**西安电子科技大学**

网

络

计

算

题目： 基于http协议

多线程断点下载器

学院： 计算机学院

班级： 1403021

姓名： 胡鑫男

**目录**

序言 3

一、项目需求 4

1. 问题描述 4

2. 要求 4

二、软件设计 4

1. 开发环境 4

2. 知识介绍 4

2.1. HTTP协议 4

2.2. Range头域 4

2.2．RandomAccessFile实现类了解 5

3. 软件实现结构 5

4. 软件设计思路 6

4.1. 实现多线程下载 6

4.2. 实现断点续传功能 6

三、软件实现 6

1. 请求文件资源 6

2. 文件切割 6

3. 断点续传起点设置 7

4. 设置断点 7

5. 文件片段请求头（Range设置） 7

四、软件测试 7

1．功能演示 7

1.1. 正常文件下载 7

1.2. 模拟网络中断（设置断点为150字节时） 8

2. 测试结果分析及改进 9

2.1．改进意见 9

五、项目总结 9

六、个人总结 10

# 序言

随着网络的发展，用户对网上资源的下载也越来越平凡。那么就要求在下载速率和用户体验上都具有良好的效果。生活中我们经常碰到在下载过程出现人为不可控的状况，比如：突然断电，网络中断等等，都可能导致我们失去之前的下载的链接，那么我们只能重新请求并从文件开端获取资源，直至下载完成，造成了网络流量的大大浪费，为此我们设想实现网络上的断点续传功能并通过多线程并发下载提高下载速率，以下则是整个下载器的设计，实现，测试的具体描述

# 一、项目需求

## 1. 问题描述

实现基于HTTP协议下的多线程断点续传功能

## 2. 要求

（1）演示多线程功能：通过显示每个线程切分的数据块大小

（2）演示断点续传功能：通过设置断点提示下载起始点

# 二、软件设计

## 1. 开发环境

整个软件编写使用Eclipse，模拟下载服务器apache-tomcat-7.0.77，文档编写Office word 2013，作图软件EdrawMax。

## 2. 知识介绍

### 2.1. HTTP协议

HyperText Transfer Protocol，超文本传输协议，是用于从WWW服务器传输超文本到本地浏览器的传输协议。它可以使浏览器更加高效，使网络传输减少。它不仅保证计算机正确快速地传输超文本文档，还确定传输文档中的哪一部分，以及哪部分内容首先显示(如文本先于图形)等。

### 2.2. Range头域

Range头域：可以请求实体的一个或者多个子范围。例如：

（1）表示头500个字节：bytes=0-499

（2）表示第二个500字节：bytes=500-999（本软件中对不同线程以及，断点续传时均需设置此头域）

（3）表示最后500个字节：bytes=-500

（4）表示500字节以后的范围：bytes=500-

（5）第一个和最后一个字节：bytes=0-0,-1

（6）同时指定几个范围：bytes=500-600,601-999

但是服务器可以忽略此请求头，如果无条件GET包含Range请求头，响应会以状态码206（PartialContent）返回而不是以200返回（程序中以206为判断条件）

### 2.2．RandomAccessFile实现类了解

（1）RandomAccessFile类可以对文件随机访问的操作，访问包括读和写操作。该类的读写是基于指针的操作。

（2）RandomAccessFile对文件进行随机访问操作时有两个模式，一种是只读(r)，另一种是读写(rw)，需在创建RandomAccessFile对象时传入一个参数进行设置，第一个参数代表要访问的文件，第二个参数代表设置访问模式 RandomAccessFile(File file,String mode)

RandomAccessFile(String filename,String mode)

只读模式： new RandomAccessFile(“filename”,”r”);

读写模式： new RandomAccessFile(filename,”rw”);

（3）重要方法

void seek(long pos)

使用该方法可以移动指针到指定位置。

## 3. 软件实现结构

1. 客户发出请求文件资源（URL）
2. 服务器反馈文件信息（并由客户端获取文件程长度）
3. 客户端对下载资源进行文件分割，并创建多线程，实现多线程下载
4. 如果有断点，重新下载时先读取上次下载记录，并设置请求资源位置（具体断点纪录，重新请求在软件实现中介绍）
5. 下载成功



## 4. 软件设计思路

### 4.1. 实现多线程下载

（1）获取链接，设置请求头

（2）设置Range：byte＝startIndex－endIndex（请求片段下载）

（3）接受响应（206）

（4）获取链接的输入流

（5）使用RandomAccessFile类对本地目标文件进行写（seek(startIndex)）

（6）关闭当前线程

（7）当所有线程均被关闭后结束

### 4.2. 实现断点续传功能

（1）创建当前线程对应的文件保存文件下载进度。

（2）当在请求时判断该记录保存文件是否存在。存在：本次请求文件下载的开始索引＝上次文件下载进度＋当前设置的文件开始位置

不存在：文件第一次下载

# 三、软件实现

## 1. 请求文件资源

本程序模拟浏览器使用HttpURLConnection对象实现

URL url = new URL(path);

HttpURLConnection connect = (HttpURLConnection) url.openConnection();

connect.setRequestMethod("GET");

connect.setReadTimeout(5000);

connect.setReadTimeout(5000);

## 2. 文件切割

循环遍历文件切割，分配每一个线程请求资源的索引位置

private static void multithreadRequest(int length, int threadNum) {

int everyLength = length/Multidownload.ThreadNum;

for (int i = 0; i < threadNum; i++) {

int startIndex = i \* everyLength;

int endIndex = ((i + 1) \* everyLength) + 1;

//特别注意最后一个线程的结束位置定死，确保文件全部下载

if(i == threadNum-1){

endIndex = length - 1;

}

new Thread(new MyThread(startIndex,endIndex,i)).start();

}}

## 3. 断点续传起点设置

startIndex += Integer.parseInt(bis.readLine());

## 4. 设置断点

if(total == 150){

return;}

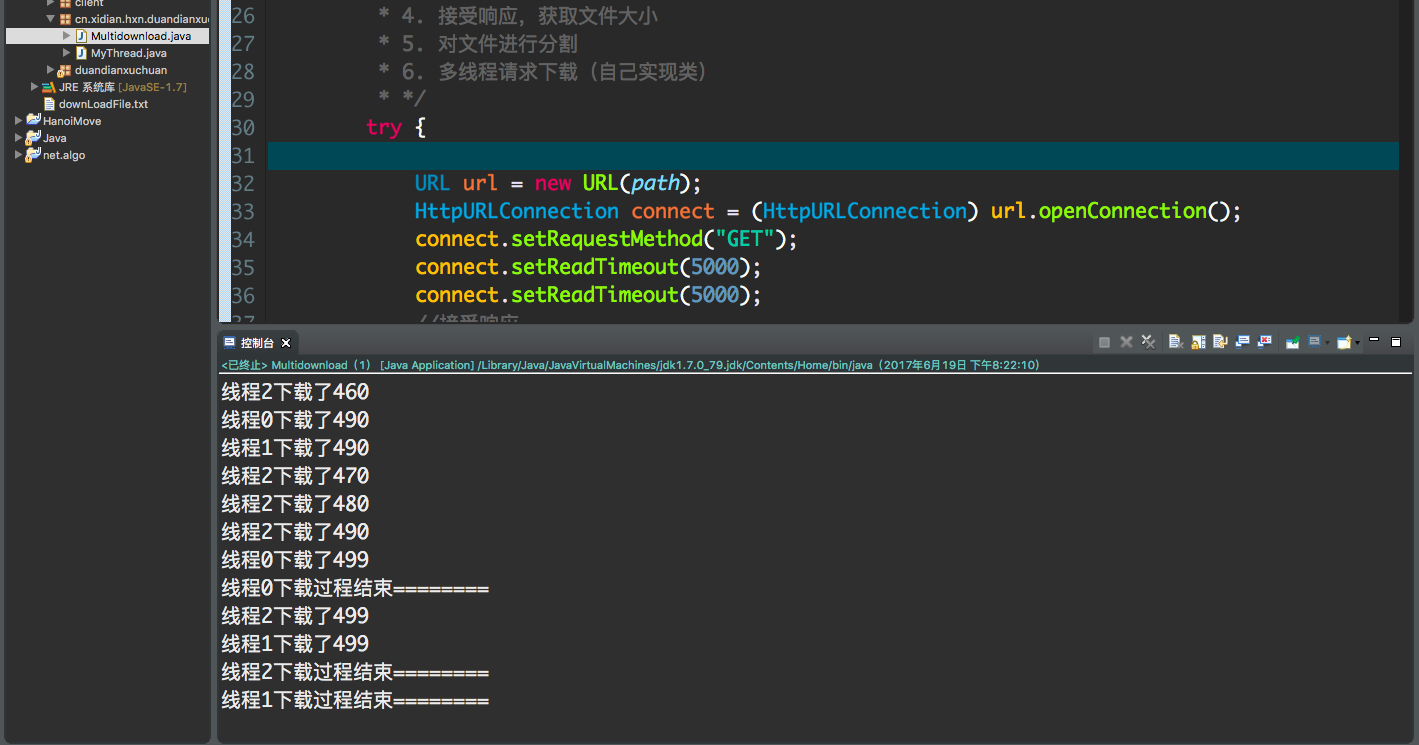
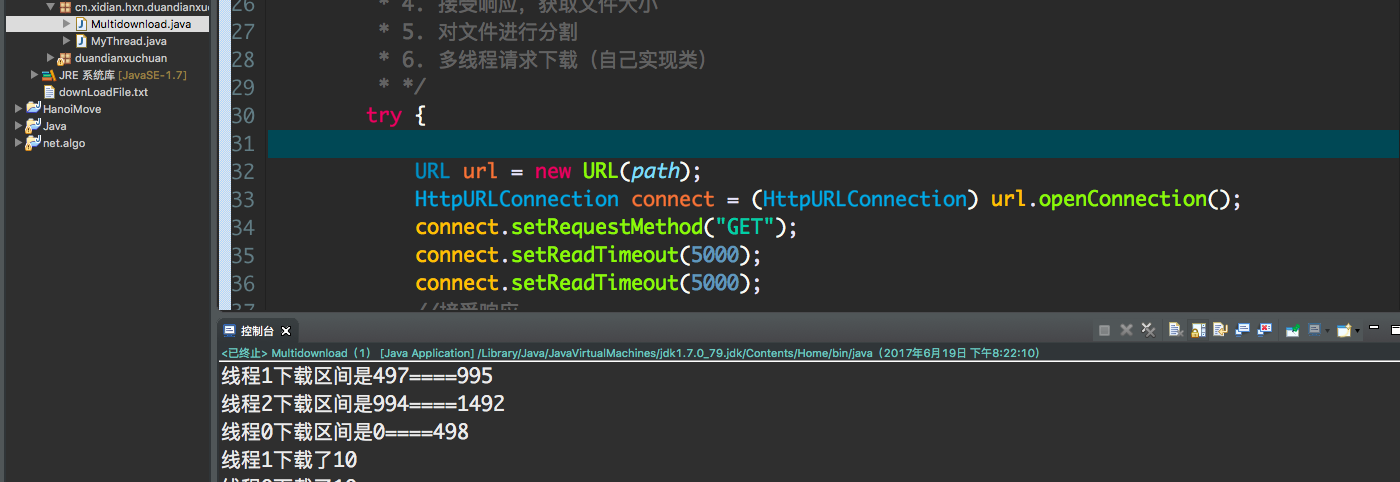
## 5. 文件片段请求头（Range设置）

connect.setRequestProperty("Range", "bytes=" + startIndex + "-" + endIndex);

# 四、软件测试

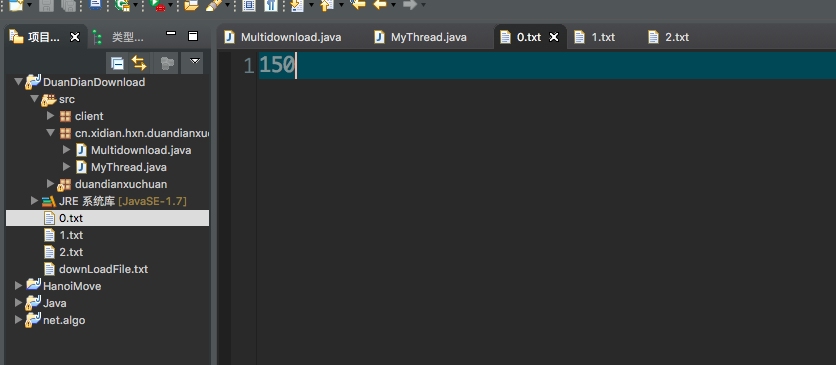
## 1．功能演示

### 1.1. 正常文件下载



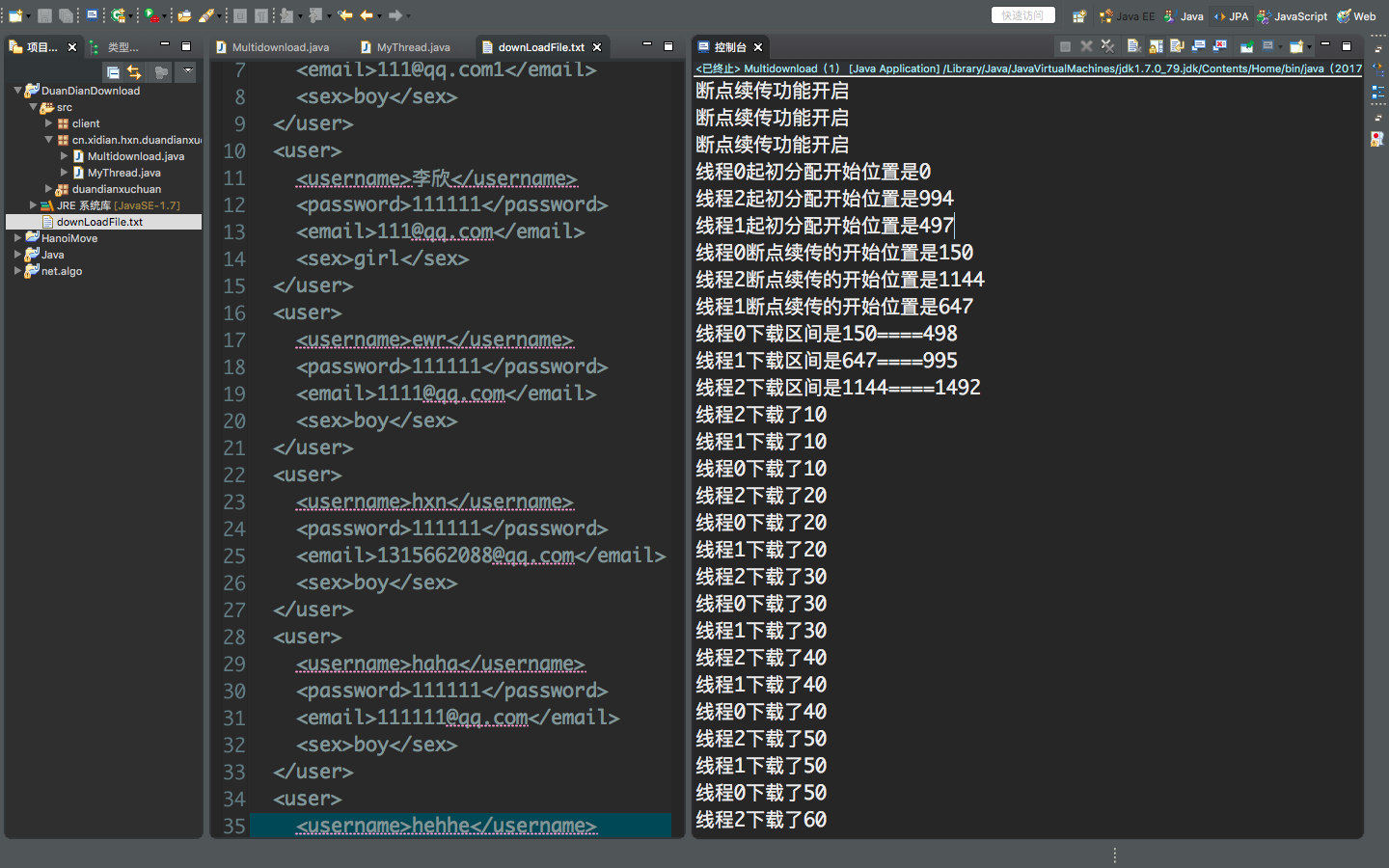
## 1.2. 模拟网络中断（设置断点为150字节时）

可以看到此事临时文件存在，并且当中记录中每一个线程上一次的下载总量

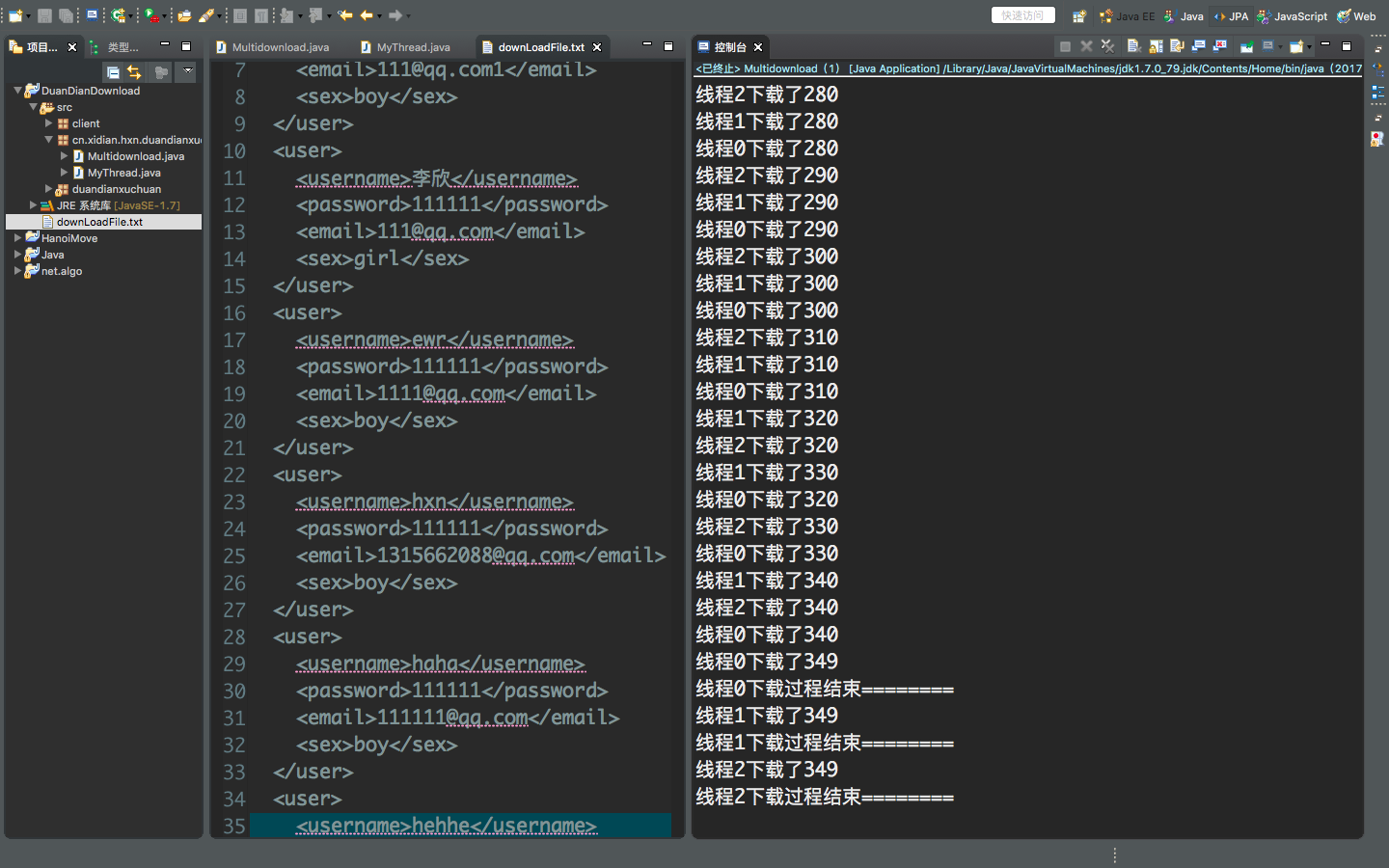


当程序在次运行判断后得知非第一次请求文件，测试输出“初始分配的起事位置”和断点续传重新请求时请求的开始位置，不难计算出我们前后相差150字节，刚

好与我们设置的断点吻合（文件中存放的记录）



直至下载成功



## 2. 测试结果分析及改进

### 2.1．改进意见

# 五、项目总结

本项目实现了文件多线程断点续传的功能，在实现过程中也不乏碰到瓶颈

比如：

1. 如何向服务器请求“片段”文件下载？
2. 如何对每个线程的下载量进行设置？
3. 如何在多线程下载的情况下保证文件被恰好下载完毕？
4. 如何存储断点纪录？
5. 存在断点纪录后如何设置请求位置？

在查阅资料请教同学之后都相继解决，但程序中仍存在不足，希望老师批评指正

# 六、个人总结

胡鑫男：

本学期完成软件“基于HTTP协议下载器”，从中学习到了很多东西也认识到了自己很多的不足。首先在着手做之前，先对HTTP协议作了初步的了解，整体框架的把握，最后去学习HTTP协议通信中的“字段”发现了协议精妙之处，同时也认识到自己离设计者还有很遥远的距离仍然需要努力；着手软件开发时我们从需求分析中总结出主要功能：断点续传＋多线程传输。断点续传：利用HTTP协议本身的Range头域存储我们需要请求的字段区域值。多线程传输问题：获取文件大小后，进行文件的切割，实现多线程下载。其中最关键的问题是如何判断文件是否“续传”以及续传的点：（1）每次传输通道建立，都生成临时文件来保存当前下载的字段数，目的一：检测临时文件是否存在判断是否断点；目的二：读取断点位置，设置请求参数