**SYSTEM IMPLEMENTATION**

import java.sql.\*;

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class dataserver

{

private dbase db;

ResultSet ls;

Connection cs;

Statement st;

Vector v,v1,allfile;

static ServerSocket socte;

static Socket soc;

ObjectInputStream dis;

ObjectOutputStream dos;

InputStream is;

OutputStream os;

String n="",ip="",s,name,nodes;

static dataserver sol;

int polte=4003,na;

static String ipa="";

static int poct =0;

public dataserver()throws Exception

{

super();

db = new dbase();

st=db.dbcon();

s="delete from userdetails";

db.delete(s);

}

public void listen()throws Exception

{

dis=new ObjectInputStream(soc.getInputStream());

String lname=(String)dis.readObject();

if(lname.equals("user"))

{

String output="";

dis=new ObjectInputStream(soc.getInputStream());

String nname2=(String)dis.readObject();

String ay[]=nname2.split("&");

int l=Integer.parseInt(ay[1]);

s="select \* from userdetails where username='"+ay[0]+"' and ip='"+ay[2]+"'";

if(db.check(s))

{

output="exist";

s="delete from userdetails where username=''";

db.delete(s);

}

else

{

s="update userdetails set username='"+ay[0]+"' where portno='"+l+"'";

db.insert(s);

s="update userdetails set password='"+ay[3]+"' where portno='"+l+"'";

db.insert(s);

}

dos=new ObjectOutputStream(soc.getOutputStream());

dos.writeObject(output);

}

else if(lname.equals("validuser"))

{

String oup1e="",status1="";

dis=new ObjectInputStream(soc.getInputStream());

String nname12=(String)dis.readObject();

String ay1[]=nname12.split("&");

int l1=Integer.parseInt(ay1[1]);

s="select status from userdetails where username='"+ay1[0]+"'";

ls=st.executeQuery(s);

if(ls.next())

{

status1=ls.getString(1);

}

if(status1.equals("disable"))

{

s="select password from userdetails where ip='"+ay1[2]+"' and username='"+ay1[0]+"'";

String pwo=db.select1(s);

s="select username from userdetails where ip='"+ay1[2]+"' and username='"+ay1[0]+"'";

String usm=db.select1(s);

if(pwo.equals(ay1[3]))

{

if(usm.equals(ay1[0]))

{

oup1e="ok";

s="update userdetails set portno='"+l1+"' where username='"+ay1[0]+"'";

db.insert(s);

s="update userdetails set status='enable' where username='"+ay1[0]+"'";

db.insert(s);

}

else

{

oup1e="ok1";

}

}

else

{

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

oup1e="ok1";

}

}

else

{

oup1e="ok1";

}

dos=new ObjectOutputStream(soc.getOutputStream());

dos.writeObject(oup1e);

}

else

{

String nameay[]=lname.split("&");

dos=new ObjectOutputStream(soc.getOutputStream());

polte=polte+1;

String p=String.valueOf(polte);

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

s="Insert into userdetails values('','','"+nameay[0]+"','"+polte+"','disable','"+nameay[1]+"','','')";

db.insert(s);

dos.writeObject(p);

System.out.println("send");

}

}

public static void main(String[] alg)

{

try

{

socte=new ServerSocket(4000);

sol=new dataserver();

while(true)

{

soc=socte.accept();

sol.listen();

}

}

catch (Exception ex)

{

System.out.println("Failed loading L&F: ");

System.out.println(ex);

}

}

}

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import javax.swing.event.\*;

import java.io.\*;

import java.net.\*;

import java.util.\*;

import java.sql.\*;

import java.lang.\*;

import java.util.Vector;

import java.awt.Toolkit;

import java.util.Timer;

import java.util.TimerTask;

import java.util.StringTokenizer;

import java.util.concurrent.\*;

public class server extends JFrame implements Runnable

{

int change;

Vector system=new Vector();

int ucl,thtime,losme;

int sto,ttime=0;

String mesg,plocess,file,sysname,s,jpot,nameu="",tcnj,dl;

public ResultSet ns,ls,csl;

public Connection con,col,co;

public Statement st,sl,sc;

DefaultListModel model1;

private static JLabel jLabel1;

private static JLabel jLabel2;

private static JLabel jLabel3;

private static JLabel jLabel4;

private static JTextArea jList1;

private static JScrollPane jscle;

private static JTextArea jTextAea1;

private static JScrollPane jscne;

private JPanel contentPane;

String logsys,log,sname,all,dosn,smesg,toype="";

static Thread t1;

String socue,doste,state,smsg,slcne,msgtime,ht,mcj,ncj,ipj,pcj,clj,flj,tlj;

FileOutputStream output,op1;

String fd;

boolean sta=true,stak;

int end;

public static Socket cs,cs1,cs2;

ObjectInputStream in,in1,in2;

ObjectOutputStream ois1,ois2,ois3;

public static server sn,c,f;

String los1;

int staa=0,che,flage=0;

// String ht;

Vector v1,v2;

int pocne,tyl=0;

Vector p2=new Vector();

Vector tcxoe=new Vector();

Vector alfie=new Vector();

public static ServerSocket mst;

public server()throws Exception

{

super();

System.out.println(">");

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

con = DriverManager.getConnection("jdbc:odbc:server");

st=con.createStatement();

col = DriverManager.getConnection("jdbc:odbc:server");

sl=col.createStatement();

co = DriverManager.getConnection("jdbc:odbc:server");

sc=co.createStatement();

s="delete from filetable";

st.executeUpdate(s);

s="update list set status='0'";

st.executeUpdate(s);

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

}

public void intz()throws Exception

{

System.out.println("[");

JLabel bg;

bg = new JLabel(new ImageIcon("predictionserver.jpg"));

pocne=6666;

v1=new Vector();

JLabel jLabel7=new JLabel();

jLabel1 = new JLabel();

jLabel2 = new JLabel();

jLabel3=new JLabel();

jLabel4 = new JLabel();

model1=new DefaultListModel();

jList1 = new JTextArea();

jscle = new JScrollPane();

jTextAea1 = new JTextArea();

jscne = new JScrollPane();

contentPane = (JPanel)this.getContentPane();

setResizable(false);

jLabel1.setText("Incoming");

jLabel2.setText("Log");

jscle.setViewportView(jList1);

jscne.setViewportView(jTextAea1);

String lew="";

contentPane.setLayout(null);

addComponent(contentPane, jLabel7, 50,20,120,120);

addComponent(contentPane, jLabel1, 15,25,150,15);

addComponent(contentPane, jLabel2, 15,230,150,15);

addComponent(contentPane, jscle, 15,50,525,177);

addComponent(contentPane, jscne, 15,250,525,177);

addComponent(contentPane, bg, 50,20,120,120);

this.setTitle("Cloud : Server ...");

this.setLocation(new Point(100, 100));

this.setSize(new Dimension(560, 500));

this.setVisible(true);

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

}

private void addComponent(Container containe,Component c,int x,int y,int width,int height)

{

c.setBounds(x,y,width,height);

containe.add(c);

}

public void filelist()throws Exception

{

alfie.removeAllElements();

File lco = new File(".","//files" );

String[] filesl = lco.list();

System.out.println( "Files in this directory are:" );

for ( String filel : filesl )

{

alfie.add(filel);

}

System.out.println(" File List Receive from server 1 \n\n\n-----\n\n"+alfie+"\n\n--------\n\n");

System.out.println( "\nFiles in this directory are:" );

System.out.println( alfie );

}

public void thszc(int flagc)throws Exception

{

flage=flagc;

t1=new Thread(this);

t1.start();

}

int tim=0;

public void cal()throws Exception

{

while(1==1)

{

if(tim==60)

{

File flx=new File(".","/cache/");

System.out.println("FLX");

String flxjem[]=flx.list();

System.out.println("flx");

int flxjew=flxjem.length;

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

int x=0;

while(x<flxjew)

{

flx=new File(".","/cache/"+flxjem[x]);

flx.delete();

x=x+1;

}

tim=0;

}

else

{

tim=tim+1;

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

Thread.sleep(1000);

}

}

}

public void msg()throws Exception

{

String fileh="",ipl="",pf="";

mst=new ServerSocket(5000);

while(true)

{

fileh="";

ipl="";

pf="";

try

{

System.out.println("listening");

cs=mst.accept();

in=new ObjectInputStream(cs.getInputStream());

String losue=(String)in.readObject();

//System.out.println(" First Client Request ::::::::"+lesue);

StringTokenizer tokens = new StringTokenizer(losue,"$");

while(tokens.hasMoreTokens())

{

//System.out.println(tokens.nextToken());

plocess=tokens.nextToken();

file=tokens.nextToken();

sysname=tokens.nextToken();

jpot=tokens.nextToken();

nameu=tokens.nextToken();

}

String filehl[]=file.split("&");

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

if(plocess.equals("select"))

{

System.out.println(" select ");

filelist();

change=0;

p2.removeAllElements();

tcxoe.removeAllElements();

for(int count=0;count<alfie.size();count++)

{

String che=(String)alfie.elementAt(count);

p2.add(che);

if(che.endsWith(file))

{

tcxoe.add(che);

}

if(file.equals("(\*.\*"))

{

if(!che.endsWith("\*.txt") &&!che.endsWith("\*.jpg") && !che.endsWith("\*.gif") && !che.endsWith("\*.pdf") )

{

tcxoe.add(che);

}

}

}

ois1=new ObjectOutputStream(cs.getOutputStream());

if(change==0)

{

ois1.writeObject(tcxoe);

}

else

{

ois1.writeObject(p2);

}

}

else if(plocess.equals("list"))

{

System.out.println("finish:"+file);

st.executeUpdate("update list set status='0' where port='"+file+"'");

if(nameu.equals("u"))

{

dl="./cache/"+sysname;

FileInputStream input= new FileInputStream(dl);

byte[] blt=new byte[input.available()];

input.read(blt);

output=new FileOutputStream("./files/"+sysname);

System.out.println("upload" );

output.write(blt);

//input.delete();

}

int count=0;

//if(1==1)

//{

//}

//if(nameu<count)

// {

// }

//else

//{

//}

}

else if(plocess.equals("chk"))

{

System.out.println("chk nameu:"+file);

// st.executeUpdate("update list set status='0' where port='"+file+"'");

ipj="0";

pcj="0";

clj="select \* from list where status='0'";

ns=sc.executeQuery(clj);

if(ns.next())

{

flj=ns.getString(1);

ipj=ns.getString(2);

pcj=ns.getString(3);

ncj=ns.getString(4);

}

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

clj="select \* from list where lname=(select max(lname) from list where status='1')";

ns=sl.executeQuery(clj);

if(ns.next())

{

tlj=ns.getString(1);

mcj=ns.getString(3);

}

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

if(file.equals(mcj))

{

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

if(Integer.parseInt(pcj)>=Integer.parseInt(mcj))

{

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

//ois1=new ObjectOutputStream(cs.getOutputStream());

//ois1.writeObject(ipj);

ipj="0";

pcj="0";

}

else

{

}

}

else

{

ipj="0";

pcj="0";

}

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

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

ois1=new ObjectOutputStream(cs.getOutputStream());

ois1.writeObject(ipj);

ois1.writeObject(pcj);

}

else if(plocess.equals("upload"))

{

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

byte[] dfile=(byte[])in.readObject();

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

tyl=tyl+1;

System.out.println("ty1:"+ tyl);

toype=String.valueOf(tyl);

System.out.println("type:"+ toype);

st.executeUpdate("insert into filetable values('"+jpot+"','"+filehl[0]+"','"+sysname+"','0','u')");

System.out.println("table");

String df="./cache/"+filehl[0];

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

FileOutputStream out=new FileOutputStream(df);

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

out.write(dfile);

System.out.println(" file upladed now ");

}

else if(plocess.equals("filecontent"))

{

FileInputStream input= new FileInputStream(file);

byte[] b=new byte[input.available()];

input.read(b);

ois1=new ObjectOutputStream(cs.getOutputStream());

ois1.writeObject(b);

File fij=new File(".","/cache/"+sysname);

FileOutputStream out=new FileOutputStream(fij);

out.write(b);

}

else

{

System.out.println("download:"+filehl.length);

String nnn="";

for(int i=0;i<filehl.length;i++)

{

jList1.append("file: "+filehl[i]+ " download request \n");

String filme=filehl[i];

tyl=tyl+1;

toype=String.valueOf(tyl);

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

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

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

System.out.println("jport:"+jpot);

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

File fd5=new File(".","/cache/"+filehl[i]);

boolean bo7=fd5.exists();

System.out.println("bo7:"+bo7);

if(bo7==false)

{

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

st.executeUpdate("insert into filetable values('"+jpot+"','"+filehl[i]+"','"+sysname+"','0','d')");

System.out.println("s");

}

else

{

jList1.append("file: "+filehl[i]+ " availale in cache \n");

File fic=new File(".","/cache/"+filehl[i]);

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

FileInputStream fio= new FileInputStream(fic);

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

byte[] b=new byte[fio.available()];

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

fio.read(b);

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

Socket slc=new Socket(sysname,Integer.parseInt(jpot));

ois1=new ObjectOutputStream(slc.getOutputStream());

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

ois1.writeObject(b);

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

ois1.writeObject(filehl[i]);

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

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

}

}

}

}

catch (Exception el1)

{

}

}

}

public void run()

{

if(flage==1)

{

try

{

msg();

}

catch(Exception jl)

{

}

}

else if(flage==9)

{

while(true)

{

try

{

String fileh="",ipc="", pcl ="",sys="",lcp="",lname="",pcname="",c="";

c="select \* from list where status='1'";

csl=sc.executeQuery(c);

int flagepname=1;

if(csl.next())

{

pcname=csl.getString(1);

flagepname=0;

}

if(flagepname==0)

{

c="select \* from list where status='0' and pname='"+pcname+"'";

csl=sc.executeQuery(c);

if(csl.next())

{

pcname=csl.getString(1);

ipc=csl.getString(2);

pcl=csl.getString(3);

lname=csl.getString(4);

}

else

{

c="select \* from list where status='0'";

csl=sc.executeQuery(c);

if(csl.next())

{

pcname=csl.getString(1);

ipc=csl.getString(2);

pcl=csl.getString(3);

lname=csl.getString(4);

}

}

}

else

{

c="select \* from list where status='0'";

csl=sc.executeQuery(c);

if(csl.next())

{

pcname=csl.getString(1);

ipc=csl.getString(2);

pcl=csl.getString(3);

lname=csl.getString(4);

}

}

c="select \* from filetable where type='0'";

csl=sc.executeQuery(c);

if(csl.next())

{

lcp=csl.getString(1);

fileh=csl.getString(2);

sys=csl.getString(3);

tcnj=csl.getString(5);

if(tcnj.equals("d"))

{

ht=".//files/"+fileh;

}

else

{

ht="u";

}

Socket cs1=new Socket(ipc,Integer.parseInt(pcl));

ObjectOutputStream oos=new ObjectOutputStream(cs1.getOutputStream());

oos.writeObject(ht);

oos.writeObject(sys);

oos.writeObject(lcp);

oos.writeObject(fileh);

oos.writeObject("0");

System.out.println(" send to cloudserver"+ht);

jTextAea1.append("Physical Machine : "+pcname+"\n"+lname+" Turn On \n File : "+fileh+"assigned");

sc.executeUpdate("update filetable set type='1' where filename='"+fileh+"' and pnz='"+lcp+"'");

System.out.println(" pc1;;;;;;;;"+pcl);

sc.executeUpdate("update list set status='1' where lname='"+lname+"' and port='"+pcl+"'");

System.out.println(" pc1-------------");

System.out.println(" pc1.............");

System.out.println(" pc10000000000000"+pcl);

}

}

catch(Exception el)

{

}

}

}

else if(flage==3)

{

try

{

cal();

}

catch(Exception jl)

{

}

}

}

public static void main(String[] alg) throws Exception

{

try

{

sn=new server();

sn.intz();

sn.thszc(1);

c=new server();

c.thszc(9);

f=new server();

f.thszc(3);

}

catch (Exception ex)

{

}

}

}

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import javax.swing.event.\*;

import java.io.\*;

import java.net.\*;

import java.util.\*;

import java.sql.\*;

import javax.sql.\*;

import java.lang.\*;

import java.util.Vector;

import java.awt.Toolkit;

import java.util.Timer;

import java.util.TimerTask;

import java.util.StringTokenizer;

import javax.naming.\*;

import javax.naming.directory.\*;

import java.util.Calendar;

class giz extends JFrame implements rouconfig

{

public long delay,time;

public static int i=0,st1,counte;

public static int a=9001,b=9003,c=9006;

public String mesg,act="active",ss1;

public ServerSocket ss;

public ServerSocket ssl;

// public ServerSocket ssz=new ServerSocket(2225);

public Connection con;

public Statement st;

public DefaultListModel model1;

public JLabel jLabel1;

public JLabel jLabel2;

public JLabel jLabel3;

public JLabel jLabel4;

public static JTextArea jList1;

public static JScrollPane jscle;

public static JTextArea jctxe,jta;

public static JScrollPane jscne;

public static JScrollPane jscoe;

public JButton jButton1;

public JButton jButton2;

public JButton jButton3;

public JButton jButton4;

public JButton jButton5;

public JButton jButton6;

public JPanel contentPane;

public Vector data=new Vector();

public ArrayList a1=new ArrayList();

public ArrayList a2=new ArrayList();

public ArrayList allname=new ArrayList();

public String logsys,log,sname,alc,dosne,smoje,ipj,pcj;

public static Thread t1,t2,t3;

public String socue,doste,state,smsg,slc,msgtime;

public FileOutputStream output,op1;

public String fd;

public boolean sta=true,stak;

public int end;

// static giz se;

//static systeml sl;

public Socket cs,cs1,cs2,csj,csn,cso,csl;

public ObjectInputStream in,in1,in2,inl,inn;

public ObjectOutputStream ois1e,oisne,oisle;

public String les;

public static int staa=0,count=3000;

public String ht;

public Vector v1,v2;

public String name[]=new String[100];

public String allsta;

public int potno;

public boolean sss;

public Vector pic=new Vector();

public Vector text=new Vector();

public Vector allfile=new Vector();

public Vector allfilel=new Vector();

public giz()throws Exception

{

}

public void gic()throws Exception

{

JLabel bg;

bg = new JLabel();

v1=new Vector();

setResizable(false);

allsta="allow";

jLabel1 = new JLabel("VM1"); //new ImageIcon("bg2.gif")

jLabel2 = new JLabel("VM2");

jLabel3 = new JLabel("VM3");

jLabel4 = new JLabel();

model1=new DefaultListModel();

jList1 = new JTextArea();

jscle = new JScrollPane();

jctxe = new JTextArea();

jscne = new JScrollPane();

jButton1 = new JButton();

jButton2 = new JButton();

jButton3 = new JButton();

jButton4 = new JButton();

jButton5 = new JButton();

jButton6 = new JButton();

contentPane = (JPanel)this.getContentPane();

jscoe = new JScrollPane();

jta = new JTextArea();

/\*t1=new Thread(this);

t2=new Thread(this);

t3=new Thread(this);\*/

JLabel jLabel7=new JLabel(new ImageIcon("server.jpg"));

jscle.setViewportView(jList1);

jscne.setViewportView(jctxe);

jscoe.setViewportView(jta);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

});

String lew="";

contentPane.setLayout(null);

addComponent(contentPane,jLabel1,25,25,150,25);

addComponent(contentPane, jscle, 5,50,180,350);

addComponent(contentPane, jLabel7, 330,50,120,120);

addComponent(contentPane,jLabel2,187,25,150,25);

addComponent(contentPane, jscne, 187,50,180,350);

addComponent(contentPane,jLabel3,369,25,150,25);

addComponent(contentPane, jscoe,369,50,180,350);

jscle.setVisible(false);

jscne.setVisible(false);

jscoe.setVisible(false);

jButton2.setEnabled(false);

jButton5.setEnabled(false);

this.setTitle("PHYSICAL SYSTEM 1 ");

this.setLocation(new Point(100, 100));

this.setSize(new Dimension(570,500));

//this.setSize(new Dimension(540, 490));

this.setVisible(true);

} // Constructor class closed ..........

private void addComponent(Container containe,Component c,int x,int y,int width,int height)

{

c.setBounds(x,y,width,height);

containe.add(c);

}

private void jButton1\_actionPerformed(ActionEvent e)

{

}

private void jButton2\_actionPerformed(ActionEvent e)

{

//System.out.println("\njButton2\_actionPerformed(ActionEvent e) called.");

// TODO: Add any handling code here

}

private void jButton3\_actionPerformed(ActionEvent e)

{

//System.out.println("\njButton3\_actionPerformed(ActionEvent e) called.");

// TODO: Add any handling code here

}

private void jButton4\_actionPerformed(ActionEvent e)

{

//System.out.println("\njButton4\_actionPerformed(ActionEvent e) called.");

// TODO: Add any handling code here

}

/\*public void msg()

{

// ss=new ServerSocket(potno);

try

{

ss=new ServerSocket(2222);

while(true)

{

System.out.println(" server is listening ...");

cs=ss.accept();

in=new ObjectInputStream(cs.getInputStream());

String resu=(String)in.readObject();

System.out.println(" receive req from server "+resu);

if(resu.equals("hai32"))

{

ObjectOutputStream oos=new ObjectOutputStream(cs.getOutputStream());

oos.writeObject("hai");

}

else if(resu.equals("fileList"))

{

allfile.removeAllElements();

File dir = new File(".","//files" );

String[] files = dir.list();

System.out.println( "Files in this directory are:" );

for ( String file : files )

{

allfile.add(file);

}

ObjectOutputStream oos=new ObjectOutputStream(cs.getOutputStream());

oos.writeObject(allfile);

}

else

{

String ip=(String)in.readObject();

String p=(String)in.readObject();

String fileh=(String)in.readObject();

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

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

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

jList1.append("file: "+resu+ " download process started \n");

Socket csj=new Socket(ip,Integer.parseInt(p));

ObjectOutputStream ooz=new ObjectOutputStream(csj.getOutputStream());

FileInputStream input = new FileInputStream(resu);

//FileInputStream input = new FileInputStream("bbb.txt");

byte[] b=new byte[input.available()];

input.read(b);

ooz.writeObject(b);

ooz.writeObject(fileh);

System.out.println("client");

jList1.append("file download completed \n");

}

}

}

catch (Exception e4)

{

//System.out.println(e4);

}

}\*/

}

class systeml extends giz implements Runnable

{

Thread th;

int che;

public systeml(int check)throws Exception

{

che=check;

th=new Thread(this);

th.start();

}

public void run()

{

// ss=new ServerSocket(potno);

try

{

ssl=new ServerSocket(che);

while(true)

{

System.out.println(che+" server is listening ...");

csl=ssl.accept();

if(che==a)

jscle.setVisible(true);

else if(che==b)

jscne.setVisible(true);

else if(che==c)

jscoe.setVisible(true);

inl=new ObjectInputStream(csl.getInputStream());

String losue=(String)inl.readObject();

System.out.println(" receive req from server "+losue);

if(losue.equals("hai32"))

{

ObjectOutputStream lll=new ObjectOutputStream(csl.getOutputStream());

lll.writeObject("hai");

}

else if(losue.equals("fileList"))

{

allfilel.removeAllElements();

File lc = new File(".","//files" );

String[] filesl = lc.list();

System.out.println( "Files in this directory are:" );

for ( String filel : filesl )

{

allfilel.add(filel);

}

ObjectOutputStream lll=new ObjectOutputStream(csl.getOutputStream());

lll.writeObject(allfilel);

}

else if(losue.equals("u"))

{

String ipl=(String)inl.readObject();

String pcl=(String)inl.readObject();

String filehl=(String)inl.readObject();

String lname=(String)inl.readObject();

losue=".//cache/"+filehl;

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

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

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

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

Socket jco=new Socket(sct,5000);

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

ObjectOutputStream ic=new ObjectOutputStream(jco.getOutputStream());

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

ic.writeObject("filecontent$"+losue+"$"+filehl+"$1$0");

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

ObjectInputStream jci=new ObjectInputStream(jco.getInputStream());

byte[] bf=(byte[])jci.readObject();

System.out.println("hai");

int tk=0;

int bi=bf.length;

int li=Integer.parseInt(lname);

int nj=0;

tk=li;

for(int k=0;k<bi;k++)

{

tk=tk+count;

System.out.println("byte"+tk);

if(tk>=bi)

{

nj=0;

if(che==a)

jList1.setText("file: "+filehl+ "\n upload process completed \n Byte : "+String.valueOf(bi));

else if(che==b)

jctxe.setText("file: "+filehl+ "\n upload process completed \n Byte : "+String.valueOf(bi));

else if(che==c)

jta.setText("file: "+filehl+ "\n upload process completed \n Byte : "+String.valueOf(bi));

break;

}

else

{

if(che==a)

jList1.setText("file: "+filehl+ "\n upload process started \n Byte : "+String.valueOf(tk));

else if(che==b)

jctxe.setText("file: "+filehl+ "\n upload process started \n Byte : "+String.valueOf(tk));

else if(che==c)

jta.setText("file: "+filehl+ "\n upload process started \n Byte : "+String.valueOf(tk));

Socket jil=new Socket(sct,5000);

ObjectOutputStream il=new ObjectOutputStream(jil.getOutputStream());

il.writeObject("chk$"+String.valueOf(che)+"$3$0$1");

System.out.println("client");

ObjectInputStream jli=new ObjectInputStream(jil.getInputStream());

ipj=(String)jli.readObject();

pcj=(String)jli.readObject();

if(!ipj.equals("0"))

{

Socket jin=new Socket(ipj,Integer.parseInt(pcj));

ObjectOutputStream in=new ObjectOutputStream(jin.getOutputStream());

in.writeObject(losue);

in.writeObject(ipl);

in.writeObject(pcl);

in.writeObject(filehl);

lname=String.valueOf(tk);

in.writeObject(lname);

nj=0;

System.out.println("changed");

break;

}

Thread.sleep(200);

}

// Socket csf=new Socket(ipl,Integer.parseInt(pcl));

//ObjectOutputStream lll=new ObjectOutputStream(csf.getOutputStream());

} //lll.writeObject(bf);

//lll.writeObject(filehl);

if(che==a)

jscle.setVisible(false);

else if(che==b)

jscne.setVisible(false);

else if(che==c)

jscoe.setVisible(false);

if(nj!= 0)

{

Socket jil=new Socket(sct,5000);

ObjectOutputStream il=new ObjectOutputStream(jil.getOutputStream());

il.writeObject("list$"+String.valueOf(che)+"$"+filehl+"$1$1$u");

System.out.println("client");

}

}

else

{

String ipl=(String)inl.readObject();

String pcl=(String)inl.readObject();

String filehl=(String)inl.readObject();

String lname=(String)inl.readObject();

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

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

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

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

Socket jco=new Socket(sct,5000);

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

ObjectOutputStream ic=new ObjectOutputStream(jco.getOutputStream());

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

ic.writeObject("filecontent$"+losue+"$"+filehl+"$1$0");

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

ObjectInputStream jci=new ObjectInputStream(jco.getInputStream());

byte[] bf=(byte[])jci.readObject();

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

int tk=0;

int bi=bf.length;

int li=Integer.parseInt(lname);

int mj=0;

tk=li;

for(int k=li;k<bi;k++)

{

tk=tk+count;

System.out.println("byte"+tk);

if(tk>=bi)

{

mj=0;

if(che==a)

jList1.setText("file: "+filehl+ "\n download process completed \n Byte : "+String.valueOf(bi));

else if(che==b)

jctxe.setText("file: "+filehl+ "\n download process completed \n Byte : "+String.valueOf(bi));

else if(che==c)

jta.setText("file: "+filehl+ "\n download process completed \n Byte : "+String.valueOf(bi));

break;

}

else

{

if(che==a)

jList1.setText("file: "+filehl+ "\ndownload process started \n Byte : "+String.valueOf(tk));

else if(che==b)

jctxe.setText("file: "+filehl+ "\ndownload process started \n Byte : "+String.valueOf(tk));

else if(che==c)

jta.setText("file: "+filehl+ "\ndownload process started \n Byte : "+String.valueOf(tk));

if((tk%10000)==0)

{

Socket jil=new Socket(sct,5000);

ObjectOutputStream il=new ObjectOutputStream(jil.getOutputStream());

il.writeObject("chk$"+String.valueOf(che)+"$3$0$1");

System.out.println("client");

ObjectInputStream jli=new ObjectInputStream(jil.getInputStream());

ipj=(String)jli.readObject();

pcj=(String)jli.readObject();

if(!ipj.equals("0"))

{

Socket jin=new Socket(ipj,Integer.parseInt(pcj));

ObjectOutputStream in=new ObjectOutputStream(jin.getOutputStream());

in.writeObject(losue);

in.writeObject(ipl);

in.writeObject(pcl);

in.writeObject(filehl);

lname=String.valueOf(tk);

in.writeObject(lname);

mj=1;

System.out.println("changed");

break;

}

}

Thread.sleep(200);

}

}

if(mj!=1)

{

Socket csf=new Socket(ipl,Integer.parseInt(pcl));

ObjectOutputStream lll=new ObjectOutputStream(csf.getOutputStream());

lll.writeObject(bf);

lll.writeObject(filehl);

}

if(che==a)

jscle.setVisible(false);

else if(che==b)

jscne.setVisible(false);

else if(che==c)

jscoe.setVisible(false);

if(mj!=1)

{

Socket jil=new Socket(sct,5000);

ObjectOutputStream il=new ObjectOutputStream(jil.getOutputStream());

il.writeObject("list$"+String.valueOf(che)+"$3$0$d");

System.out.println("client");

}

}

}

}

catch (Exception e9)

{

//System.out.println(e4);

}

}

}

class psystem1

{

public static void main(String[] alg) throws Exception

{

giz se=new giz();

se.gic();

systeml sl=new systeml(se.a);

systeml s=new systeml(se.b);

systeml sc=new systeml(se.c);

}

}