

深圳大学实验报告

课程名称: Java 程序设计

实验项目名称: 必实验 6 网络编程应用

学院: 计算机与软件学院

专业: 计算机科学与技术

指导教师: 潘微科

报告人: 黎浩然 学号: 2018112061 班级: 01

实验时间: 2019 年 12 月 20 日 (周五) -2019 年 12 月 25 日 (周三)

实验报告提交时间:_____

教务部制

实验目的与要求：

实验目的： 掌握网络通信协议及相关程序设计。

实验要求：

(1). 利用数据报通信方式试编写一程序，该程序生成两个客户端，一个服务器端，两个客户端可以相互进行简短的文字交流。在报告中附上程序截图、完整的运行结果和简要文字说明。(40 分：数据报通信 5 分，两个客户端 5 分，一个服务器端 5 分，实现文字交流 10 分，程序注释和截图 5 分，运行结果截图 5 分，文字说明 5 分)

(2). 编写 Java 应用程序，根据作业“单机版中英文单词测试程序”的要求，将之改为网络版。客户端和服务器端建立套接字连接后，服务器端向客户端发送一个题目；客户端将答案发送给服务器端；服务器端判断客户端的答案是否正确...

要求使用图形界面，其他要求同“单机版中英文单词测试程序”。

在报告中附上程序截图、完整的运行结果和简要文字说明。(40 分：套接字连接 5 分，客户端和服务器端实现双向通信 5 分，界面美观/用户体验 5 分，单选/多选混合出题和题库 5 分，详细的程序注释和截图 5 分，完整的测试过程截图 5 分、最终的测试结果截图 5 分，文字说明 5 分)

报告写作。要求：主要思路有明确的说明，重点代码有详细的注释，行文逻辑清晰可读性强，报告整体写作较为专业。**(20 分，这一项的评分，采用 20 分、10 分和 0 分三个级别)**

说明：

(1) 本次实验课作业满分为 100 分，占总成绩的比例（待定）。

(2) 本次实验课作业截至时间 2019 年 12 月 25 日（周三）23:59。

(3) 报告正文：请在指定位置填写，本次实验**需要单独提交源程序文件**。

(4) 个人信息：**WORD** 文件名中的“姓名”、“学号”，请改为你的姓名和学号；实验报告的首页，请准确填写“学院”、“专业”、“报告人”、“学号”、“班级”、“实验报告提交时间”等信息。

(5) 提交方式：截至时间前，请在 Blackboard 平台中提交。

(6) 发现抄袭（包括复制&粘贴整句话、整张图），该次作业记零分。

(7) 延迟提交，不得分；如有特殊情况，请于截至日期之后 **48 小时内** 发邮件到 panweike@szu.edu.cn，并在邮件中注明课程名称、作业名称、姓名、学号等信息，以及

特殊情况说明，我收到后会及时回复。

（8）期末考试阶段补交无效。

(1). 利用数据报通信方式试编写一程序，该程序生成两个客户端，一个服务器端，两个客户端可以相互进行简短的文字交流。在报告中附上程序截图、完整的运行结果和简要文字说明。(40分：数据报通信5分，两个客户端5分，一个服务器端5分，实现文字交流10分，程序注释和截图5分，运行结果截图5分，文字说明5分)

程序如下：

建立三个 public 类，分别为 ClientA，ClientB 和 Server，因为对于 ClientA 和 ClientB 有相似的结构，这里只介绍 ClientA 和 Server

ClientA：

```
// TODO Auto-generated method stub
DatagramSocket datagramsocket = null;
DatagramPacket datagrampacket = null;
Scanner in = new Scanner(System.in); //获取标准输入流
try {
    datagramsocket = new DatagramSocket(4555); //建立数据报套接字 端口号为2555
    String str = null;
    System.out.println("Connected:"); //已连接
    System.out.print("My: ");
    new Thread(new AcceptMsgA(datagramsocket)).start();
    while(!(str=in.nextLine()).contains("bye")) { //当输入的信息不含有bye //保持连接
        str = str+ " from A"; //加上信息后缀 表示信息来自A
        datagrampacket = new DatagramPacket(str.getBytes(),str.getBytes().length,
            InetAddress.getByName("127.0.0.1"),8000); //服务器端口为8000
        datagramsocket.send(datagrampacket); //发送
        System.out.print("My: ");
    }
    str = str+ " from A";
    datagrampacket = new DatagramPacket(str.getBytes(),str.getBytes().length,
        InetAddress.getByName("127.0.0.1"),8000); //服务器端口为8000
    in.close(); //关闭输入流
    System.out.println("Disconnected!"); //断开连接
} catch (SocketException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (UnknownHostException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

这里，为了实现客户端同时发送和接受信息，我使用了两个线程，主线程负责发送信息；AcceptMsg 是实现了 Runnable 接口的一个类，负责接收信息；同时，当 AcceptMsg 接受到含有 bye 的字符串时，不再接受信息。

```

class AcceptMsgA implements Runnable {
    DatagramSocket socket = null;
    AcceptMsgA(DatagramSocket socket) {
        this.socket = socket;
    }
    public void run() {
        byte [] data = new byte[1024];
        DatagramPacket packet = new DatagramPacket(data,data.length);
        while(true) {
            try {
                socket.receive(packet);
                if(new String(data,"utf-8").trim().contains("bye")) { //如果接收到bye, 这不再接受信息
                    System.out.println();
                    System.out.println(new String(data,"utf-8").trim());
                    break;
                }
                System.out.println();
                System.out.println(new String(data,"utf-8").trim()); //输出收到的信息
                System.out.print("My: ");
                data = new byte[1024];
                packet = new DatagramPacket(data,data.length);
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }
}

```

ClientB 与 ClientA 类相似，可以自行查看源代码.

对于 Server 类，我采用以下策略：ClientA 和 ClientB 每次发送信息时自动加上 from A 和 from B 后缀表示信息来源，然后 Server 根据后缀判断信息来自何方并将信息转发到另一个 Client

```

public class Server {
    private static final int CLIENTA = 4555;
    private static final int CLIENTB = 3555;
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        DatagramSocket datagramsocket = null;
        DatagramPacket datagrampacket = null;
        byte [] data = new byte[1024];
        try {
            datagramsocket = new DatagramSocket(8000);           //建立数据报套接字 端口号8000
            while(true) {
                datagrampacket = new DatagramPacket(data,data.length); //package //关联data
                System.out.println("Waiting message:");
                datagramsocket.receive(datagrampacket);           //接受信息
                if(new String(data,"utf-8").contains("from A")) {   //信息来自于A
                    System.out.println("A message from A: ");
                    System.out.println(new String(data,"utf-8").trim());
                    if(new String(data,"utf-8").contains("bye")) {
                        datagrampacket = new DatagramPacket(data,
                            data.length,InetAddress.getByName("127.0.0.1"),CLIENTB);
                        datagramsocket.send(datagrampacket);
                        data = null;
                        break;
                    }
                    datagrampacket = new DatagramPacket(data,
                        data.length,InetAddress.getByName("127.0.0.1"),CLIENTB);
                    datagramsocket.send(datagrampacket);
                    data = new byte[1024];
                }
                else if(new String(data,"utf-8").contains("from B")) { //信息来自于B
                    System.out.println("A message from B: ");
                }
            }
        }
    }
}

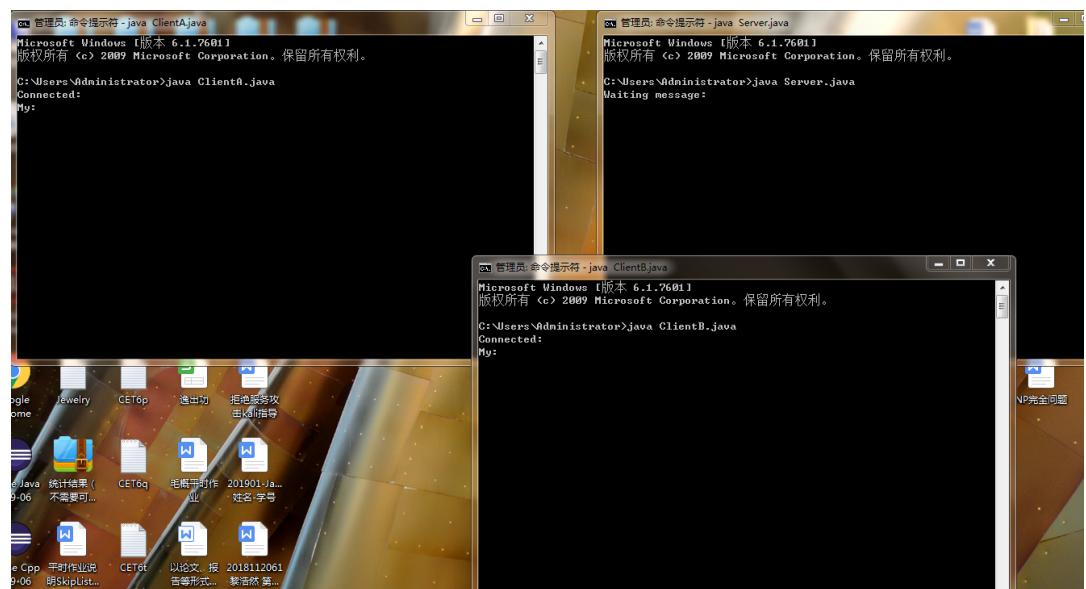
```

```
        System.out.println("A message from B: ");
        System.out.println(new String(data,"utf-8").trim());
        if(new String(data,"utf-8").contains("bye")) {
            datagrampacket = new DatagramPacket(data,
                data.length,InetAddress.getByName("127.0.0.1"),CLIENTA);
            datagramsocket.send(datagrampacket);
            data = null;
            break;
        }
        datagrampacket = new DatagramPacket(data,    //转发信息
            data.length,InetAddress.getByName("127.0.0.1"),CLIENTA);
        datagramsocket.send(datagrampacket);
        data = new byte[1024];
    }
    datagramsocket.close();
} catch (SocketException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}
}
```

但当 Server 遇到 bye 字符串时，则在转发完最后一次信息（bye）后跳出循环，不再转发来自任何一方的字符串。

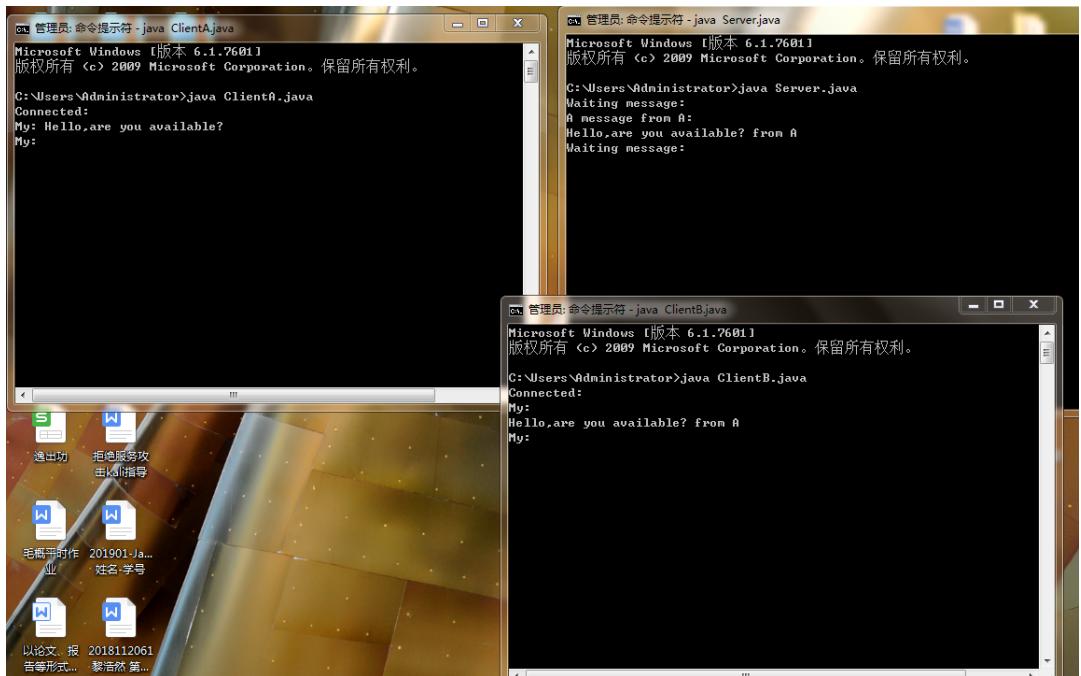
运行结果如下：（由于编码问题，暂不支持中文聊天）

在 cmd 下编译程序：



左方为 ClientA，右下方为 ClientB，右上方为 Server

A 给 B 发送 “Hi, are you available” 后



The image shows three separate Windows command-line windows (cmd.exe) running as administrators. The windows are arranged in a grid-like fashion. The top-left window is titled '管理员: 命令提示符 - java ClientA.java'. It displays the following text:
Microsoft Windows [版本 6.1.7601]
版权所有 <©> 2009 Microsoft Corporation。保留所有权利。
C:\Users\Administrator>java ClientA.java
Connected:
My: Hello, are you available?
My:
The top-right window is titled '管理员: 命令提示符 - java Server.java'. It displays:
Microsoft Windows [版本 6.1.7601]
版权所有 <©> 2009 Microsoft Corporation。保留所有权利。
C:\Users\Administrator>java Server.java
Waiting message:
A message from A:
Hello, are you available? from A
Waiting message:
The bottom-left window is titled '管理员: 命令提示符 - java ClientB.java'. It displays:
Microsoft Windows [版本 6.1.7601]
版权所有 <©> 2009 Microsoft Corporation。保留所有权利。
C:\Users\Administrator>java ClientB.java
Connected:
My:
Hello, are you available? from A
My:
The background of the desktop shows several document icons on a dark blue background.

一段闲聊之后：

```

c:\ 管理员:命令提示符 - java ClientA.java
Microsoft Windows [版本 6.1.7601]
版权所有 <c> 2009 Microsoft Corporation. 保留所有权利。
C:\Users\Administrator>java ClientA.java
Connected:
My: Hello, are you available?
My: Yes, what's your problem? from B
My: I think Java homework is too hard!
My: Oh, I think you should find the interest to learn java from B
My: Eh, but I do not know how to find resources about java
My: You can try to look Thinking in java from B
My:

c:\ 管理员:命令提示符 - java Server.java
Microsoft Windows [版本 6.1.7601]
版权所有 <c> 2009 Microsoft Corporation. 保留所有权利。
C:\Users\Administrator>java Server.java
Waiting message:
A message from A:
Hello, are you available? from A
Waiting message:
A message from B:
Yes, what's your problem? from B
Waiting message:
A message from A:
I think Java homework is too hard! from A
Waiting message:
A message from B:
Oh, I think you should find the interest to learn java from B
Waiting message:
A message from A:

命令提示符 - java ClientB.java
Windows [版本 6.1.7601]
(c) 2009 Microsoft Corporation. 保留所有权利。
Administrator>java ClientB.java
My:
Hello, are you available? from A
My: Yes, what's your problem?
My:
I think Java homework is too hard! from A
My: Oh, I think you should find the interest to learn java
My:
Eh, but I do not know how to find resources about java from A
My: You can try to look Thinking in java
My:

```

Server 保存并显示所有聊天记录，以判断是否有人在进行违法犯罪行为

(2). 编写 Java 应用程序，根据作业“单机版中英文单词测试程序”的要求，将之改为网络版。客户端和服务器端建立套接字连接后，服务器端向客户端发送一个中文（或英文）单词和四个中文（或英文）解释（分别为 A,B,C,D），可以有多个解释是正确的（也就是多选题）；客户端将答案发送给服务器端；服务器端判断客户端的答案是否正确（如果错选或漏选均得零分），并将已经给出的题目的数量和答对的数量返回给客户端，同时也把下一个中文（或英文）单词和四个中文（或英文）解释发送给客户端；如此循环 10 次结束单词测试；测试结束后，服务器端将答题的详细情况保存到一个本地文本文件（包括每个题目及四个选项、正确答案、客户端提交的答案、每一题答对或答错、总的答对题数和答错题数），并将该文本文件传送给客户端保存。要求使用图形界面。词库应至少包含 10 个中文和英文单词，以及相应的正确解释和错误解释。注意：出题的时候要求中文和英文混着出题。在报告中附上程序截图、完整的运行结果和简要文字说明。(40 分：套接字连接 5 分，客户端和服务器端实现双向通信 5 分，界面美观/用户体验 5 分，中英文混合出题和题库 5 分，详细的程序注释和截图 5 分，完整的 10 次测试结果截图 5 分，最终文件的保存和传输 5 分，文字说明 5 分)

程序如下：

服务器端类：

```
public class Server {
    private static final int SERVER_PORT = 9999;
    ServerSocket server = null;          //服务器端套接字
    GenerateWords thread;               //线程
    Socket client = null;               //客户端套接字

    public Server() {
        try {
            server = new ServerSocket(SERVER_PORT);
        } catch (IOException e) {
            e.printStackTrace();
        }
        while(true) {
            try {
                client = server.accept();
                System.out.println("客户端的地址为: " + client.getInetAddress());
            } catch (IOException e) {
                e.printStackTrace();
                System.out.println("等待客户端连接...");
            }
            if(client != null) {
                GenerateWords temp = new GenerateWords(client);
                new Thread(temp).start();
            }
        }
    }
}
```

服务器端会在程序运行的时候在标准输出流中显示一些信息，每接入一个客户端都会新建一个线程单独服务。

Override Run 方法:

```
@Override
public void run() {
    while (true)
    {
        if(total >= 10)
            break;
        int type = (int) (Math.random()*127);

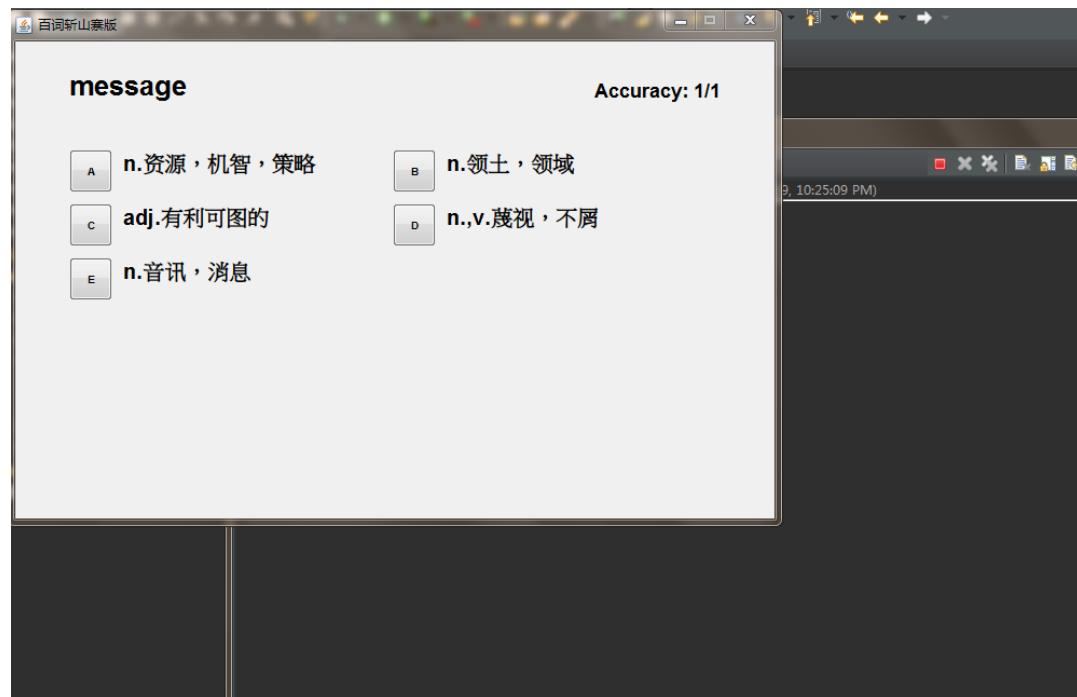
        if(type%3==0)
            getTestEnToCh();
        else if (type%3==1)
            getCHtoEN();
        else
            getMultipleTest();
        //检查回答是否正确
        checkAnswer();
    }
}
```

在控制台中接受设置选项:

```
while(true)
{
    try {
        //接收问题和选项
        String questionText = in.readUTF();
        String choiceA = in.readUTF();
        String choiceB = in.readUTF();
        String choiceC = in.readUTF();
        String choiceD = in.readUTF();
        String choiceE = in.readUTF();
        words.question.setText(questionText);
        words.labelA.setText(choiceA);
        words.labelB.setText(choiceB);
        words.labelC.setText(choiceC);
        words.labelD.setText(choiceD);
        words.labelE.setText(choiceE);
        words.corrects = in.readInt();
        words.total = in.readInt();
        words.setAccuracy();
        if(words.total >= 10 || !words.getFrameState())
            break;
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

其余的类与作业 10 类似

运行结果：





深圳大学学生实验报告用纸

指导教师批阅意见:

成绩评定:

指导教师签字:

2019年 月 日

备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。