.1.250;

}

Host nice

{

Hardware Ethernet 00:40:f4:31:c9:9e;

Fixed-address 192.168.1.2;

}

This will result in DHCP server assigning a client an IP address from the range 192.168.1.50 to 192.168.1.250.The lease for the IP address will be 600 seconds and maximum lease time will be 7200 seconds.The server will tell the client to use the subnet mask as 255.255.255.0 and broadcast address as 192.168.1.255.

Also, we can assign specific IP address to the clients based on their hardware Ethernet address.

The above code will assign IP address 192.168.1.2 to the client with hardware Ethernet address 00:40:f4:31:c9:9e.

After editing the file, we have to restart the DHCP server.This is done by issuing the command service dhcpd restart. When the dhcpd server start , it will create a file called dhcpd.leases in the /var/lib/dhcpd directory. if the file is not automatically created ,we have to create a file and save it. The file will have the entries like

lease 192.168.1.61 {

starts 2 2006/02/21 06:21:14;

ends 2 2006/02/21 06:31:14;

tstp 2 2006/02/21 06:31:14;

binding state free;

hardware ethernet 00:40:f4:31:b2:6b;

}

#### ILLUSTRATION: DHCP CLIENT

To make system as DHCP client, we have to edit the file /etc/sysconfig/network-scripts/ifcfg-eth0.In the file, we remove entries IPADDR,NETMASK and BROADCAST .Then ,we have to assign BOOTPROTO to DHCP. Then the will look like

DEVICE =etho

ONBOOT=yes

BOOTPROTO=DHCP

After saving the file and rebooting the system, the server will supply an IP address. To check the IP address ,we use the ifconfig command.

eth0 Link encap:Ethernet HWaddr 00:40:f4:31:c9:9e;

inet addr:192.168.1.2 Bcast:192.168.1.255 Mask:255.255.255.0

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:402 errors:0 dropped:0 overruns:0 frame:0

TX packets:19 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:100

RX bytes:57508 (56.1 Kb) TX bytes:1140 (1.1 Kb)

Interrupt:5 Base address:0xc00

lo Link encap:Local Loopback

inet addr:127.0.0.1 Mask:255.0.0.0

UP LOOPBACK RUNNING MTU:16436 Metric:1

RX packets:664 errors:0 dropped:0 overruns:0 frame:0

TX packets:664 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

RX bytes:44941 (43.8 Kb) TX bytes:44941 (43.8 Kb)

##### RESULT

A system was configured as server and a set of system as DHCP clients. It was verified that the DHCP server assigned an IP address to the clients within the range specified. Fixed addressing also was verified.

**13.CONFIGURING SAMBA SERVER**

Samba is a suite of Unix application that speaks the SMB [Server Message Block] protocol. SMB, which stands for Server Message Block, is a protocol for sharing files, printers, and serial ports, and communications abstractions such as named pipes and mail slots between computers. SMB is a client Server, request-response protocol. By supporting this protocol, samba allows Unix Servers to communicate with the same networking protocol as Microsoft Windows Products. Samba effectively allows using a Linux Server as a network server for group of windows machines operating on window network. The can share files between them.

We can do these things with Samba

* Share directories, printers and CD-ROM drives on a Linux machine with Windows machines.

# Access an SMB share (any storage shared on a Windows host) or a printer on a Windows client with Linux machines.

# Samba Daemons:

The Samba suite revolves around a pair of Unix daemons that provide shared resources to SMB clients on the network. These daemons are:

smbd

A daemon that allows file and printer sharing on an SMB network and provides authentication and authorization for SMB clients. It handles all the notifications between the samba server and the network clients. In addition, it is responsible for user authentication, resource locking and data sharing through the smb protocol.

nmbd

A daemon that looks after the Window Internet Name Service (WINS), and assist with browsing. This daemons listens for name server requests and provides appropriate information when called upon

Samba Configuration Files:

smb.conf, the configuration file for Samba. By default, the smb.conf is in /etc/samba/

This file is separated into two basic parts. One for global option and other for shared services. A shared service can either be a file space service or printable service. The file-space service is a directory to which clients are given access and use the space in it as an extension of their local file system. A printable service provides access by events to print services, such as printers managed by the Samba Server.

The Samba configuration file structure is as shown below

[global]

---------

[homes]

----------

[printers]

----------

[tmp]

Names inside the square brackets define the unique sections. The [global] section defines a few variables that Samba will use to define sharing for all resources. The [homes] section allows a remote user to access their home directory on the local (Linux) machine. The [printer] section allows sharing of printers. The [tmp] section allow share of files. The remaining are individual configuration options unique to that service. These options will continue until a new-bracketed section encounters, or until end of the file is reached.

This file can be edited as per the requirements of the user. Many entries are commented using a semicolon or a #. This can be removed to make the lines effective. smbpasswd, is the file stores passwords for various users added to Samba. By default, it is in /etc/samba/ . The log file for Samba is /var/log/samba/smb.log

## Setting up a Samba Server

Modify the existing smb.conf . A sample configuration file is shown below:

# global section influences all other services

[global]

#specify netbios name

netbios name = thunder

# current\_machine’s IP address

interfaces = 192.168.1.29

server string = Samba server

# Allowable networks for accessing this machine

hosts allow = 192.168.1. 192.168.2. 127.

log file = /var/log/samba/smb.log

smb passwd file = /etc/samba/smbpasswd

max log size = 10

# Workgroup for windows client

workgroup = WORKGROUP

guest account = nobody

# Connection validity checking interval

keep alive = 10

encrypt passwords = yes

# Registered users of Samba Server

valid users = louis root

# Printing Type

printing = bsd

printcap name = /etc/printcap

load printers = yes

print command = /usr/bin/lpr -r -P%p %s

lpq command = /usr/bin/lpq -P%p

lprm command = /usr/bin/lprm -P%p %j

# Ignores case

case sensitive = no

socket options = TCP\_NODELAY SO\_RCVBUF=8192 SO\_SNDBUF=8192

# service: allows access to home directories of users

[homes]

comment = Home directories

valid users = %S

browseable = no

writable = yes

# Allows writable access

create mode = 0666

directory mode = 0777

# service: allows access to a directory specified

[tmp]

comment = Browse-Me

# Other directories shared

path = /samba\_share

browseable = no

read only = no

writable = yes

public = yes

create mode = 0666

# service: allows access to optical drive

[cdrom]

comment = Linux Optical Drive

path = /mnt/cdrom

read only = yes

# Allows simultaneous access from various windows clients on cd-rom drive

locking = no

# service: allows access to printers connected in Linux

[printers]

comment = All Printers

path = /var/spool/samba

browseable = no

printable = yes

writable = no

*NOTE*: For sharing a printer, you must first configure your printer on the Linux machine using the printtool utility. Issue the following commands:

testparm configfilename *filename*

To Test validity of smb.conf file using test parser. Here you will get a message saying all the sections in the smb.conf file are processed and allowed the requested connections to above said sections

service *smb* start

For starting SMB and NMB services

smbpasswd [ -a *username* ]

Adds a new user for Samba. If the given user already exists, this simply changes his/her password

nmblookup *hostname*

For mapping machine name to IP address, you have to supply hostname

* Getting the Windows machine from our Linux machine on the LAN

smbclient -s *samba\_configuration-file* -U *smb\_username* -W *workgroup* -L *hostname*

smbclient can communicate to an SMB/CIFS server. This is for testing Samba daemons. Here you will get a message saying the interface of the Samba Server machine is added, and a positive response is received from the Samba Server machine. Then you will be asked for your password. This displays summary of requested services and configuration setup with available Samba Servers on the network along with the Server on the current machine.

mount -t smbfs -o username=*name* //*windows\_host*/*share\_directory* /*target directory*

This is for mounting a remote samba share into the specified directory on our Linux machine. You will be again asked your password. With –t specify file system as smbfs. Username=name allows an ordinary Samba user to mount the file system. Likewise you may mount any storage (can also be a removable storage) to specified directory.

You can use printtool utility to detect, configure and install driver for the printer connected to a Windows machine.

* Getting the Linux machine from a Windows machine on the LAN

Browse the ‘Network Neighborhood’ or ‘My Network Places’ on the Windows client. Search for the computer having the netbios name (normally hostname) of the your Linux machine on which you have setup the Samba server on the network. You may find it and will be asked the password for logging into the machine (in case where guests are not allowed). Provide the name of any registered valid Samba user and his password. After logging in you can have access to your home directory, other shared directories, optical drives (if any), and printers connected to this Linux machine.

**14. DOMAIN NAME SERVICE**

## INTRODUCTION

In networking, names as well as IP addresses are assigned to devices. IP addresses are four-byte integer number, each byte separated by a period (.). The name of the machine is a collection of characters. The hostname and the IP address can be used interchangeably. But network connection always takes place based on an IP address. The system converts hostname to an address before the network connection is established. Then the question arises why to assign both the name and an address to a host, would not an IP Address suffice? The answer lies in the fact that it is cumbersome to remember addresses for the human mind while it is easy to remember the names.

ZONE

The term zone is often used interchangeably with the word domain. But, it is a collection of domain information contained in a domain database file. This file that contains the domain information is called the zone file. Each name server communicates with the zone. Each zone has at least one name server responsible for knowing the address information for each machine within that zone.

SERVER CONFIGURATION

The daemon we use with DNS configuration is named. The following files have to be configured to configure named.

1) named.conf: Configuring this file sets the general named parameters and points to the sources of domain database. The selected zone also is configured here. The configuration statements of this file are the following

1. directory: Defines the directory for all subsequent file references
2. primary: Defines the server as primary for the specified zone
3. secondary: Defines the server as secondary for the specified zone
4. cache: points to the cache file
5. forwarders: Lists the servers to which the queries are forwarded.
6. options: Specifies the global options.

zone command

The zone command is used to specify domains that the nameserver will serve. The keyword zone should be followed by the domain name placed within double quotes. Several options can be placed in the zone block. After the domain name the class has to be specified. The class for Internet is IN. Several options can be placed within the zone statement. Two essential options are type and file. The type is used to specify the zone’s type and file is used to specify the name of the zone file to be used for that zone.

The type can be master, slave, stub, forward and hint. Master stands for the primary server; slave indicates that the server is a secondary server, which has to be updated periodically, stub zone copies only other name server entries, instead of the entire zone, a forward zone directs all queries to name servers specified by the forwarders statement and hint zone specifies the set of root name servers used by all Internet DNS servers.

e.g. zone “don.edu” in

{

type master;

file don.edu.zone

}

There are other statements within this file. They are acl, server, options and logging. The options statement defines global option and can be used once in the configuration file. Two important option statements commonly found are the directory option and the forwarders option. The directory option holds the location of the name server’s zone and the cache file on your system. The Forwarders option contains the list of name servers to which queries are forwarded if they cannot be resolved by the local DNS server.

e.g. options{

directory “/var/named”;

forwarders { 192.168.1.34;

192.168.1.47;

};

};

2) named.local: This file is used to locally reserve the loop back address

3) named.ca: This file contains all the root level domains.

4) named.host: This is the file that maps host names to the IP addresses.

5) named.rev: It is the zone file for the reverse domain that maps IP address to the host name

All the files except the named.conf contain information in the form of resource records. Resource Records are explained in detail below.

ILLUSTRATION FOR SETTING A DNS SERVER

The system that is selected as the server has the IP Address 192.168.1.17. The Domain name of the network is don.edu. The host name of the server machine is shijo. The two systems selected as the clients are 192.168.1.175 and 192.168.1.50. The domain names of the clients are shimna and litty respectively. Steps for configuration are listed below.

\* Login to the root

\* Activate the named daemon in the network setup

\* Edit the /etc/named.conf file

zone “don.edu” IN { type master; file “don.edu.zone”;

};

\* Create the zone file don.edu.zone in the /var/named directory

The file should contain the following statements

$TTL 86400

@ IN SOA don.edu. root.don.edu { 42 ; serial

3H ; refresh 15M ; retry 1N ; expire

1D ; minimum

IN NS DBCPC24 don.edu

DBCPC IN 192.168.10.24

\* Edit the file /etc/resolv.conf

nameserver 192.168.1.17

search don.edu.

\* Restart the daemon with the following command

# service named restart

ILLUSTRATION FOR SETTING A DNS CLIENT

\* Edit the file /etc/resolv.conf

nameserver 192.168.1.17

search don.edu.

\*Edit the file /etc/sysconfig/network scripts/ifconfig.eth0

DNS1 192.168.1.17

DOMAIN = don.edu

Service network restart

Service named restart

Ping DBCPC

DIG Command

This command is used to retrieve the host name with the –x option followed by the IP address of the system.

E.g., dig –x 192.168.1.175

This command will return the following information

•Question section: Which redisplays the queried IP address

•Answer Section: It gives the corresponding host name

•Authority section: It gives the server name

•Additional section: This displays the IP address of the server

The remote systems can be pinged using the host name supplied by the dig –x command.

RESULT :

One Host in the don.edu domain was configured as the DNS server and two hosts were configured as DNS clients. The Zone file in the server assigned the host names of the clients to the IP Addresses. And it was possible to communicate between the systems using the host name. DNS Reverse Mapping was configured successfully and tested using dig command.

**15.FTP CONFIGURATION**

# Example config file /etc/vsftpd/vsftpd.conf

#

# The default compiled in settings are fairly paranoid. This sample file

# loosens things up a bit, to make the ftp daemon more usable.

# Please see vsftpd.conf.5 for all compiled in defaults.

#

# READ THIS: This example file is NOT an exhaustive list of vsftpd options.

# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's

# capabilities.

#

# Allow anonymous FTP? (Beware - allowed by default if you comment this out).

anonymous\_enable=YES

#

# Uncomment this to allow local users to log in.

local\_enable=YES

#

# Uncomment this to enable any form of FTP write command.

write\_enable=YES

#

# Default umask for local users is 077. You may wish to change this to 022,

# if your users expect that (022 is used by most other ftpd's)

local\_umask=022

#

# Uncomment this to allow the anonymous FTP user to upload files. This only

# has an effect if the above global write enable is activated. Also, you will

# obviously need to create a directory writable by the FTP user.

#anon\_upload\_enable=YES

#

# Uncomment this if you want the anonymous FTP user to be able to create

# new directories.

#anon\_mkdir\_write\_enable=YES

#

# Activate directory messages - messages given to remote users when they

# go into a certain directory.

dirmessage\_enable=YES

#

# The target log file can be vsftpd\_log\_file or xferlog\_file.

# This depends on setting xferlog\_std\_format parameter

xferlog\_enable=YES

#

# Make sure PORT transfer connections originate from port 20 (ftp-data).

connect\_from\_port\_20=YES

#

# If you want, you can arrange for uploaded anonymous files to be owned by

# a different user. Note! Using "root" for uploaded files is not

# recommended!

#chown\_uploads=YES

#chown\_username=whoever

#

# The name of log file when xferlog\_enable=YES and xferlog\_std\_format=YES

# WARNING - changing this filename affects /etc/logrotate.d/vsftpd.log

#xferlog\_file=/var/log/xferlog

#

# Switches between logging into vsftpd\_log\_file and xferlog\_file files.

# NO writes to vsftpd\_log\_file, YES to xferlog\_file

xferlog\_std\_format=YES

#

# You may change the default value for timing out an idle session.

#idle\_session\_timeout=600

#

# You may change the default value for timing out a data connection.

#data\_connection\_timeout=120

#

# It is recommended that you define on your system a unique user which the

# ftp server can use as a totally isolated and unprivileged user.

#nopriv\_user=ftpsecure

#

# Enable this and the server will recognise asynchronous ABOR requests. Not

# recommended for security (the code is non-trivial). Not enabling it,

# however, may confuse older FTP clients.

#async\_abor\_enable=YES

#

# By default the server will pretend to allow ASCII mode but in fact ignore

# the request. Turn on the below options to have the server actually do ASCII

# mangling on files when in ASCII mode.

# Beware that on some FTP servers, ASCII support allows a denial of service

# attack (DoS) via the command "SIZE /big/file" in ASCII mode. vsftpd

# predicted this attack and has always been safe, reporting the size of the

# raw file.

# ASCII mangling is a horrible feature of the protocol.

#ascii\_upload\_enable=YES

#ascii\_download\_enable=YES

#

# You may fully customise the login banner string:

#ftpd\_banner=Welcome to blah FTP service.

#

# You may specify a file of disallowed anonymous e-mail addresses. Apparently

# useful for combatting certain DoS attacks.

#deny\_email\_enable=YES

# (default follows)

#banned\_email\_file=/etc/vsftpd/banned\_emails

#

# You may specify an explicit list of local users to chroot() to their home

# directory. If chroot\_local\_user is YES, then this list becomes a list of

# users to NOT chroot().

#chroot\_local\_user=YES

#chroot\_list\_enable=YES

# (default follows)

#chroot\_list\_file=/etc/vsftpd/chroot\_list

#

# When "listen" directive is enabled, vsftpd runs in standalone mode and

# listens on IPv4 sockets. This directive cannot be used in conjunction

# with the listen\_ipv6 directive.

listen=YES

#

# This directive enables listening on IPv6 sockets. To listen on IPv4 and IPv6

# sockets, you must run two copies of vsftpd with two configuration files.

# Make sure, that one of the listen options is commented !!

#listen\_ipv6=YES

pam\_service\_name=vsftpd

userlist\_enable=YES

tcp\_wrappers=YES

[root@DBCPC20 Desktop]# service vsftpd restart

Shutting down vsftpd: [ OK ]

Starting vsftpd for vsftpd: [ OK ]

You have new mail in /var/spool/mail/root

[root@DBCPC20 Desktop]# ftp 192.168.10.11

Connected to 192.168.10.11 (192.168.10.11).

220 (vsFTPd 2.2.2)

Name (192.168.10.11:root): anonymous

331 Please specify the password.

Password:

230 Login successful.

Remote system type is UNIX.

Using binary mode to transfer files.

ftp> ls

227 Entering Passive Mode (192,168,10,11,134,226).

150 Here comes the directory listing.

drwxr-xr-x 2 0 0 4096 Mar 02 2012 pub

226 Directory send OK.

ftp> exit

221 Goodbye.

**16. NETWORK FILE SYSTEM CONFIGURATIONS (NFS)**

The network file system is a distributed file system that provides transparent access to remote disks. NFS allows us to centralize administration of disks. Instead of duplicating common directories such as */usr/local* on every system, NFS provides a single copy of the directory that is shared by all systems on the network. To host running NFS, remote file systems are indistinguishable from local ones.

Setting up NFS

Setting up NFS on clients and servers involves starting the domains that handle the NFS RPC protocol, starting additional daemons for auxiliary services such as file locking and then simply exporting file systems from the NFS servers and mounting them on the clients.

On an NFS client, we need to have the *lockd* and *statd* daemons running in order to use NFS. These daemons are generally started in a boot script.

On an NFS server, NFS services are started with the *nfsd* daemon as well as the file locking daemons used on the client.

# Exporting file systems by the server

Usually a host decided to become an NFS server if it has file systems to export to the network. A server keeps a list of currently exported file system and associated access restrictions in a file and compares incoming NFS mount requests to entries in this table. It is up to the server to decide if the client can mount a file system.

The exported file system table is initialized from the */etc/exports* file. The super user may export other file system once the server is up and running, so the */etc/exports* file and the actual list of currently exported file systems, */etc/xtab* are maintained separately.

After the server system is up the super user can export additional file systems via the *exportfs* command. The *exportfs* command is used to maintain the current table of exported file systems for NFS.

Syntax of *exportfs* is

exportfs –a

Mounting file systems by the Client system

NFS client can mount any file system, or part of a file system. That has been exported from an NFS server. File system can be mounted explicitly using the mount command.

mount< nfs server IP addr>:<remote fs><local mount point>

The mount command assumes that the type is NFS if a host address appears in the device specification.

# Server setup

1. Activate nfs daemon

Setup : activate nfs

1. Export the file system
   1. Edit the file/etc/exports give the

Vi/etc/exports

/home/crab \*(rw)

/home/siraj \*(rw)

# Client setup

mount the remote file system

mount 192.168.1.120:/home/crab/mnt/nfs/crab

17.WEBSERVER CONFIGURATION

Server setup

\* Activate httpd daemon activate httpd

\* Activate the named daemon in the network setup

\* Edit the /etc/named.conf file

zone “don.edu” IN { type master; file “don.edu.zone”;

};

\* Create the zone file don.edu.zone in the /var/named directory

The file should contain the following statements

$TTL 86400

@ IN SOA don.edu. root.don.edu { 42 ; serial

3H ; refresh 15M ; retry

1N ; expire

1D ; minimum

IN NS DBCPC24 don.edu

DBCPC IN 192.168.10.24

\* Edit the file /etc/resolv.conf

nameserver 192.168.1.17

search don.edu.

\* Restart the daemon with the following command

# service named restart

[root@DBCPC6 Desktop]# vi /etc/httpd/conf/httpd.conf

Name VirtualHost 192.168.10.7

<VirtualHost 192.168.10.7>

Servername “dbc7.hari.edu”

DocumentRoot /home/httpd/html/dbc7

</VirtualHost>

[root@DBCPC6 Desktop]# cd /home

[root@DBCPC6 home]# mkdir httpd

[root@DBCPC6 home]# cd httpd

[root@DBCPC6 httpd]# mkdir html

[root@DBCPC6 httpd]# cd html

[root@DBCPC6 html]# mkdir dbc7

[root@DBCPC6 html]# cd dbc7

[root@DBCPC6 dbc7]# vi index.html

<html>

<body>

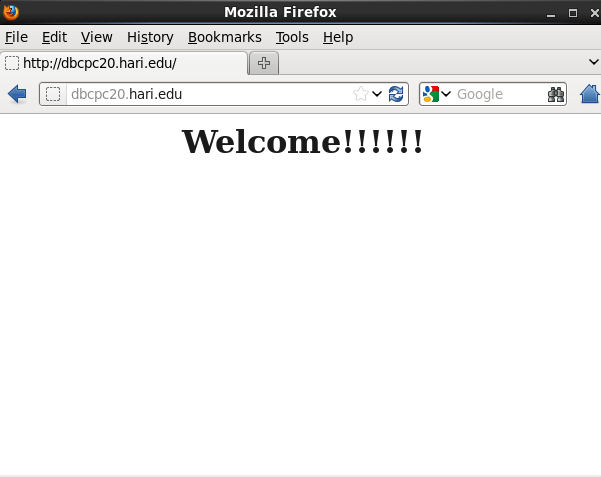
Welcome!!!!!!!

</body>

</html>

[root@DBCPC6 dbc7]#service httpd restart

OUTPUT



**18.TCP CHAT PROGRAM**

//chat server

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<sys/stat.h>

#include<netinet/in.h>

#include<netdb.h>

#include<stdio.h>

#define MYPORT 5009

#define BACKLOG 3

#define MAXBUF 100

int main(void)

{

int server\_sockfd,client\_sockfd;

int client\_len,len,i,result,cmp;

struct sockaddr\_in addr\_s,addr\_c;

char buf[MAXBUF];

char message[100];

printf("Press ^c to terminate\n");

if((server\_sockfd=socket(AF\_INET,SOCK\_STREAM,0))==-1)

{

printf("\nError in socket creation");

exit(1);

}

addr\_s.sin\_family=AF\_INET;

addr\_s.sin\_addr.s\_addr=INADDR\_ANY;

addr\_s.sin\_port=htons(MYPORT);

bzero(&(addr\_s.sin\_zero),8);

if(bind(server\_sockfd,(struct sockaddr\*)&addr\_s,sizeof(addr\_s))==-1)

{

printf("Bind error.\n");

exit(1);

}

if(listen(server\_sockfd,BACKLOG)==-1)

{

printf("Listen error.\n");

exit(1);

}

printf("Server:waiting.\n");

if((client\_sockfd=accept(server\_sockfd,(struct sockaddr\*)&addr\_c,&client\_len))==-1)

{

printf("Accept error.\n");

exit(1);

}

while(1)

{

if((len=recv(client\_sockfd,buf,sizeof(buf),0))==-1)

{

printf("Receive error.\n");

exit(1);

}

buf[len]='\0';

printf("<client> %s\n",buf);

printf("Server>");

gets(buf);

printf("\n");

if((result=send(client\_sockfd,buf,sizeof(buf),0))==-1)

{

printf("Sending error.");

exit(1);

}

}

close(client\_sockfd);

close(server\_sockfd);

return 0;

}

//chat client

#include<sys/types.h>

#include<sys/socket.h>

#include<sys/stat.h>

#include<stdio.h>

#include<unistd.h>

#include<sys/un.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<netdb.h>

#define MYPORT 5009

#define MAXBUF 100

int main(int argc,char \*argv[])

{

int sockfd,result,len,cmp;

char buf[MAXBUF];

struct sockaddr\_in address;

char message[100];

struct hostent \*he;

if((he=gethostbyname(argv[1]))==NULL)

{

printf("Error.\n");

exit(1);

}

if((sockfd=socket(AF\_INET,SOCK\_STREAM,0))==-1)

{

printf("Socket creation error.\n");

exit(1);

}

address.sin\_family=AF\_INET;

address.sin\_addr=\*((struct in\_addr\*)he->h\_addr);

address.sin\_port=htons(MYPORT);

bzero(&(address.sin\_zero),8);

printf("Client IP:%s\n",inet\_ntoa(address.sin\_addr));

result=connect(sockfd,(struct sockaddr\*)&address,sizeof(address));

if(result==-1)

{

printf("Connection error.\n");

exit(1);

}

printf("\n");

do

{

printf("<client>");

gets(buf);

printf("\n");

cmp=strcmp(buf,"quit");

if(cmp!=0)

{

if((result=send(sockfd,buf,sizeof(buf),0))==-1)

{

printf("Send error.\n");

exit(1);

}

if((len=recv(sockfd,buf,sizeof(buf),0))==-1)

{

printf("Receive error.\n");

exit(1);

}

printf("<server>");

buf[len]='\0';

printf("%s",buf);

printf("\n");

}

else

{

strcpy(buf,"Client disconnected");

send(sockfd,buf,sizeof(buf),0);

exit(1);

}

}while(strcmp(buf,"quit"));

close(sockfd);

exit(0);

}

**19.TCP WRAPPER**

HOST.ALLOW

This file contains access rules which are used to allow or deny connections to network services that either use the tcp\_wrappers library or that have been started through a tcp\_wrappers-enabled xinetd.

HOST.DENY

This file contains access rules which are used to deny connections to network services that either use the tcp\_wrappers library or those have been started through a tcp\_wrappers-enabled xinetd. The rules in this file can also be set up in /etc/hosts.allow with a 'deny' option instead.

**20. FILE TRANSFER USING SOCKET**

#### ALGORITHM

#### Server side

Step 1: start

Step 2: input header files

Step3: declare variables

Step 4: declare variables of sock addr\_in for storing server and client address

Step 5: initialize the server socket address with family, port number and IP

address

Step 6: create the socket

Step 7: bind the socket with socket address

Step 8: repeat through step 12 as long as there is something to communicate

Step 9: receive the file from the client

Step 10: open and read the requested file.

Step11: send file to the client

Step12: close the sockets

Step13: stop

#### Client side

Step 1: start

Step 2: include necessary header files

Step3: declare variables

Step 4: declare sock addr\_in variables for storing server address

Step 5: initialize the server socket address with family, port number and IP

address of the server

Step 6: create the socket

Step 7: read and send the filename to the server

Step 10: receive file content from the server

Step 11: display data on the monitor

Step 12: Stop

// ftp server

#include <sys/types.h>

#include <sys/socket.h>

#include <sys/stat.h>

#include <netinet/in.h>

#include <netdb.h>

#include <stdio.h>

#include <fcntl.h>

#define MYPORT 5006

#define BACKLOG 3

#define MAXBUF 100

int main(void)

{

int server\_sockfd,client\_sockfd;

int client\_len,len,il,ll,fd,cou,bs,ls,length;

struct sockaddr\_in addr\_s,addr\_c;

char buf[MAXBUF],input[20],a[20],c;

server\_sockfd=socket(AF\_INET,SOCK\_STREAM,0);

if(server\_sockfd==-1)

{

printf("socket creation error");

exit(1);

}

addr\_s.sin\_family = AF\_INET;

addr\_s.sin\_addr.s\_addr = INADDR\_ANY;

addr\_s.sin\_port = htons(MYPORT);

bzero(&(addr\_s.sin\_zero),8);

bs=bind(server\_sockfd, (struct sockaddr\*)&addr\_s, sizeof(addr\_s));

if(bs==-1)

{

printf("Binding error");

exit(1);

}

ls=listen(server\_sockfd,BACKLOG);

if(ls==-1)

{

printf("Listen error");

exit(1);

}

while(1)

{

printf("\nServer : Waiting.....\n");

client\_len=sizeof(struct sockaddr);

client\_sockfd=accept(server\_sockfd, (struct sockaddr\*)&addr\_c, &client\_len);

if(client\_sockfd==-1)

{

printf("Accept error\n");

exit(1);

}

len=recv(client\_sockfd, buf, sizeof(buf), 0);

if(len==-1)

{

printf("Receive error\n");

exit(1);

}

buf[len++]='\0';

printf("File name requested from client : %s\n",buf);

printf("\nServer has successfully received file");

strcpy(input,buf);

fd=open(input,O\_RDONLY);

cou=read(fd,buf,1000);

//printf("\nbuf = %s",buf);

send(client\_sockfd,buf,cou,0);

printf("\nServer send the content back to the client");

close(client\_sockfd);

}

close(server\_sockfd);

close(fd);

}

// ftp Client

#include <sys/types.h>

#include <sys/socket.h>

#include <stdio.h>

#include <unistd.h>

#include <sys/un.h>

#include <netinet/in.h>

#include <netdb.h>

#define MYPORT 5006

#define MAXBUF 100

int main(int argc,char \*argv[])

{

int sockfd,result,len,re,il=0,ll;

char c,buf[MAXBUF],a[20];

struct sockaddr\_in address;

struct hostent \*he;

if((he = gethostbyname(argv[1])) == NULL)

{

printf("Error..\n");

exit(1);

}

if((sockfd=socket(AF\_INET, SOCK\_STREAM, 0)) == -1)

{

printf("Socket creation error\n");

exit(1);

}

address.sin\_family = AF\_INET;

address.sin\_addr = \*((struct in\_addr\*)he->h\_addr);

address.sin\_port = htons(MYPORT);

bzero(&(address.sin\_zero),8);

printf("Client IP : %s\n",inet\_ntoa(address.sin\_addr));

result = connect(sockfd, (struct sockaddr\*)&address, sizeof(address));

if(result == -1)

{

printf("Connect error.\n");

exit(1);

}

printf("Enter file name : ");

scanf("%s",a);

while(a[il]!='\0')

{

il++;

}

a[il]='\0';

strcpy(buf,a);

result=send(sockfd, buf, il, 0);

if(result == -1)

{

printf("Send error\n");

exit(1);

}

else

ll=0;

printf("\nFile content is below : \n");

while(1)

{

ll=0;

ll=recv(sockfd, buf, sizeof(buf), 0);

if(ll==0)

{

exit(1);

}

buf[ll++]='\0';

printf("%s",buf);

}

close(sockfd);

}

**OUTPUT**

Server : Waiting.....

File name requested from client : test

Server has successfully received file

Server send the content back to the client

Server : Waiting.....

Client IP : 127.0.0.1

Enter file name : test

File content is below :

hello how r u

what is ur name?

**ADVANCED MICROPROCESSORS AND MICROCONTROLLERS**

**PROGRAM : 1**

**AIM:**Program for interchanging the values of two memory locations.

DATA SEGMENT

VALUE1 DB 0Ah

VALUE DB 14h

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

MOV AX,DATA

MOV DS, AX

MOV AL, VALUE1

XCHG VALUE2 , AL

MOV VALUE1, AL

INT 21h

CODE ENDS

END

**PROGRAM :2**

**AIM:**find the average of two values stored in memory locations named FIRST and SECOND and puts the results in the memory location AVGE.

DATA SEGMENT

FIRST DB 90h

SECOND DB 78h

AVGE DB ?

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS :DATA

START: MOV AX,DATA

MOV DS,AX

MOV AL,FIRST

ADD AL,SECOND

MOV AH,00h

ADC AH,00h

MOV BL,02h

DIV BL

MOV AVGE ,AL

CODE ENDS

END START

**PROGRAM :3**

**AIM:**convert the ASCII code to its BCD equivalent.

CODE SEGMENT

ASSUME CS:CODE

START: MOV BL,’5’

MOV AL, ‘9’

AND BL,0Fh

AND AL,0Fh

MOV CL,04h

ROL BL,CL

OR AL,BL

CODE ENDS

END START

**PROGRAM :4**

**AIM:**add a byte number from one memory location to a byte from the next memory location and put the sum in the third memory location .also ,save the carry flag in the least significant bit of the fourth memory location.

DATA SEGMENT

NUM1 DB 25h

NUM2 DB 80h

RESULT DB ?

CARRY DB

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START: MOV AX,DATA

MOV DS,AX

MOV AL,NUM1

ADD AL,NUM2

MOV RESULT,AL

RCL AL,01

AND AL,00000001B

MOV CARRY,AL

MOV AH,4CH

INT 21H

CODE ENDS

END START

**PROGRAM :5**

**AIM:**Assume a constant inflation factor that is added to a series of prices stored in the memory. Theprograme copies the new price over the old price. It is assumed that price data is available in BCD forms.

ARRAYS SEGMENT

PRICE DB 36h,55h,27h,42h,38h,41h,29h,39h

ARRAYS ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:ARRAYS

START: MOV AX,ARRAYS

MOV DS,AX

LEA BX,PRICES

MOV CX,0008h

DO\_NEXT: MOV AL,[BX]

ADD AL,0Ah

DAA

MOV [BX],AL

INC BX

DEC CX

JNZ DO\_NEXT

MOV AH,4CH

INT 21H

CODE ENDS

END START

**PROGRAM :6**

**AIM:**program for printing alphabets(A-Z).

CODE SEGMENT

ASSUME:CS:CODE

MAINP: MOV CX,1AH

MOV DL,41H

NEXTC: MOV AH,02H

INT 21H

INC DL

LOOP NEXTC

MOV AX,4C00H

INT 21H

CODE ENDS

END MAINP

**PROGRAM :7**

**AIM:**Program to compare a pair of character entered through keyboard.

DATA SEGMENT

XX DB ?

YY DB ?

DATA ENDS

CODE SEGMENT

ASSUME: CS:CODE,DS:DATA

MAINP: MOV AX,DATA

MOV DS,AX

MOV CX,03H

NEXTP: MOV AH,01H

INT 21H

MOV XX,AL

MOV AH,01H

INT 21H

MOV YY,AL

MOV BH,XX

MOV BL,YY

CMP BH,BL

JNE NOT\_EQUAL

EQUAL: MOV AH,02H

MOV DL,’Y’

INT 21H

JMP CONTINUE

NOT\_EQUAL:MOV AH,02H

MOV DL,’N’

INT 21H

CONTINUE: LOOP NEXT P

MOV AH,4C H

INT 21 H

CODE ENDS

END MAINP

**PROGRAM :8**

**AIM:**Program to find the largest and tha smallest array values.

DATA SEGMENT

ARRAY DW -1,2000,-4000,32767,500,0

LARGE DW ?

SMALL DW ?

DATA ENDS

END

CODE SEGMENT

MOV AX,DATA

MOV DS,AX

MOV DI,OFFSET ARRAY

MOV AX,[DI]

MOV DX,AX

MOV BX,AX

MOV CX,6

A1: MOV AX,[DI]

CMP AX,BX

JGE A2

MOV BX,AX

JMP A3

A2: CMP AX,DX

JLE A3

MOV DX,AX

A3: ADD D1,2

LOOP A1

MOV LARGE,DX

MOV SMALL,BX

MOV AX,4C00h

INT 21h

CODE ENDS

**PROGRAM :9**

**AIM**:Matching two strings of same length stored in memory locations.

DATA SEGMENT

PASSWORD DB ‘FAILSAFE’

DESTSTR DB ‘FEELSAFE’

MESSAGE DB ‘Strings are equal $’

DATA ENDS

CODE SEGMENT

ASSUME:CS:CODE,DS:DATA,ES:DATA

MOV AX,DATA

MOV DS,AX

MOV ES,AX

LEA SI,PASSWORD

LEA DI,DESTSTR

MOV CX,08

CLD

REPE CMPSB

JNE NOTEQUAL

MOV AH,09

MOV DX,OFFSET MESSAGE

INT 21h

NOEQUAL: MOV AX,4C00h

INT 21h

CODE ENDS

END

**PROGRAM :10**

**AIM**:Program to add two five-byte numbers using array.

DATA SEGMENT

NUM1 DB 0FFh, 10h, 01h, 11h, 20h

NUM2 DB 10h, 20h, 30h, 40h, 0FFh

SUM DB 6DUP(0)

DATA ENDS

LEN EQU 05h

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START: MOV AX,DATA

MOV DS,AX

MOV SI,00

MOV CX,0000

MOV CL,LEN

CLC

AGAIN: MOV AL,NUM1[SI]

ADC AL,NUM2[SI]

MOV SUM[SI],AL

INC SI

LOOP AGAIN

RCL AL,01h

AND AL,01h

MOV SUM[SI],AL

FINISH: MOV AX,4C00h

INT 21h

CODE ENDS END START