|  |  |
| --- | --- |
| Inline Text Wrapping Picture | Inline Text Wrapping Picture |

Java网络编程结题报告

学 号: 2016212024，2016211977，

2016211965，2016211961

姓 名: 张世杰，曾嘉，徐朗朗，程元

学 院: 软件学院

专业(领域): 软件工程

研究方向: Java网络编程

导师姓名: 朱旭振

北京邮电大学

2018年6月9日

|  |  |  |  |
| --- | --- | --- | --- |
| 论文题目 | 基于JAVA网络编程的聊天软件 | | |
| 论文类型 | 应用研究 | 选题来源 |  |
| 开题日期 | 2018-04-02 | 是否开题题目 | 是 |
| 论文开始日期 | 2018-06-09 | 报告日期 | 2018-06-12 |
| 报告地点 | 教三333 | 报告时间 | 晚上18:30-20:20 |
| **1、研究背景和内容**  **1.1、研究内容简介**  近几年来，即时通讯软件(InstantMessenger)的发展突飞猛进。在短短的几年内，网络即时通信大有取代传统通信方式之势，继电子邮件之后，即时通讯对我们的通信方式又进行了一场革命，这种革命甚至延续到了我们的感情领域，只要双方都在互联网上。即时通讯所拥有的实时性、跨平台性、成本低、效率高等诸多优势，使之日益受到网民们的喜爱。即时通讯(Instant messaging,简称IM)是一个终端服务，允许两人或多人使用网路即时的传递文字信息、档案、语音与视频交流。分电话即时通讯手机和网站即时通讯代表百问www.baiask.com，手机即时通讯代表是短信，视频即时通讯如QQ，MSN，百度hi等应用形式。  　　即时通讯是一个终端连往一个即时通讯网路的服务。即时通讯不同于e-mail在于它的交谈是即时的。大部分的即时通讯服务提供了presence awareness的特性──显示联络人名单，联络人是否在缐上与能否与联络人交谈。  　　在早期的即时通讯程序中，用户输入的每一个字符都会即时显示在双方的萤幕，且每一个字符的删除与修改都会即时的反应在屏幕上。这种模式比起使用e-mail更像是电话交谈。在现在的即时通讯程序中，交谈中的另一方通常只会在本地端按下送出键(Enter或是Ctrl+Enter)后才会看到讯息信息。  　　在网际网路上受欢迎的即时通讯服务包含了MSN Messenger、AOL Instant Messenger、Yahoo! Messenger、NET Messenger Service、Jabber、ICQ与QQ。 这些服务有赖于许多想法更久的(与普遍)的缐上聊天媒介，如Internet Relay Chat一样知名。  　　1970年代早期，一种更早的即时通讯形式是柏拉图系统(PLATO system)。之后在1980年代，UNIX/Linux的交谈即时信息被广泛的使用于工程师与学术界，1990年代即时通讯更跨越了网际网路交流。1996年11月，ICQ是首个广泛被非UNIX/Linux使用者用于网际网路的即时通讯软体。在ICQ的介绍之后，同时在许多地方有一定数量的即时通讯方式发展，且各式的即时通讯程序有独立的协定，无法彼此互通。这引导使用者同时执行两个以上的即时通讯软体，或者他们可以使用支持多协定的终端软体，如Gaim、Trillian或Jabber。  　　近年来，许多即时通讯服务开始提供视讯会议的功能，网络电话(VoIP)，与网路会议服务开始整合为兼有影像会议与即时信息的功能。于是，这些媒体的分别变的越来越模糊。  1.2 应用现状  最早的即时通讯软体是ICQ，ICQ是英文中I seek you的谐音，意思是我找你。四名以色列青年于1996年7月成立Mirabilis公司，并在11月份发布了最初的ICQ版本，在六个月内有85万用户注册使用。  　　早期的ICQ很不稳定，尽管如此，还是受到大众的欢迎，雅虎也推出Yahoo! pager，美国在线也将具有即时通讯功能的AOL包装在Netscape Communicator，而后微软更将Windows messenger内建于Microsoft Windows XP作业系统中。  　　腾讯公司推出的腾讯QQ也迅速成为中国最大的即时消息软件。 即时消息软件也面临着互联互通、免费或收费问题的困扰。  即时通信最初是由AOL、微软、雅虎、腾讯等独立于电信运营商的即时通信服务商提供的。但随着其功能日益丰富、应用日益广泛，特别是即时通信增强软件的某些功能如IP电话等，已经在分流和替代传统的电信业务，使得电信运营商不得不采取措施应对这种挑战。2006年6月，中国移动已经推出了自己的即时通信工具———Fetion，中国联通也将推出即时通讯工具“超信”，但由于进入市场较晚，其用户规模和品牌知名度还比不上原有的即时通信服务提供商。  近年来，即时通信市场用户规模增长迅速，市场规模前景广阔。随着互联网用户的快速增长，即时通信用户也日益增长，2004年中国即时通信用户数达7000万人，2005年达到9300万人， 2006年中国即时通信用户增长29%，达到1.2亿人。另外，iResearch根据MorganStanley发表的研究数字整理发现，中国即时通信市场规模在2005年约为6200万美元， 2006年达到7700万美元，到2010年将达到1.14亿美元。  移动即时通信市场有着更加巨大的成长空间和诱惑力。据iResearch2007年5月发布的调查报告显示，2005年我国移动即时通信的用户已达到500万； 2006年，我国移动即时通信注册用户量达到700万；未来几年移动即时通信用户规模将呈现快速增长，预计到2010年中国移动即时通信用户将达到2000万人。  从1999年腾讯推出OICQ算起，即时通信在中国的发展已经有9年的时间。目前国内即时通信领域的竞争呈现“诸侯混战”的态势。其中，腾讯QQ一枝独秀，注册用户数已超过了5.3亿，同时最高在线人数突破l950万；MSN进军中国市场，在没有花大力气的情况下也发展迅速；各大门户网站纷纷挤入即时通信市场跑马圈地；国内的电信运营商也是虎视眈眈。据艾瑞咨询2005年对即时通信市场的调查统计，QQ的市场占有率达到78.8%，MSN为13%，网易泡泡为1.6%，淘宝旺旺为2.1%，雅虎通为0.7%，其他即时通信软件占3.1%。  可见，当前即时通信市场的格局处于垄断竞争状态，代表娱乐方向的腾讯和代表商务方向的微软MSN已成为主流即时通信提供商。在确立了各自领域的领先者地位后，它们又纷纷向对方的用户群体渗透，如腾讯推出了用于商务环境的客户端软件，而MSN也在客户端软件中增加了诸多娱乐要素以吸引娱乐用户。同时，由于进入门槛较低，即时通信系统的同质化竞争日益突出，加剧了市场的竞争，今后即时通信市场的决胜将取决于内容和增值服务的创新上。  我国的移动运营商从几年前就开始涉足即时通信市场。最初是与主流的即时通信软件和业务提供商合作开展业务。例如，中国移动和腾讯合作，提供移动QQ；中国联通也和MSN、腾讯合作，推出了移动即时通信业务。目前，移动运营商已经清楚地认识到即时通信巨大的市场前景和利润，已不仅仅满足于提供网络平台所获得的收入，而是希望直接参与到即时通信的运营之中，从“网络提供商”转变到“服务提供商”，也都下决心要发展自己掌控的即时通信业务。移动运营商的进入必将使这个竞争已经非常激烈的市场更加激烈，并将使即时通信市场的竞争格局发生显着变化。  1.3研究内容与创新工作  1.3.1研究内容  当前市面上大部分通讯软件具有完备的功能但在后台运行占用大量CPU、内存等系统资源，计划通过释放无用功能占用的大量系统资源降低对系统的占用，同时使软件具有较为完备的功能；去除市场上大多数通讯软件中复杂花哨且无用的功能，精简界面，并加强主要功能  1.3.2 研究目标   开发精简的聊天通讯软件  能满足用户日常一对一通讯、群组（聊天室）通讯、用户间文件传输共享功能。 对系统资源占用少  降低各项功能对系统资源的占用，及时释放系统资源，尽可能减少由此软件造成的系统卡顿   去除无用功能  去除如花边新闻推送等无用功能，突出即时通讯属性  2.相关技术背景介绍  2.1 开发语言与开发工具介绍  2.1.1 Java EE 简介  Java se是由Sun Microsystems公司于1995年5月推出的Java[程序设计语言](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%E8%AF%AD%E8%A8%80)和[Java平台](https://baike.baidu.com/item/Java%E5%B9%B3%E5%8F%B0)的总称。用Java实现的HotJava浏览器（支持Java applet）显示了Java的魅力：跨平台、动态的Web、Internet计算。从此，Java被广泛接受并推动了Web的迅速发展，常用的浏览器现在均支持Java applet。  Java SE 是Java平台标准版的简称（***Java Platform, Standard Edition***） (***also known as Java 2 Platform***) ，用于开发和部署桌面、服务器以及嵌入设备和实时环境中的*Java*应用程序。*Java SE*包括用于开发*Java Web*服务的类库，同时，*Java SE*为*Java EE和Java ME*提供了基础。  *Java SE*（*Java Platform, Standard Edition，Java*标准版）就是基于*JDK*和*JRE*的。  *Java*的安全模型可以追溯到该平台的早期时代，当时人们主要将它看作一种增强用户体验的浏览器扩展机制。执行的*Java*代码可以从各种源派生，而其中一些的来源是未知的或者不可靠的。相应地，该平台的安全性最初主要关注的是解决验证被执行的代码可信任的问题，而且整个游戏围绕着在浏览器中执行*applet*。但是，这个模型只是简单地划分为*trusted*和*untrusted*部分，甚至连中等复杂的应用程序都无法运行。  从 1.2 版开始，*Java*作为一个编程平台逐渐得到了人们的认可，而不再只是一个浏览器扩展，*Sun*开始提供更加灵活的安全功能，首先是可配置安全策略的概念。*Java*文档 介绍了它的发展情况。  当*Java*平台开始进入企业环境时，很快就会明显地感觉到，纯粹基于代码的功能无法管理大型应用程序的安全性。*Java*平台的1.4 版引入了一种叫做*Java Authentication and Authorization Service (JAAS)*的新特性，用于将基于用户的权限项整合到安全策略中。现在，[堆栈](https://baike.baidu.com/item/%E5%A0%86%E6%A0%88)上某个特殊代码框架的权限既基于代码的源(其 CodeSource)，又基于验证时分配给用户的身份、组和角色。   * + 1. IDE Eclipse简介   Eclipse是一个开放源代码的软件开发项目，专注于为高度集成的工具开发提供一个全功能的、具有商业品质的工业平台。它主要由**Eclipse项目**、**Eclipse工具项目**和**Eclipse技术项目**三个项目组成，具体包括四个部分组成——Eclipse Platform、JDT、CDT和PDE。JDT支持Java开发、CDT支持C开发、PDE用来支持插件开发，Eclipse Platform则是一个开放的可扩展IDE，提供了一个通用的[开发平台](https://baike.baidu.com/item/%E5%BC%80%E5%8F%91%E5%B9%B3%E5%8F%B0)。它提供建造块和构造并运行集成[软件开发工具](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E5%BC%80%E5%8F%91%E5%B7%A5%E5%85%B7)的基础。Eclipse Platform允许工具建造者独立开发与他人工具无缝集成的工具从而无须分辨一个工具功能在哪里结束，而另一个工具功能在哪里开始。  3.系统架构设计  3.1关键技术设计 | | | |

|  |
| --- |
| 1. **socket网络编程**   1.计算机与计算机间的通信:需要地址  互联网协议(IP)地址用来唯一标识互联网上的计算机  IP:4段用点隔开的0-255的十进制数  转换为域名(好记)  域名服务器(DNS):用来将主机名字转换成IP  互联网协议:以包的形式，低层协议  较高层:传输控制协议(TCP)和用户数据报协议(UDP)  Java支持基于流的通信(TCP)和基于包的通信(UDP)  TCP:能够发现丢失的传输信息并重新发送(不会丢失)UDP会丢失  2.Socket:  套接字:两台主机之间逻辑连接的端点  一个服务器和多个客户端之间的通信:客户端发送请求，服务器响应请求，一旦建立连接，客户端和服务器通过套接字通信  3.服务器套接字:创建服务器需要创建一个服务器套接字，并将它附加到一个端口上，服务器从这个端口监听连接，端口标识套接字上的TCP服务  端口:0-65536，特定0-1024  在端口上创建服务器套接字serverSocket:ServerSocket serversocket=new Serversocket(port)  如果试图在已使用的端口上创建服务器套接字，就会导致java.net.BindException异常  4.客户端套接字:  监听连接:Socket socket=serversocket.accept();  打开一个套接字，请求连接:Socket socket=new Socket(serverName,port)  serverName可以为IP或者域名，如果是域名则DNS转换为IP  程序可以使用主机名localhost或IP127.0.0.1引用客户端运行的计算机  如果在不同的计算机上运行服务器和客户端，就将localhost替换为服务器的主机名或IP地址  如果不能找到主机的话，抛出异常java.net.UnknownHostException  如果服务器没有运行，客户端程序将会异常java.net.ConnectException终止  5.通过套接字进行数据传输  输入流和输出流:getInputStream(),getOutputStream()  创建InputStream和OutputStream流  InptStream input=socket.getInputStream();  OutputStream output=socket.getOutputStream();  InputStream流读取字节  OutputStream流写入字节  DataInputStream、BufferedReader、PrintWriter包装InputStream和OutputStream以读写int,double,string之类的数据  DataInputStream input=new DataInputStream(socket.getInputStream)  Input.readDouble()从客户端接收double类型数据  由于文本文件需要编码和解码，所以二进制I/O比文本I/O效率更高，所以最好用二进制  创建一个服务器套接字时，必须为其指定一个端口，当客户端与服务器相连时，在客户端上创建一个套接字。这个套接字有自己的本地端口，端口个数由java虚拟机自动选取  如果想看到客户端的本地端口，System.out.ptintln(“local port:”+socket.getLocalPort())  6.InetAddress类  获得客户端Ip地址和主机名字  得到与客户端相连的套接字上的一个InetAddress实例:  InetAddress inetAddress=socket.getInetAddress();  显示客户端的主机名和IP地址:  System.out.println(“Client’s host name is”+inetAddress.getHostName());  System.out.println(“Client’s IP Address is”+inetAddress.getHostAddress());  使用静态方法getByName通过主机名或IP地址创建一个InetAddress实例  InetAddress address=InetAddress.getByName(“liang.armstrong.edu”)   1. **Java GUI设计**   **. JavaGUI主要开发工具 -- Swing类库的诞生与功能**  　　一个合格的Java Developer，不仅要掌握技术，还要有一定的Java历史背景知识储备。所以先简要介绍一下用于JavaGUI开发的主要类库：Swing。  　　在Java 1.0时代便有设计GUI的基本类库Abstract Window Toolkit，简称**AWT**。AWT库工作原理是将处理用户界面元素的任务委派给目标平台（操作系统）的本地GUI工具箱，由本地GUI工具箱负责用户界面元素的创建和动作。这种工作方式是**有利也有弊**，先说下利处：   * 处理速度可能要快一点。 * 可以适应不同的平台，“一次编写，随处使用”。   　　弊处：   * 观感依赖于目标平台。 * 有些平台并没有像Windows或Mac这样丰富的界面组件（早期）。所以把AWT的设计工作限制在了“最小公分母”。 * 不同平台存在不同的bug。   　　1996年，**Netscape**创建了另一种GUI库IFC，他的工作方式是 **将用户界面组件绘制在空白窗口上，而对等体只需要负责创建和绘制空白窗口**。Sun和Netscape合作并完善了这种方式，创建了名为Swing的用户界面库，这便是Swing的**诞生**。   　　但是Swing并没有完全取代AWT，到目前Java SE 8中依旧有AWT与Swing两个功能类库：   * AWT import java.awt （java包为核心包） * Swing import javax.swing （javax包为功能扩展包）   　　Swing没有完全取代AWT的原因是：Swing是基于AWT的架构之上，Swing仅仅是提供了能力更强大的用户界面组件。在Swing编写的程序中，还是需要AWT进行**事件处理**。简单说就是，**Swing是用户界面类，AWT是底层机制**。  https://images2015.cnblogs.com/blog/1036165/201610/1036165-20161014174608203-282685719.jpg  AWT和Swing中框架和组件类的继承层次      **2.创建JFrame框架**  Frame意为框架，也就是最顶层的窗口，可以在框架里添加组件。我们创建一个窗口首先要创建一个框架。  https://images2015.cnblogs.com/blog/1036165/201610/1036165-20161018182422310-62542542.jpg  注：Swing组件类都以“J”开头，如 JButton，JFrame等，AWT组件不带“J"。如果Swing组件和AWT组件一起用可能会导致视觉和行为的不一致。 |

工作成果

1. 实现即时通讯软件的GUI界面
2. 实现即时通讯软件用户与用户间根据用户IP地址及端口号建立通信
3. 实现用户使用过程中点对点的通信、点对多的通信
4. 实现用户与用户间点对点的文件传输

为降低系统资源占用、出现系统漏洞风险，用户与用户间的文字通信采用了UDP协议进行传输，通过非阻塞式确保降低系统占用及死锁风险，文件传输也同样采用UDP协议进行用户间的文件传输；

文件发送过程中，发送方输入文件路径，点击右侧（Send）发送按钮，即可向对方用户发送该路径下指定文件（需该文件存在）；发送方点击发送按钮发送后，接收方会弹出接收文件按钮，用户输入储存路径及名称后即可储存对方发送的文件至本地。

对摄像头的调用，通过使用opencv.jar、javacpp.jar、javacv等包，通过包内提供的一系列函数实现调用本机摄像头并录制摄像头捕捉到的影像。通过以下方法实现视频录取：

**package** testCV;

**import** javax.swing.JFrame;

**import** org.bytedeco.javacpp.Loader;

**import** org.bytedeco.javacpp.avcodec;

**import** org.bytedeco.javacpp.opencv\_core.IplImage;

**import** org.bytedeco.javacpp.opencv\_objdetect;

**import** org.bytedeco.javacv.CanvasFrame;

**import** org.bytedeco.javacv.Frame;

**import** org.bytedeco.javacv.FrameGrabber;

**import** org.bytedeco.javacv.FrameRecorder;

**import** org.bytedeco.javacv.OpenCVFrameConverter;

**public** **class** testCV {

**public** **static** **void** recordCamera(String outputFile, **double** frameRate)

**throws** Exception, InterruptedException, org.bytedeco.javacv.FrameRecorder.Exception {

Loader.*load*(opencv\_objdetect.**class**);

FrameGrabber grabber = FrameGrabber.*createDefault*(0);//本机摄像头默认0，这里使用javacv的抓取器

grabber.start();//开启抓取器

OpenCVFrameConverter.ToIplImage converter = **new** OpenCVFrameConverter.ToIplImage();//转换器

IplImage grabbedImage = converter.convert(grabber.grab());//抓取一帧视频并将其转换为图像

**int** width = grabbedImage.width();

|  |
| --- |
| **int** height = grabbedImage.height();    FrameRecorder recorder = FrameRecorder.*createDefault*(outputFile, width, height);  recorder.setVideoCodec(avcodec.***AV\_CODEC\_ID\_H264***); // avcodec.AV\_CODEC\_ID\_H264，编码  recorder.setFormat("flv");//封装格式  recorder.setFrameRate(frameRate);    recorder.start();//开启录制器  **long** startTime=0;  **long** videoTS=0;  CanvasFrame frame = **new** CanvasFrame("camera", CanvasFrame.*getDefaultGamma*() / grabber.getGamma());  frame.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);  frame.setAlwaysOnTop(**true**);  Frame rotatedFrame=converter.convert(grabbedImage);  **while** (frame.isVisible() && (grabbedImage = converter.convert(grabber.grab())) != **null**) {  rotatedFrame = converter.convert(grabbedImage);  frame.showImage(rotatedFrame);  **if** (startTime == 0) {  startTime = System.*currentTimeMillis*();  }  videoTS = 1000 \* (System.*currentTimeMillis*() - startTime);  recorder.setTimestamp(videoTS);  recorder.record(rotatedFrame);  Thread.*sleep*(40);  }  frame.dispose();  recorder.stop();  recorder.release();  grabber.stop();    }  } |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **计划及进度安排**  **程序开发工作计划**   |  |  |  | | --- | --- | --- | | 时间 | 研究内容 | 预期效果 | | 2018年4月上旬 | 需求调研，资料查询，前期调研和技术储备。 | 对研究的需求进行彻底分析，大概确定项目开发所需要使用的技术手段，并确定本系统可能使用到的开发算法。 | | 2018年4月中至2018年5月 | 系统架构设计和开发环境搭建阶段。根据前期的调研和学习，着手进行系统架构设计，搭建开发环境，并对基础功能开始开发。 | 程序的系统架构设计完成，并完成程序的基础功能设计，完成环境搭建，完成程序的基础功能。 | | 2018年5月 | 正式开发阶段。完成项目架构设计、数据库表设计、前端设计、后端设计。完成基本功能如聊天、语音视频通话等。 | 整个系统研发接近完成，完成所有需求和功能。 | | 2018年6月上旬 | 程序测试和优化阶段。完善设计，增添相关功能。 | 完善程序需求和功能，进行程序内存需求等优化。 | | 2018年6月中至2018年7月 | 论文撰写阶段。整理程序开发期间的知识所得，并进行论文撰写。 | 完成论文的撰写。 | |
| **问题及整改方案**  本课题当前版本的开发已经告一段落，主要存在的问题是GUI过于简单，无法显示对方信息等，在今后的开发改进中，会扩展系统性能，并在多样环境下进行充分测试，以期对系统进行最大限度的优化和改进。 |

|  |
| --- |
|  |
| |  |  |  |  | | --- | --- | --- | --- | | 姓 名 | 职 称 | 职务 | 工 作 单 位 | |  |  | 组长 | 北京邮电大学 | |  |  | 成员 | 北京邮电大学 | |  |  | 成员 | 北京邮电大学 |   **评审小组** |

|  |
| --- |
| **导师评语**  论文工作正常进行，后续抓紧时间完成毕业论文撰写。 |
| 导师：  日期： 年 月 日 |
| **阶段报告小组意见：** |
| 负责人：  日期： 年 月 日 |
| **学院意见：** |
| 负责人：  日期： 年 月 日 （签章） |