MULTITHREADED DOWNLOAD MANAGER

PROJECT REPORT

Parallel and Distributed Computing

Submitted

by

Pulkit Malhotra	**17BCE0095**
Rachit Agarwal	**17BCE0304**
Shaurya Singh	**17BCE0911**
Prashant Rana	**17BCE2330**

School of Computer Science & Engineering

Winter Sem 2019-20

TABLE OF CONTENTS

CHAPTER NUMBER-1

INTRODUCTION

1.1 Abstract

Downloading a file is a frequently worked domain where we use the default download managers of a given browser to download a file. While most download managers download the file from the same source, some downloading managers can likewise increase the download speeds by downloading from different sources simultaneously. Despite the fact that browsers may have download administrators in built as an element, they are separated by the way that they don't organize exact, finish and unbroken downloads of data. While some download administrators are completely fledged projects devoted to downloading any data more than at least one convention (e.g. http), many are incorporated into installers or refresh directors and used to download parts of a particular program (or set of projects), eg. contains Google and Adobe's update.

1.2 Organization of Report

The rest of the project is organized as follows.
Chapter 2 deals with Introduction and Literature Survey. Chapter 3 explains proposed system. Chapter 4 focuses system design. Chapter 5 describes system implementation. Chapter 6 includes the outputs and discussions.

CHAPTER NUMBER-2

2.1 Introduction

Download managers were among the top (includes torrent customers as they are actually download managers also) applications showing a flag promotion in the UI. Many download managers accompany the highlights like video and sound retrieving from well-known websites like YouTube etc., They likewise support site snatching. queue handling is another main element of download administrator. They additionally can pause and resume downloads, and force speed confinements too. This highlight come exceptionally helpful in locales where power failures is an issue. Furthermore, a large portion of the business download chiefs can download following client arranged timetables and download in like manner. A couple of download administrators claim to build the download speed by a factor of ordinarily. Download managers additionally have tight integration with engines. For the most part they do this by introducing an expansion to the client's engine(browser). Download speeding up, otherwise called multipart download, is a term for the strategy utilized by programming, for example, download administrators to download a solitary record by part.

2.2 Literature Survey

Title	Author	Journal Name& Date	Key Concepts	Advantages	Disadvantage
Proceedings in the Third IEEE Workshop on Internet Applications	Gkantsidis, M. Ammar and E. Zegura.	IEEE WIAPP 2003	Functionality of parallel downloading When grasped inside the server. Survey on different type of parallel downloading plans &customers point of view on their execution.	Clients easily identify the execution change between multithread parallel downloading and single processor based downloading.	Parallel downloading brings extra overhead due to extra use of system resources.

The future trends of distributed computing	S.G.M. Koo, C. Rosenberg, and Dongyan xu	IEEE FTDS	Most results suggest that while PD may achieve a shorter downloading time, at the same time its impact on the framework and server is important.	Parallel downloading achieves higher collected downloading throughput. It gives shorter downloading time experience to the user.	Negative impact of parallel downloading on Processor framework.
Operating systems with Parallel threading	Zhou Xu, Lu Xianliang, Hou Mengshu and Zhan Chaun	ACM SIGOPS, Operating, Systemsreview, Homepage 2005	Advance parallel downloading is a cutting edge parallel download plan in P2P condition. It also tell about initial investment of threads for each downloading server.	After speedier servers finish their own specific work, parallel downloading reallocates some part of their work to deficient server who works slow.	Due to allocation of threads to slower server, it produces immense pressure on processors.
"Distributed Computing Systems Proceedings"	Jiantao Song, Chaofeng Sha and Hong Zhu	IEEE FTDS 2003	Managing the reception of a large file from the internet through parallel downloading and analyzing of this huge record	In this paper author portrays some arrangement Of parallel downloading approach by which client can improve its performance in general sense.	By this method although the Traffic configuration is optimal from the point of view of the clients ,but it may be bad from the point of view of the systems.
"Benefits of paraloading over single association"	Allen Miu, and Eugene, Shih.	"Laboratory of Computer Science" 2004			
"Distributed Computing Systems Proceedings 2"	J. Funasaka, N. Nakawaki, and K. Ishida.	"Distributed computing workshop" 2003	Here J.Funasaka purposes a adaptable parallel downloading technique for big multiple files .which depends upon the symmetry of the file.	By this methodologies separate a record into level with pieces whose size is settled from the before.	This approach may not be work for files whose size isn't settled before, which will be vary when same no of threads will assign to the all the files.

Title	**Author**	**Journal Name& Date**	**Key Concepts**	**Advantages**	**Disadvantage**	**Future improvement**
Parallel Downloading Using Variable Length Blocks for Proxy Servers	Atsushi Kawano, Junichi Funasaka, Kenji Ishida	27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07) - 30 July 2007	Proposed an proxy server which gets the latest records at rapid speed by utilizing the parallel downloading innovation	Rapid speed. Packets are reordered and sent to the customer.	However, this proxy server has the issue that the cushion may turn out to be too huge.	The piece measure is resolved powerfully as indicated by the system condition with the goal that the involved cushion space contracts.
A generalized basiccycle calculation method for efficient array redistribution	Ching-Hsien Hsu , Sheng-Wen Bai , Yeh-Ching Chung	IEEE Transactions on Parallel and Distributed Systems. Dec 2000	In many scientific applications , dynamic array redistribution is usually required to enhance the performance of an algorithm. They provide a method to do it efficiently.	We present a generalized basiccycle calculation (GBCC) method to efficiently perform a BLOCK-CYCLIC(s) over P processors to BLOCK-CYCLIC(t) over Q processors array redistribution.	The experimental results show that the GBCC method outperforms the PITFALLS method and the ScaLAPACK method for all test samples. No disadvantages could be gathered.	It has proved to provide the best results in dynamic allocation and no further scope for improvement can be found.
Topology - aware server selection method	Y. Higashi , S. Ata	Second IEEE Consumer Communications and Networking	server selection method for parallel	approach tries to minimize the total number of shared links between a client and	if some connections share the same bottleneck link, the throughput is not	Provide a way to virtually handle cases when a bottleneck arises

for dynamic parallel download ing		Conference, 2005. CCNC. 2005 6-6 Jan. 2005	downloadin g by taking the physical network topology into consideratio n.	downloading servers.	increased and the additional connection simply wastes the server resources.	due to physical topology and decide whether parallel is a better approach.
Multithrea ded Pipeline Synthesis for DataParallel Kernels	Mingxing Tan1 , Bin Liu2 , Steve Dai1 , Zhiru Zhang	2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD) , 2-6 Nov. 2014	a multithreade d pipelining approach that enables context switching to allow outof-order thread execution for dataparallel kernels.	proposed techniques can significantly improve the effective pipeline throughput over conventional approaches .	The hardware overhead associated with context management is significantly more.	If a balance is found between hardware overhead and improvement in parallelism in kernel then it could be scalable to a huge level.

Multithreaded Pipeline Synthesis for Data-Parallel Kernels	Mingxing Tan , Bin Liu , Steve Dai , Zhiru Zhang	IEEE/ACM International Conference on Computer-Aided Design (ICCAD) - 2014	Pipelining, ContextSwitching	Pros: A multithread pipelining approach focusing on context switching. Allows out-of-order thread execution for data parallel kernels. Efficient scheduling algorithms to minimize the hardware Overhead in context management Cons: The synthesized pipeline enforces an in-order execution between threads, It could be less effective in avoiding the memory latency with irregular stream of address.	Proposed techniques can significantly improve the effective pipeline throughput over conventional approaches while conserving hardware resources There has been a growing interest in optimizing HLS techniques for irregular applications with non-deterministic workload and unpredictable memory access latency.
EFFECT OF THREAD LEVEL PARALLELISM ON THE PERFORMANCE OF OPTIMUM	Mehdi Alipour , Hojjat Taghdisi	International Journal of Embedded Systems and	Thread Level Parallelism, Optimum single-thread architecture	Explores multi-threaded architectures. Focus on finding optimum single thread architecture as multi-threaded architecture are a derived of these.	

ARCHITECTURE FOR EMBEDDED APPLICATIONS		Applications (IJESA) Vol.2, No.1, March 2012		Results show that running two threads on a singlethread processor with limited area and power budget, in average, Leads to increased performance for some representative embedded applications.	
A Multi-Threaded PIPELINED Web Server Architecture for SMP/SoC Machines	Gyu Sang Choi, Jin-Ha Kim, Deniz Ersoz and Chita R. Das	Proceedings of the 14th international conference on World Wide Web May 10 - 14, 2005	Multi threading, Pipelining, HPC	Results indicate that PIPELINED server architecture is a viable design option for SMP/SoC machines	Implementing proposed model in an SMP machine. Performance Impact analysis of dynamic Web contents on the model proposed
Thread Assignment of Multithreaded Network Applications in Multicore/Multithreaded Processors	Petar Radojkovic, Vladimir Cakarevic, Javier Verdu, Alex Pajuelo, Francisco J. Cazorla,	IEEE Transactions on Parallel and Distributed Systems (Volume: 24, Issue: 12, Dec. 2013)	Thread assignment, Lightweight Kernel (LWK), Blackbox scheduler	Systematic method for thread assignment of multithreaded network applications running on multicore/multithreaded processors. Proposed method is evaluated with an industrial case study for a set of multithreaded network applications	Finding optimal thread assignment on multithreaded processors comprising many cores is an NPcomplete problem
Characterizing the Performance and Energy Efficiency of Simultaneous Multithreading in Multicore Environments	Balaji Subramaniam and Wuchun Feng,	41st International Conference on Parallel Processing Workshops 2012	Concurrency, Multicore processing, Multithreading, simultaneous multithreading	Focus on concurrent throttling. Improving the processor utilization by simultaneous multi threading. Limitation: Benefit of concurrency throttling is highly dependent on the workload	
Parallel File Download in Peer-to-Peer Networks with Random Service Capacities	Keqin Li	IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum 2013	Parallel download and chunk allocation, peer-to-peer downloading algorithms	The problem of efficient parallel file download in peer-to-peer networks is addressed. Various parallelism algorithms compared to find the optimal solution for parallel downlading. Problem: there is an important issue of optimal parallelism which minimizes the combined effect of intracluster and intercluster overhead of parallel download and load imbalance	Further research should take more challenging scenarios into consideration. Benefit from peers with incomplete files

Title	**Author**	**Journal Name& Date**	**Key Concepts**	**Advantages**	**Disadvantages**
Thread-Associative Memory for Multicore and Multithreaded Computing	Shuo Wang and Lei Wang	IEEE Conferences - ISLPED'06 Proceedings of the 2006 International Symposium on Low Power Electronics and Design 2006	Thread Associativity; Memory contention among concurrent threads	1) Incorporating thread-specific information explicitly into onchip memory hardware. 2) The proposed technique can be a solution for energy efficiency	Proposed for only large and less active high-performance caches.
Efficient Development Methodology for Multithreaded Network Application	Nguyen Duc Chinh, Ettikan Kandasamy, Lam Yoke Khei	The 5th Student Conference on Research and Development December 2007	Pitfalls in multithreaded applications	1) Discovery of common pitfalls in multithreaded network applications. 2) Comparison of performance of a single threaded network application with multithreaded network applications.	Difficulty in detection of thread related errors like race errors, deadlocks, etc.
Performance Analysis of a Dynamic Parallel Downloading Scheme from Mirror Sites Throughout the Internet	Allen Miu and Eugene Shih.	Performance Analysis of a Dynamic Parallel Downloading Scheme from Mirror Sites Throughout the Internet, 6.892 Term Paper, December 1999.	Para loading; Parallel Data transfer through Multiple Mirror Servers	Para-loading leads to a new way of retrieving files from the internet.	1)The real time performance gains from para-loading is not as high as theoretical suggestions. 2)Method not suitable to all operating environments
Parallel Downloading Algorithm for Large-volume File Distribution	Haitao Chen, Zhenghu Gong and Zunguo Huang	Sixth International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT'05) 2005	Distribution of Large volume files;	1)Modelling of Parallel Downloading problem as a directed graph. 2)Proposition of a smart connection management algorithm	1)Incentive compatibility and cheat-proofing in our group-based mechanism and its ability to handle malicious users.

CHAPTER NUMBER-3

SYSTEM DESIGN

For the proposed system, we shall be using the following softwares and platforms for development purpose:

- Platform – Net Beans IDE
- An open source development environment
- Programming language - JAVA.
- We need the Java Development Kit with the Java compiler

There are several sources for a JRE/JDK. Here are some of the more common/popular ones (listed alphabetically):

- IBM JDK
- Open JDK
- Oracle JDK
- We are using ORACLE JDK:
- http://www.oracle.com/technetwork/articles/java/jdk-7-NetBeans-download-432126.html

NetBeans:

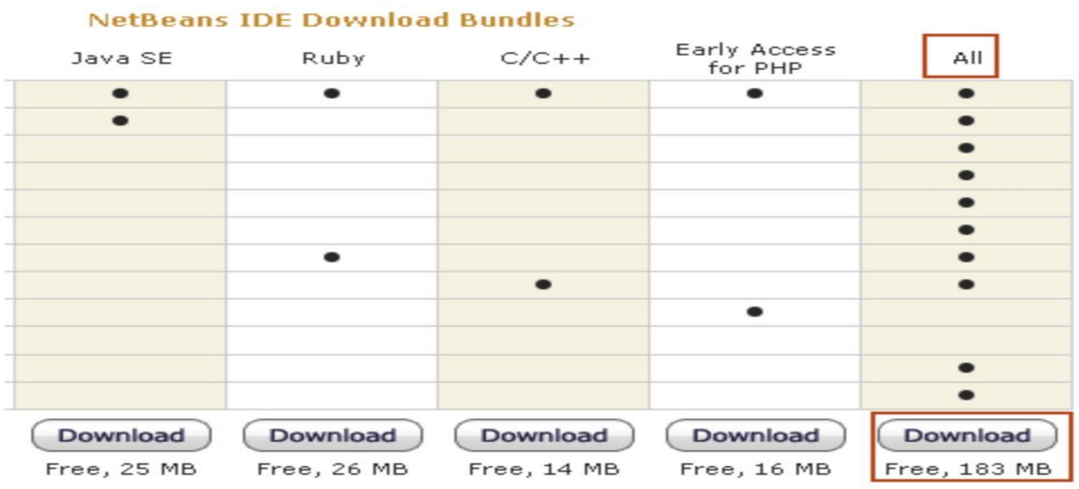
- NetBeans is an open-source integrated development environment (IDE) for developing with java, **php**, c++, and other programming languages.
- NetBeans is also referred to as a platform of modular components used for developing java desktop **applications**.
- NetBeans aims to help programmers develop software.
- NetBeans contains both a compiler and an interpreter, and switches between them according to need.
- NetBeans supports many languages and comes with a variety of plugins:
- https://NetBeans.org/downloads/index.html

Installation of NetBeans:

- Install JDK to use NetBeans for Java programming, you need to first install Java Development Kit (JDK).
- Download. Download "**NetBeans** IDE" installer from http://**NetBeans**.org/downloads/index.html
- Run the Installer. Run the downloaded installer

STEPS PACKAGE INSTALLATION:

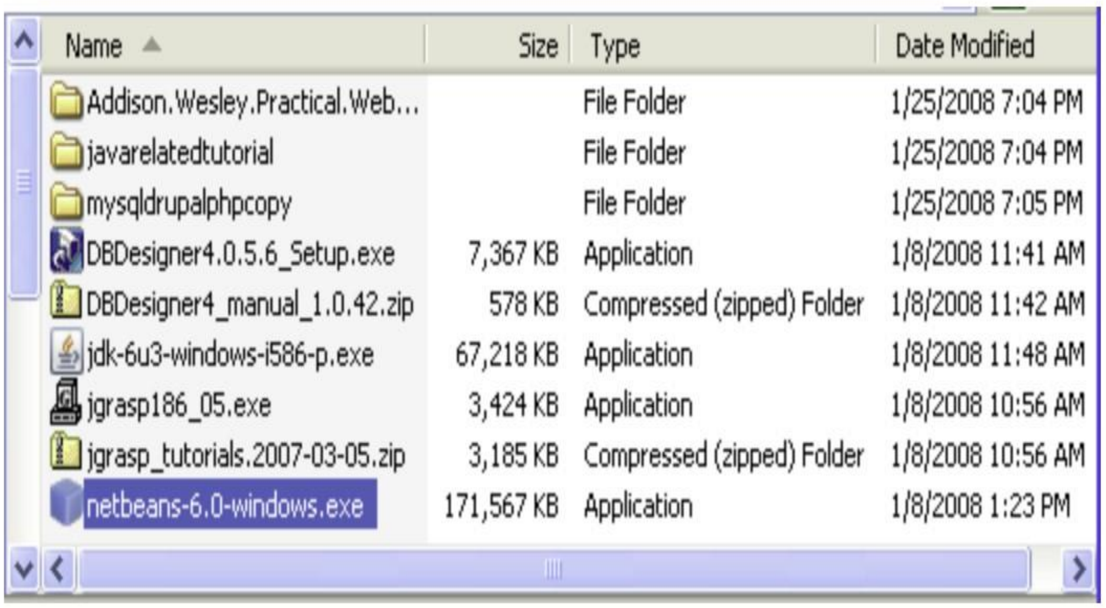
■ **As usual, go to NetBeans.org to download the latest version of NetBeans. We downloaded the full version that is the full bundled 'package'.**



NetBeans IDE Download Bundles

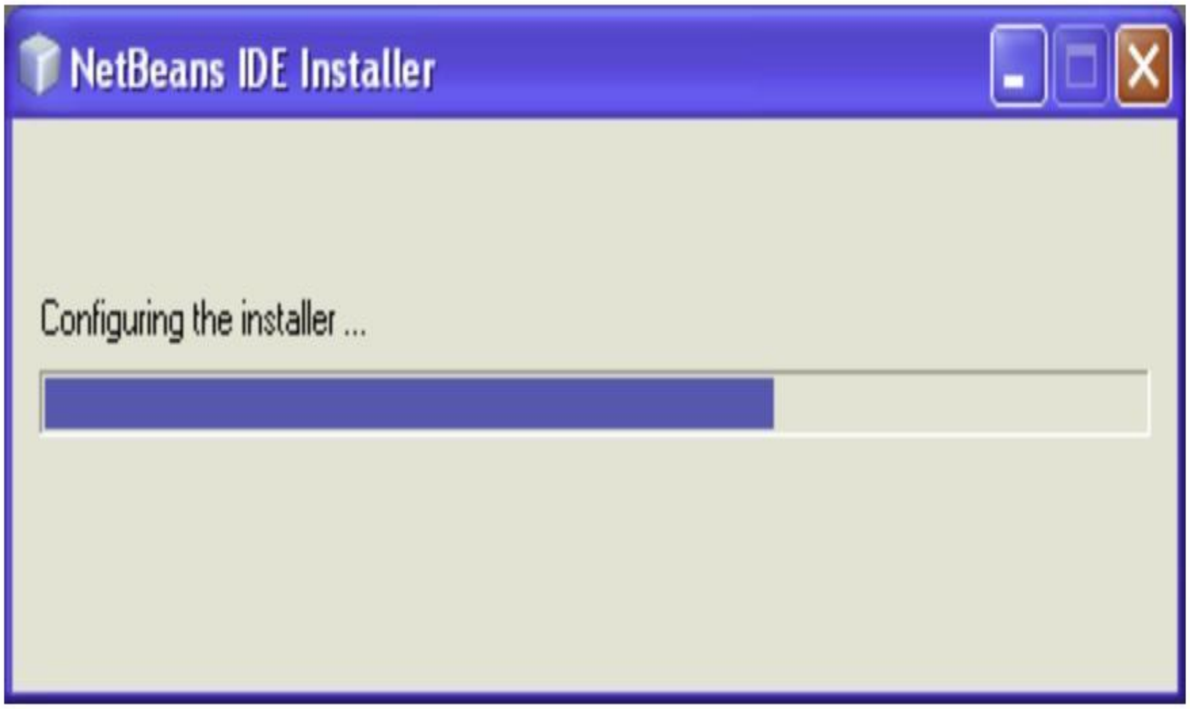
Java SE	Ruby	C/C++	Early Access for PHP	All
●	●	●	●	●
●				●
				●
				●
				●
				●
	●			●
		●		●
			●	
				●
				●
Download	Download	Download	Download	Download
Free, 25 MB	Free, 26 MB	Free, 14 MB	Free, 16 MB	Free, 183 MB

■ **Double click the Windows exe self-extracting file.**



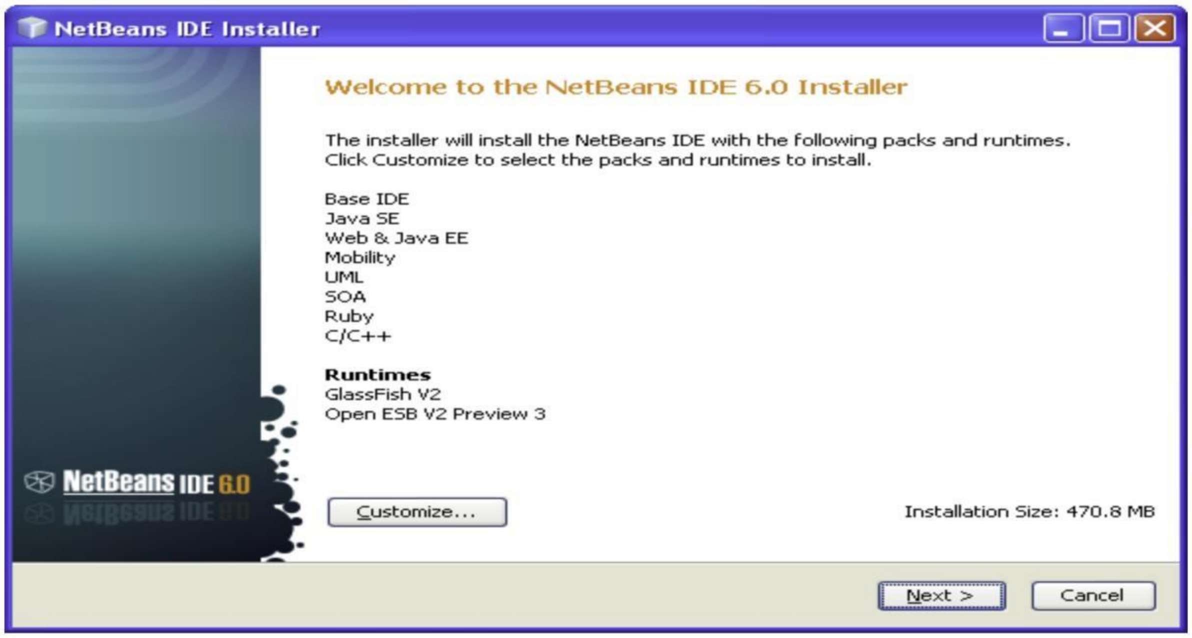
Name	Size	Type	Date Modified
Addison.Wesley.Practical.Web...		File Folder	1/25/2008 7:04 PM
javarelatedtutorial		File Folder	1/25/2008 7:04 PM
mysqldrupalphpcopy		File Folder	1/25/2008 7:05 PM
DBDesigner4.0.5.6_Setup.exe	7,367 KB	Application	1/8/2008 11:41 AM
DBDesigner4_manual_1.0.42.zip	578 KB	Compressed (zipped) Folder	1/8/2008 11:42 AM
jdk-6u3-windows-i586-p.exe	67,218 KB	Application	1/8/2008 11:48 AM
jgrasp186_05.exe	3,424 KB	Application	1/8/2008 10:56 AM
jgrasp_tutorials.2007-03-05.zip	3,185 KB	Compressed (zipped) Folder	1/8/2008 10:56 AM
netbeans-6.0-windows.exe	171,567 KB	Application	1/8/2008 1:23 PM

- **The NetBeans IDE installer will be launched.**


NetBeans IDE Installer

Configuring the installer ...

- **Then the NetBeans IDE installer wizard welcome page displayed. A list of default packs and runtimes shown in this welcome page. Click the Customize button.**


NetBeans IDE Installer

Welcome to the NetBeans IDE 6.0 Installer

The installer will install the NetBeans IDE with the following packs and runtimes.
Click Customize to select the packs and runtimes to install.

Base IDE
Java SE
Web & Java EE
Mobility
UML
SOA
Ruby
C/C++

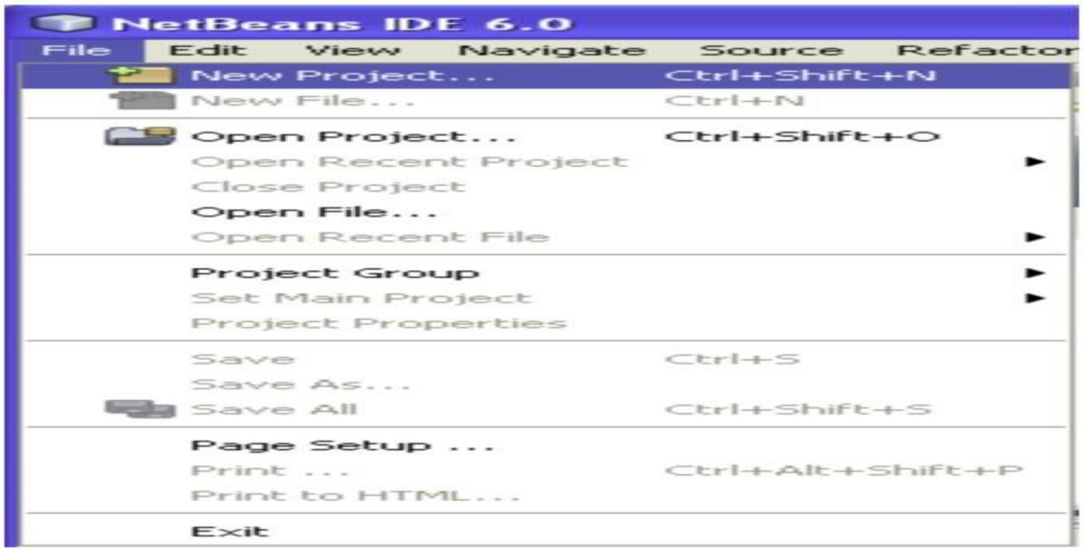
Runtimes
GlassFish V2
Open ESB V2 Preview 3

Installation Size: 470.8 MB

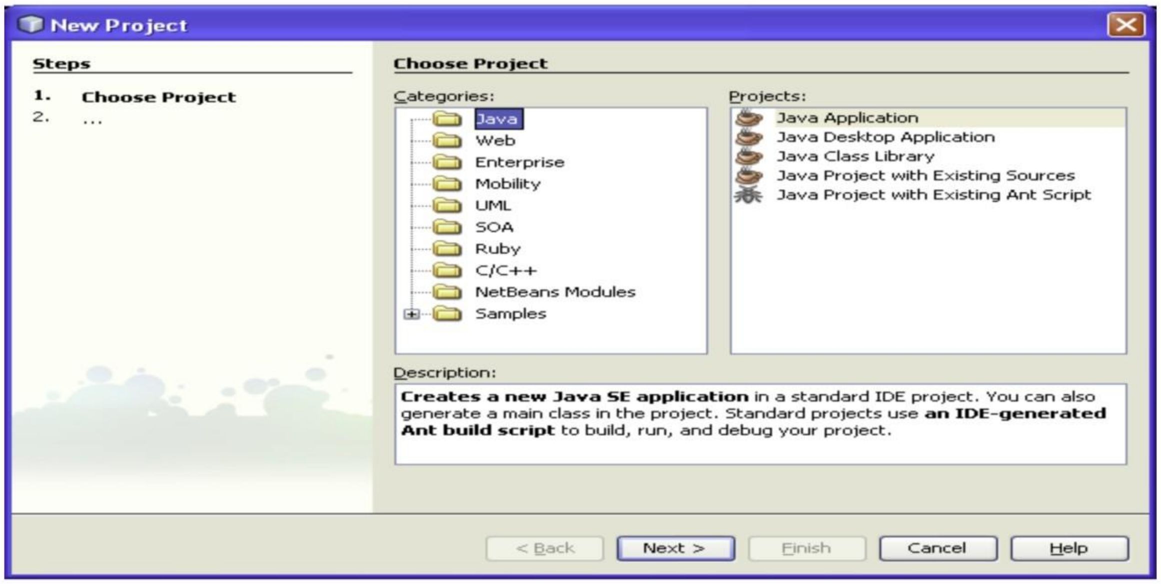
- Then we will choose path or workspace for the software ,where it will save.
- Then we will choose path or workspace for the software ,where it will save
- Next is the NetBeans IDE installation summary. If you want to change the previously selected settings, click Back otherwise click Install.

Creating and Running a program:

- Start NetBeans.
- Create a new Java Project:

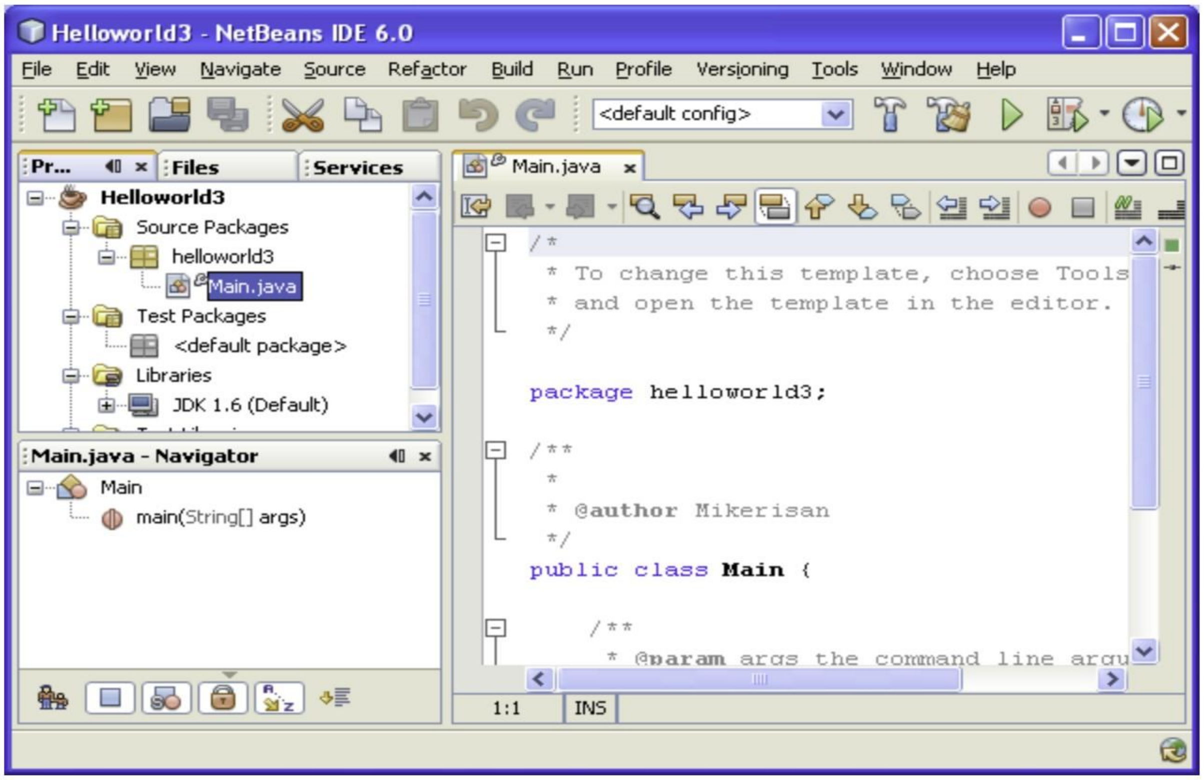


■ Select from the menu File --> New --> Java Project.



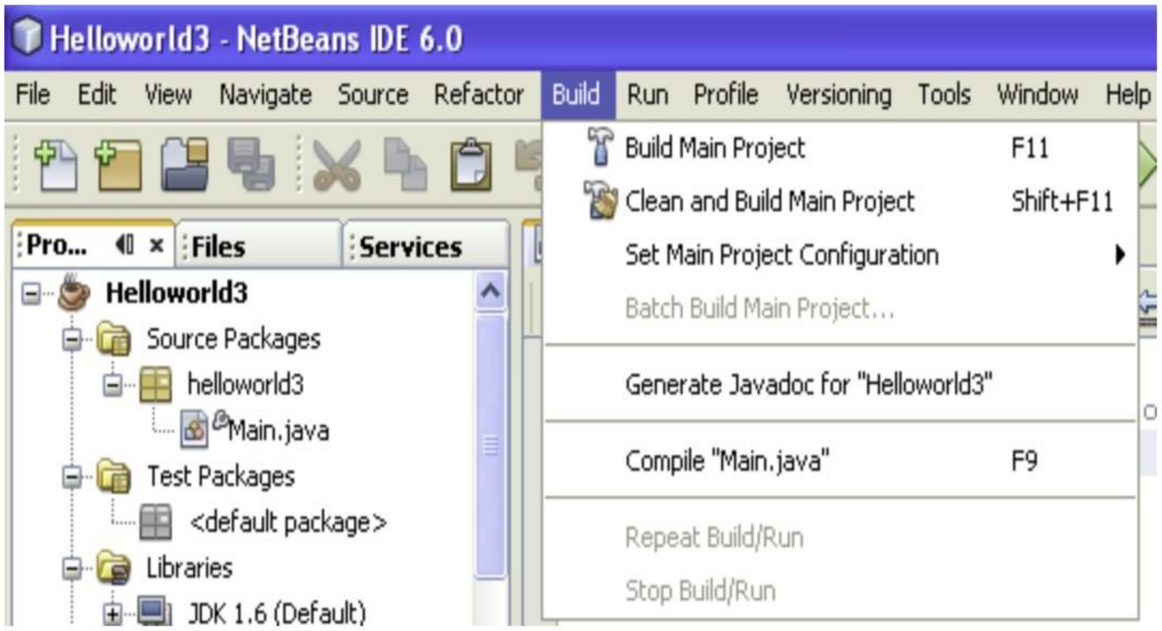
Creates a new Java SE application in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

■ Hello world program:



```
/*
 * To change this template, choose Tools
 * and open the template in the editor.
 */

package helloworld3;

/**
 *
 * @author Mikerisan
 */
public class Main {

    /**
     * @param args the command line argu
```

■ Build the Java program through Build menu. Click Build Main Project.


Helloworld3 - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
Build Main Project F11
Clean and Build Main Project Shift+F11
Set Main Project Configuration
Batch Build Main Project...
Generate Javadoc for "Helloworld3"
Compile "Main.java" F9
Repeat Build/Run
Stop Build/Run
Pro... Files Services
Helloworld3
Source Packages
helloworld3
Main.java
Test Packages
<default package>
Libraries
JDK 1.6 (Default)

CHAPTER NUMBER-4

SYSTEM IMPLEMENTATION

4.1 Algorithm and Analysis

- A thread is created for **IDM** which creates a thread for **Download manager**.
- Another thread is created for **progresspanel** which monitors the progress of the download.
- When download is complete, a new action is executed in the main thread which stops the 2 running threads and executes the **downloadcomplete** thread.
- **Downloadcomplete** class is executed with the help of a thread which displays the final download complete screen which displays the downloaded file.
- We have followed Object-Oriented approach to build this project in JAVA. We have divided our tasks into sub-modules, each module performing some specific functionality. These modules work together and download a file
- By splitting the download process into multiple blocks and assigning a thread to each block which opens up a connection with the server and finally when all threads are finished, all threads are joined to get the final downloaded file.

4.2 Module Description

I) **Downloader.java**

File handles following tasks

a) Creates map(a type of list) of threads.

b) declares current state of the download as one of the following: PAUSED, DOWNLOADING or CANCELED.

c) Taking the url and type conversion to string, fetching the size of the file.

d) Calculating the progress (%) using the formula (Downloaded / FileSize) * 100. e) Set the state of the download.

f) Start or resume the download.

g) Increase the downloaded size.

h) Set the state has changed and notify the observers.

i) Create thread to download a part of file.

j) See whether the thread is finished or not, wait for the thread to finish.

ii) IDM.java

File handles following tasks

a) Get and set the maximum number of connections possible per download(This isessentially the service provided by the downloading website).

b) Get the downloader object in a list(the object from previous class).

c) Get the download list.

d) Verify if the URL is valid.

iii)Progresspanel.java

File handles following tasks:

a) Create an instance of DownloadTableModel.

b) Set up the table.

c) Allow only one row at a time to be selected.

d) Set up progress bar as progress renderer for progress column.

e) Set table's row height large enough to fit progress bar.

f) Create button, text fields, scrollpane and table.

g) Set the text for title bar, for jbnAdd button set the text to "Add Download", for jbnPausebutton set the text to "Pause", for jbnRemove button set the text to "Remove", for jbnRemove button set the text to "Remove", for jbnResume button set the text to "Resume", for jbnExit button set the text to "Exit".

h) Create a layout and add these components to the layout specifying the dimensions of every component.

i) Assign the function to the button.

j) If buttons are pressed then change the state of the button.

iv) DownloadCompleteFrame.java

This class manages the download table data:

a) Initializes various components of the progress check function

b) Reflects the progress of download by showing percentage of file downloaded.

c) Invokes startThread(IDM) to create and stop new thread for download.

d) JSwing is used to develop the GUI and various functions like JPanel,JFrame,JButton are used for development.

4.3 Areas of Parallelism

The download manager problem is divided broadly into sections of

- Checking the progress of download
- Executing the download
- Giving feedback when the download is complete

Separate threads have been allocated for these actions and hence they have been executed parallelly.

4.4 Use of Multicores in Java Virtual Machine

The standard JVM runs parallel Java threads in parallel, runs multiple threads for Garbage Collection and also the JIT compiler, and runs a number of other java-level but not user threads for a bunch of housekeeping tasks. The JVM is well tested and proven on hundreds of cores with thousands of runnable threads.

The JVM can be "pinched" down to a specific number of cores by any OS which can limit CPUs (e.g. linux cgroups), as long as the OS also multi-tasks the runnable threads on the available cores.

The Azul/Zing JVM runs well on ~1000 cores and ~100K runnable threads (so 10x more cores and 100x more runnable threads).

The Java Runnable interface is designed for handling parallel and concurrent programs. Any class that implements this interface must implement this function:

```
@Override

public void run() {

  //implementation here

}
```

CHAPTER NUMBER 5

SOURCE CODE FOR EACH MODULE

5.1 IDM.java

```
package IDMPackage;

import java.io.*;       //Input Output data streams import java.net.URL; import

java.net.HttpURLConnection; import java.nio.charset.MalformedInputException; import

java.text.DecimalFormat;     //Use to format numbers using a formatting pattern specified by us

import java.util.Date;              //Used to display date and time  import java.util.logging.Level;

//recording application activity - warning,exception information import java.util.logging.Logger;

//recording application activity - warning,exception information public class IDM implements

Runnable {//instance intended to be executed by thread private String urlName; private String

OutputFileName;     int threadNumber;     private URL url;     private HttpURLConnection con;

private BufferedInputStream input;     public BufferedOutputStream output;     private byte[] buffer;

int downloaded;     int totalLength;     int startPos = 0;     int endPos = 0;     int pos = 0;     public

boolean threadState;


   public static int progress = 0;

ProgressPanel panel;

   int b;

   Date date;     int bread = 0;     int bytesPerSec = 0;     public IDM(String url,

ProgressPanel panel, String directry) {         date = new Date();        //b =

date.getSeconds();         threadState = true;         this.urlName = url;

this.OutputFileName = directry + "\\" + url.substring(url.lastIndexOf('/') + 1);

       try {

           System.out.println("full directory with file name: " + OutputFileName);

this.url = new URL(urlName);             this.con = (HttpURLConnection)

this.url.openConnection();             input = new

BufferedInputStream(con.getInputStream());             totalLength =

con.getContentLength();            downloaded = 0;            output = new
```

```
BufferedOutputStream(                    new
FileOutputStream(OutputFileName));            buffer = new byte[2 * 1024];


            this.panel = panel;
        } catch (MalformedInputException malformedInputException) {
malformedInputException.printStackTrace();         } catch
(IOException ioException) {             ioException.printStackTrace();

        }
    }
    public void setState(boolean state) {
threadState = state;
    }
    public int getDownloaded() {
return downloaded;
    }
    @Override
public  void  run()  {
int bytesRead;       try
{
            while ((bytesRead = this.input.read(buffer)) != -1) {
               endPos = bytesRead;  new
               IDM.ReadThreadClass(buffer, 0, bytesRead);
               System.out.println("Pos: " + startPos + ":" + endPos);
               startPos = endPos;  this.output.flush();
            }
         this.output.flush();
         this.input.close();
         this.output.close();
        } catch (MalformedInputException malformedInputException) {
         malformedInputException.printStackTrace();
        } catch (IOException ioException) {
         ioException.printStackTrace();
        }
    }
    ProgressPanel oo = new ProgressPanel();
class ReadThreadClass implements Runnable {
        private int startPs, endPs;         byte[] buffewr;         public
ReadThreadClass(byte[] buffer, int startPos, int endPos) {
startPs = startPos;             endPs = endPos;             buffewr = buffer;
run();
        }
        public ReadThreadClass(int bytesRead) {
endPs = bytesRead;
        }
        boolean a = false;
public void run() {
            try {                 output.write(buffer, 0, endPs);                 downloaded
+= endPs;                 double downloadedd = ((double) downloaded /
totalLength) * 100;                 progress = (int) downloadedd;
panel.setProgressValue(progress);
                  System.out.println("progress: " + progress);                 double percentage =
Double.parseDouble(new DecimalFormat(".##").format(downloadedd));
System.out.println("downloaded: " + downloaded + " : " + (percentage) + "%");
panel.setTextToArea(totalLength, percentage);
            } catch (IOException ex) {
                Logger.getLogger(IDM.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }
}
```

5.2 Progresspanel.java

```
package IDMPackage; public class ProgressPanel

extends javax.swing.JPanel {

   Thread thread;     String

outputDirectory;     private

String outputFile;

   /**

    * Creates new form ProgressPanel

    */

   public ProgressPanel() {

initComponents();

   }

   public void setProgressValue(int value) {

progressBar.setValue(value);         if

(progressBar.getPercentComplete() == 1.0) {

          System.out.println("FINISHED");

 DownloadCompleteFrame dcf = new DownloadCompleteFrame(outputDirectory, urlTextField.getText(), outputFile);

       downloader.downloaderFrame.repaint();    downloader.downloaderFrame.setVisible(true);

      }

   }

public void setUrlTextField(String url) {

urlTextField.setText(url);

   }

   public void startThread(IDM idm) {

       thread = new Thread(idm);

       thread.start();

   }

public void stopThread() {

       this.thread.stop();

   }
```

```
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
progressBar = new javax.swing.JProgressBar();
pauseButton = new javax.swing.JButton();
resumeButton = new javax.swing.JButton();
cancelButton = new javax.swing.JButton();
jLabel3 = new javax.swing.JLabel();
urlTextField = new javax.swing.JTextField();
jSeparator1 = new javax.swing.JSeparator();
jSeparator2 = new javax.swing.JSeparator();
jScrollPane1 = new javax.swing.JScrollPane();
jTextArea1 = new javax.swing.JTextArea();

        setMaximumSize(new java.awt.Dimension(800, 300));
setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel1.setText("Progress");         add(jLabel1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 49, -1, 23));
progressBar.setForeground(new java.awt.Color(102, 255, 102));
progressBar.setStringPainted(true);         add(progressBar, new
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 49, 410, 23));

        pauseButton.setText("Pause");
pauseButton.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
pauseButtonActionPerformed(evt);
            }
        });
        add(pauseButton, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 90, -1, -1));

        resumeButton.setText("Resume");
resumeButton.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
resumeButtonActionPerformed(evt);
            }
        });
        add(resumeButton, new org.netbeans.lib.awtextra.AbsoluteConstraints(260, 90, -1, -1));

        cancelButton.setText("Cancel");
cancelButton.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
cancelButtonActionPerformed(evt);
            }
        });
        add(cancelButton, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 90, -1, -1)); jLabel3.setText("URL              ");
add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 11, 42, 27));

        urlTextField.setEditable(false);
urlTextField.setMaximumSize(new java.awt.Dimension(100, 20));
urlTextField.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
urlTextFieldActionPerformed(evt);
            }
        });
        add(urlTextField, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 11, 410, 27));
add(jSeparator1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 122, 500, 20));
add(jSeparator2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 230, 500, 0));
```

```
jTextArea1.setColumns(20);        jTextArea1.setRows(5);

jScrollPane1.setViewportView(jTextArea1);       add(jScrollPane1, new

org.netbeans.lib.awtextra.AbsoluteConstraints(70, 130, 400, 90));

    }// </editor-fold>

    private void urlTextFieldActionPerformed(java.awt.event.ActionEvent evt) {

    }

    private void pauseButtonActionPerformed(java.awt.event.ActionEvent evt) {

  //Thread.interrupted();

this.thread.suspend();

    }

    private void resumeButtonActionPerformed(java.awt.event.ActionEvent evt) {

this.thread.resume();

    }

    private void cancelButtonActionPerformed(java.awt.event.ActionEvent evt) {                                        stopThread();

downloader.downloaderFrame.remove(this);                                        downloader.downloaderFrame.repaint();

downloader.downloaderFrame.setVisible(true);

    }

    // Variables declaration - do not modify

private javax.swing.JButton cancelButton;

private javax.swing.JLabel jLabel1;     private

javax.swing.JLabel jLabel3;     private

javax.swing.JScrollPane jScrollPane1;     private

javax.swing.JSeparator jSeparator1;     private

javax.swing.JSeparator jSeparator2;     private

javax.swing.JTextArea jTextArea1;     private

javax.swing.JButton pauseButton;     private

javax.swing.JProgressBar progressBar;     private

javax.swing.JButton resumeButton;     private

javax.swing.JTextField urlTextField;
```

```
    // End of variables declaration         private javax.swing.JTextArea jTextArea;     void

setTextToArea(int b, double c) {         jTextArea1.append("Total Size :" + (float) b / 1024 + " kbs " +

" : percentage:" + c + "%" + "\n");

    }

    void setOutputDirectory(String string, String file) {

outputDirectory = string;         outputFile = file;

    }

}
```

5.3 Downloader.java

```
package IDMPackage; import java.awt.event.ActionEvent; //A semantic event which indicates that a

component-defined action occured

import java.awt.event.ActionListener; //The event is passed to every every ActionListener object that registered to receive such
events using the component's addActionListener method.

import java.io.File; //this statement means all the classes of io package will be imported. used when we are using input/output
stream.

import javax.swing.*; //Java Swing is a lightweight Graphical User Interface (GUI) toolkit

public class downloader extends javax.swing.JFrame implements ActionListener {

static void downloadComplete() {

    }

    IDM idm;

    Thread thread;

    ProgressPanel progressPanel;     JLabel

label;     public static JFrame

downloaderFrame;

    /**

     * Creates new form downloader

     */
    public downloader() {
initComponents();
    }
    /**
  private void initComponents() {         jLabel1 = new javax.swing.JLabel();
urlTextField = new javax.swing.JTextField();         downloadButton = new
javax.swing.JButton();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setBounds(new java.awt.Rectangle(0, 0, 800, 800));

        setMaximumSize(new java.awt.Dimension(800, 600));
getContentPane().setLayout(new java.awt.FlowLayout());
jLabel1.setText("URL     ");         getContentPane().add(jLabel1);
urlTextField.setColumns(50);         urlTextField.addActionListener(new
java.awt.event.ActionListener() {             public void
actionPerformed(java.awt.event.ActionEvent evt) {
urlTextFieldActionPerformed(evt);
            }
        });
        getContentPane().add(urlTextField);         downloadButton.setText("Save
To");         downloadButton.addActionListener(new
java.awt.event.ActionListener() {             public void
actionPerformed(java.awt.event.ActionEvent evt) {
downloadButtonActionPerformed(evt);
            }
        });
        getContentPane().add(downloadButton);
setBounds(0, 0, 706, 323);
    }// </editor-fold>

    private void downloadButtonActionPerformed(java.awt.event.ActionEvent evt) {
```

```
//        System.setProperty("http.proxyHost", "192.168.10.50");

//        System.setProperty("http.proxyPort", "8080");        JFileChooser

fileChooser = new JFileChooser();

fileChooser.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);

int result = fileChooser.showSaveDialog(this);

        File file = fileChooser.getSelectedFile();

        System.out.println("directry: " + file.getPath());

progressPanel = new ProgressPanel();

        try {

            idm = new IDM(urlTextField.getText(), progressPanel, file.getPath());

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

         downloaderFrame.add(progressPanel);

         progressPanel.startThread(idm);

         progressPanel.setUrlTextField(urlTextField.getText());

              progressPanel.setOutputDirectory(file.getPath(), urlTextField.getText().substring(urlTextField.getText().lastIndexOf('/') +
1));    urlTextField.setText("");

         downloaderFrame.repaint();

         downloaderFrame.setVisible(true);

        } catch (Exception e) {

        }

}

    private void urlTextFieldActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

    }



    try {

            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels())

               {  if ("Nimbus".equals(info.getName())) {

               javax.swing.UIManager.setLookAndFeel(info.getClassName());

                    break;

                }
            }
        } catch (ClassNotFoundException ex) {
         java.util.logging.Logger.getLogger(downloader.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(downloader.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(downloader.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(downloader.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
   java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {                 downloaderFrame = new
downloader();//.setVisible(true);
                //
                downloaderFrame.setVisible(true);
                //JScrollPane scrollPane = new JScrollPane(downloaderFrame);
            }
        });
    }
    // Variables declaration - do not modify
private javax.swing.JButton downloadButton;
private javax.swing.JLabel jLabel1;     private
javax.swing.JTextField urlTextField;
    // End of variables declaration
    @Override     public void
actionPerformed(ActionEvent e) {
    }
}
```

5.4 DownloadCompleteFrame.java

```
package IDMPackage; import java.awt.Desktop; import java.io.File;

import java.io.IOException; import java.util.logging.Level; import

java.util.logging.Logger; public class DownloadCompleteFrame

extends javax.swing.JFrame {     public DownloadCompleteFrame()

{         initComponents();

            }

    public DownloadCompleteFrame(String outputDirectory,String url,String outputFile) {

initComponents();         this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);

        this.jLabel1.setText(url);

this.jLabel3.setText(outputFile);

this.jLabel5.setText(outputDirectory);

this.setVisible(true);

    }

    private void initComponents() {         jLabel1

= new javax.swing.JLabel();         jLabel2 = new

javax.swing.JLabel();         jLabel3 = new

javax.swing.JLabel();         jLabel4 = new

javax.swing.JLabel();         jLabel5 = new

javax.swing.JLabel();         jLabel6 = new

javax.swing.JLabel();         jSeparator1 = new

javax.swing.JSeparator();         jButton1 = new

javax.swing.JButton(); jButton2 = new

javax.swing.JButton();         jButton3 = new

javax.swing.JButton();

setDefaultCloseOperation(javax.swing.Window

Constants.EXIT_ON_CLOSE);

setTitle("Download Complete");

jLabel1.setName("UrlLabel");
```

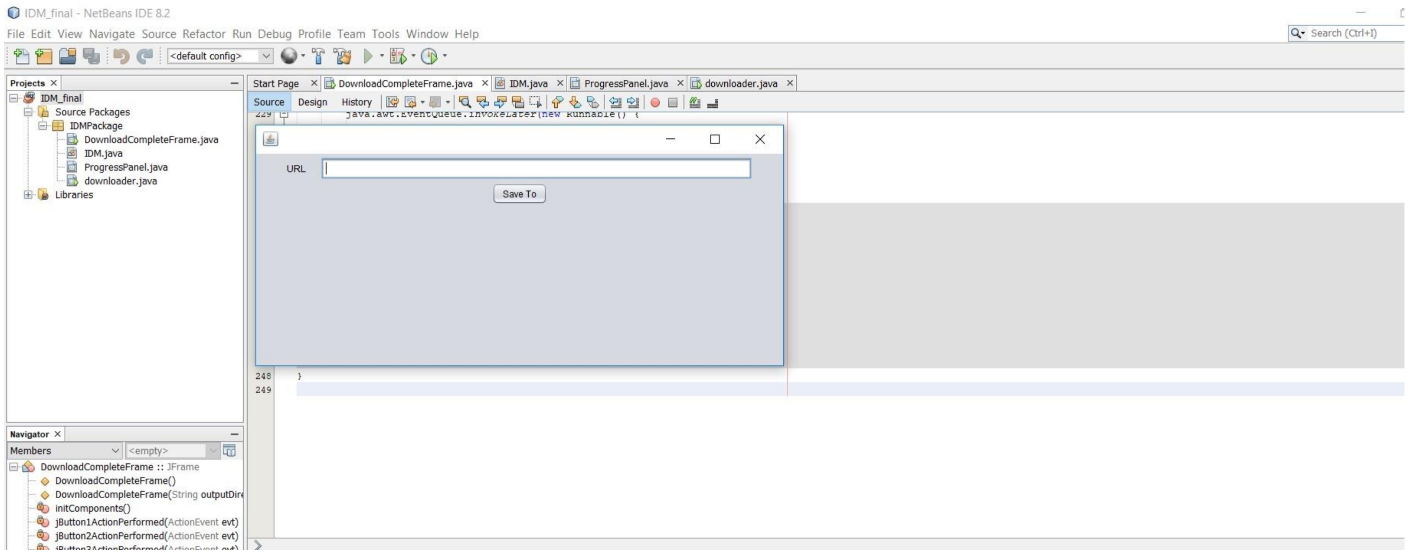
```
jLabel2.setText("URL");

jLabel3.setName("fileNameLabel");

jLabel4.setText("File Name");

jLabel5.setName("fileDirectory");

jLabel6.setText("File Directory");

jButton1.setText("Open Folder");

jButton1.addActionListener(new

java.awt.event.ActionListener() {             public

void

actionPerformed(java.awt.event.ActionEvent

evt) {                 jButton1ActionPerformed(evt);

            }

        });

        jButton2.setText("Close");         jButton2.addActionListener(new

java.awt.event.ActionListener() {             public void

actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

            }

        });

        jButton3.setText("Open");         jButton3.addActionListener(new

java.awt.event.ActionListener() {             public void

actionPerformed(java.awt.event.ActionEvent evt) {

jButton3ActionPerformed(evt);

            }

        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);         layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

                    .addComponent(jSeparator1,javax.swing.GroupLayout.PREFERRED_SIZE,662,
javax.swing.GroupLayout.PREFERRED_SIZE)

                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                        .addGroup(layout.createSequentialGroup()

                            .addComponent(jLabel2,javax.swing.GroupLayout.PREFERRED_SIZE,94,
javax.swing.GroupLayout.PREFERRED_SIZE)

                            .addGap(64, 64, 64)

                            .addComponent(jLabel1,javax.swing.GroupLayout.PREFERRED_SIZE,463,
javax.swing.GroupLayout.PREFERRED_SIZE))

                        .addGroup(layout.createSequentialGroup()

                            .addComponent(jLabel4,javax.swing.GroupLayout.PREFERRED_SIZE,94,
javax.swing.GroupLayout.PREFERRED_SIZE)

                            .addGap(64, 64, 64)

                            .addComponent(jLabel3,javax.swing.GroupLayout.PREFERRED_SIZE,463,
javax.swing.GroupLayout.PREFERRED_SIZE))

                        .addGroup(layout.createSequentialGroup()

                            .addComponent(jLabel6,javax.swing.GroupLayout.PREFERRED_SIZE,94,
javax.swing.GroupLayout.PREFERRED_SIZE)

                            .addGap(53, 53, 53)

                            .addComponent(jLabel5,javax.swing.GroupLayout.PREFERRED_SIZE,463,
javax.swing.GroupLayout.PREFERRED_SIZE))))

                .addGap(41, 41, 41))

            .addGroup(layout.createSequentialGroup()

                .addGap(66, 66, 66)

                .addComponent(jButton3,javax.swing.GroupLayout.PREFERRED_SIZE,124,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(64, 64, 64)

                .addComponent(jButton1,javax.swing.GroupLayout.PREFERRED_SIZE,124,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(69, 69, 69)

                .addComponent(jButton2,javax.swing.GroupLayout.PREFERRED_SIZE,124,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addContainerGap())

        );
        layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(42, 42, 42)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel2,javax.swing.GroupLayout.PREFERRED_SIZE,21,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel1,javax.swing.GroupLayout.PREFERRED_SIZE,21,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addComponent(jLabel4, javax.swing.GroupLayout.DEFAULT_SIZE, 26, Short.MAX_VALUE)
                    .addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addComponent(jLabel6, javax.swing.GroupLayout.DEFAULT_SIZE, 26, Short.MAX_VALUE)
                    .addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
                .addGap(27, 27, 27)
                .addComponent(jSeparator1,javax.swing.GroupLayout.PREFERRED_SIZE,26,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jButton3)
                    .addComponent(jButton1)
                    .addComponent(jButton2))
                .addContainerGap(29, Short.MAX_VALUE))
        );
pack();
    }// </editor-fold>                              private void
jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            // TODO add your handling code here:
           //String dir =  jLabel5.getText()+"\"+jLabel3.getText();
            String x = jLabel5.getText();
String aa= "             char a[] =
x.toCharArray();             for(int
i=0;i<a.length;i++){
                if( '\\'== a[i])
                {
aa+="\\\\";
                }
else                 {
aa+=a[i];
                }
            }
          //  System.out.println(aa+"\\\\"+jLabel3.getText());
             String cmds[] = new String[] {"cmd", "/c", aa+"\\\\"+jLabel3.getText()
           };//C:\\Documents and Settings\\henry\\My Documents\\test.pdf"};
                Runtime.getRuntime().exec(cmds);
        } catch (IOException ex) {
            Logger.getLogger(DownloadCompleteFrame.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
        this.dispose();
    }
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
                Desktop desktop = Desktop.getDesktop(); try
{
   // desktop.
desktop.open(new File(this.jLabel5.getText()));
} catch (IOException e){ }
    }
try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
if ("Nimbus".equals(info.getName())) {
javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(DownloadCompleteFrame.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
        } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(DownloadCompleteFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(DownloadCompleteFrame.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(DownloadCompleteFrame.class.getName()).log(java.util.logging.Level.SEVERE,  null,
ex);        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
             //   new DownloadCompleteFrame().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JLabel jLabel1;     private
```

```
javax.swing.JLabel jLabel2;     private

javax.swing.JLabel jLabel3;     private

javax.swing.JLabel jLabel4;     private

javax.swing.JLabel jLabel5;     private

javax.swing.JLabel jLabel6;     private

javax.swing.JSeparator jSeparator1;

    // End of variables declaration

}
```

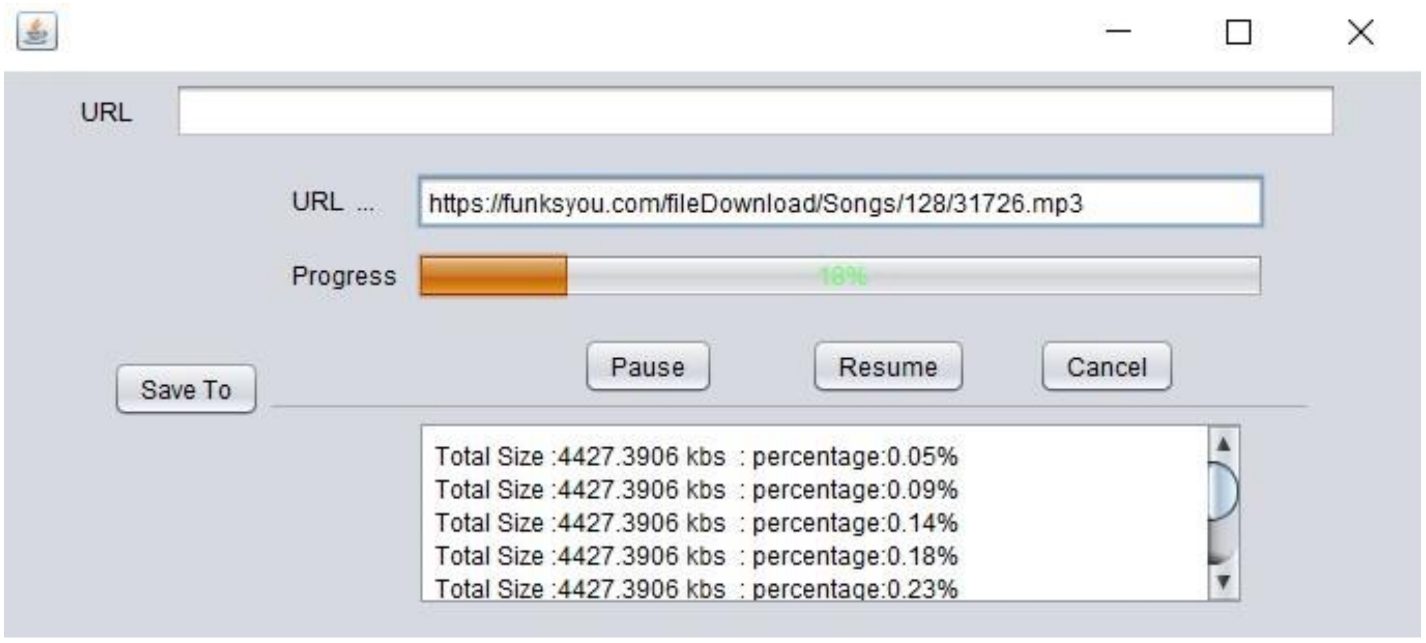
CHAPTER NUMBER-6

RESULTS AND DISCUSSIONS

6.1 Outputs


IDM_final - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
IDM_final
Source Packages
IDMPackage
DownloadCompleteFrame.java
IDM.java
ProgressPanel.java
downloader.java
Libraries
URL
Save To

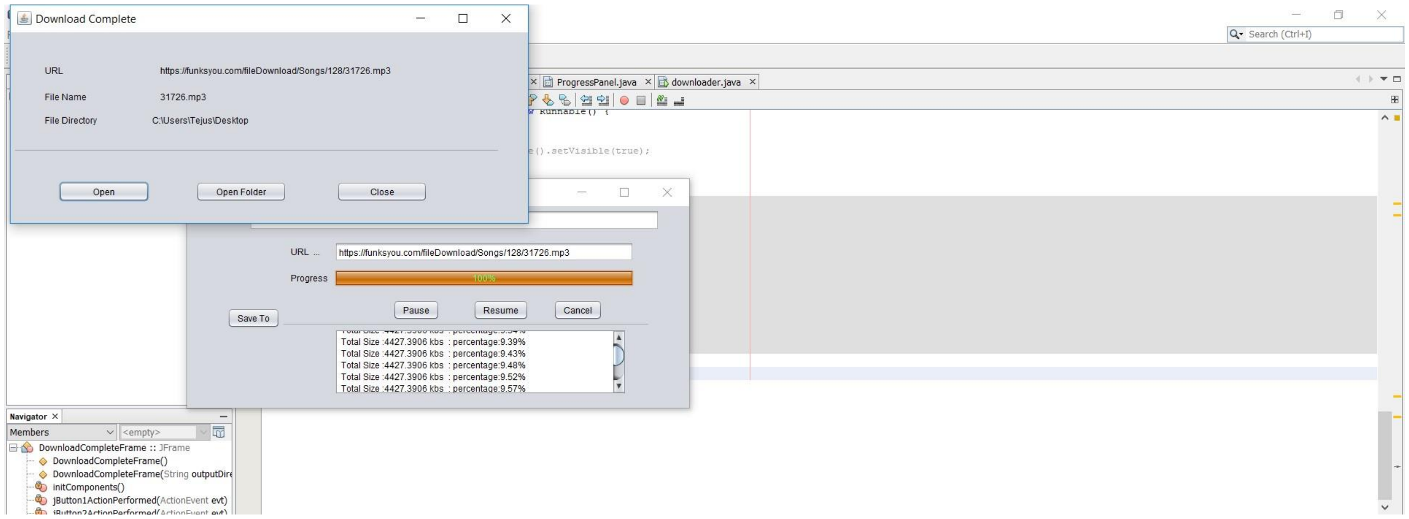
Upon adding URL and setting a location to save:



https://funksyou.com/fileDownload/Songs/128/31726.mp3

```
Total Size :4427.3906 kbs  : percentage:0.05%
Total Size :4427.3906 kbs  : percentage:0.09%
Total Size :4427.3906 kbs  : percentage:0.14%
Total Size :4427.3906 kbs  : percentage:0.18%
Total Size :4427.3906 kbs  : percentage:0.23%
```

Once Download is complete:



Download Complete

URL https://funksyou.com/fileDownload/Songs/128/31726.mp3

File Name 31726.mp3

File Directory C:\Users\Tejus\Desktop

URL ... https://funksyou.com/fileDownload/Songs/128/31726.mp3

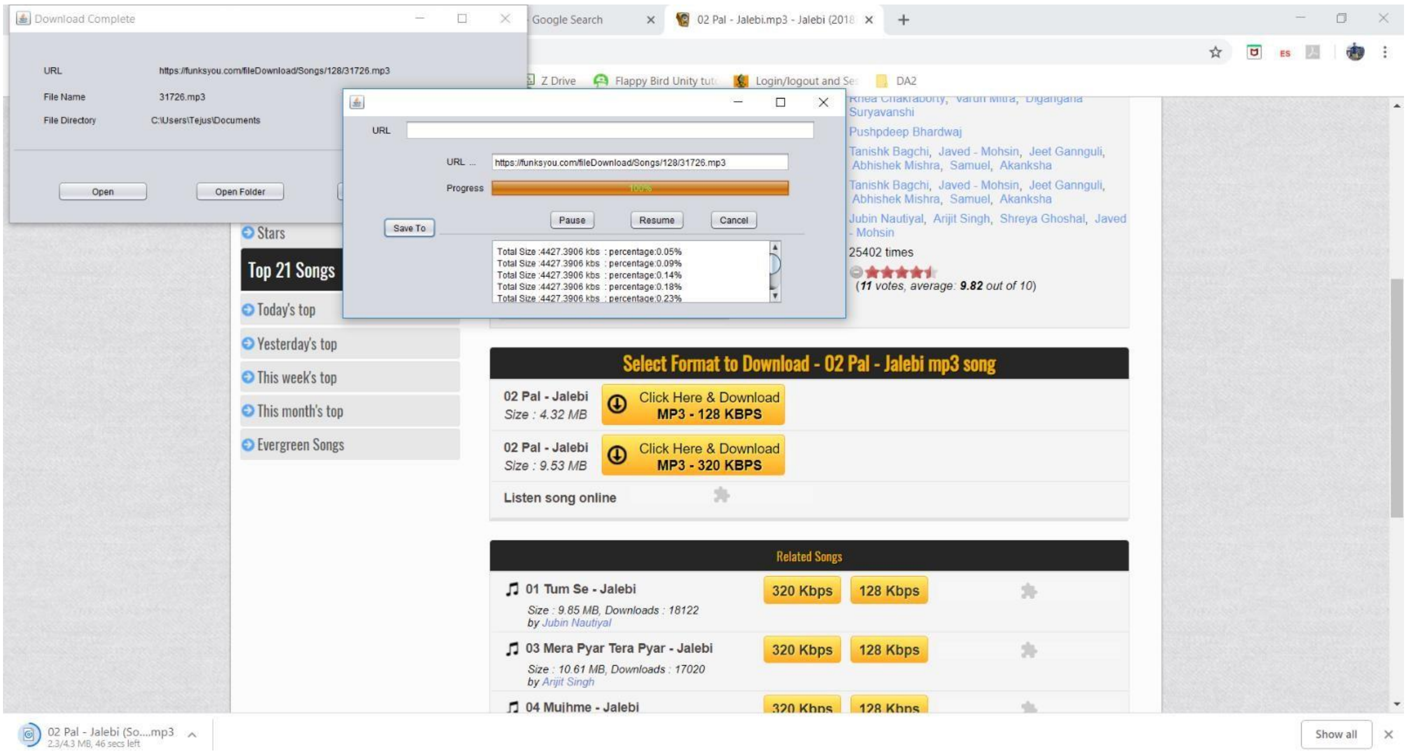
Progress 100%

```
Total Size :4427.3906 kbs  : percentage:9.39%
Total Size :4427.3906 kbs  : percentage:9.43%
Total Size :4427.3906 kbs  : percentage:9.48%
Total Size :4427.3906 kbs  : percentage:9.52%
Total Size :4427.3906 kbs  : percentage:9.57%
```

```
DownloadCompleteFrame :: JFrame
    DownloadCompleteFrame()
    DownloadCompleteFrame(String outputDire
    initComponents()
    jButton1ActionPerformed(ActionEvent evt)
```

COMPARISON WITH GOOGLE CHROME

When started together:



Download Complete

URL https://funksyou.com/fileDownload/Songs/128/31726.mp3

File Name 31726.mp3

File Directory C:\Users\Tejus\Documents

URL ... https://funksyou.com/fileDownload/Songs/128/31726.mp3

Progress 100%

```
Total Size :4427.3906 kbs  : percentage:0.05%
Total Size :4427.3906 kbs  : percentage:0.09%
Total Size :4427.3906 kbs  : percentage:0.14%
Total Size :4427.3906 kbs  : percentage:0.18%
Total Size :4427.3906 kbs  : percentage:0.23%
```

25402 times

(11 votes, average: 9.82 out of 10)

Select Format to Download - 02 Pal - Jalebi mp3 song

02 Pal - Jalebi Size : 4.32 MB	Click Here & Download MP3 - 128 KBPS
02 Pal - Jalebi Size : 9.53 MB	Click Here & Download MP3 - 320 KBPS
Listen song online	

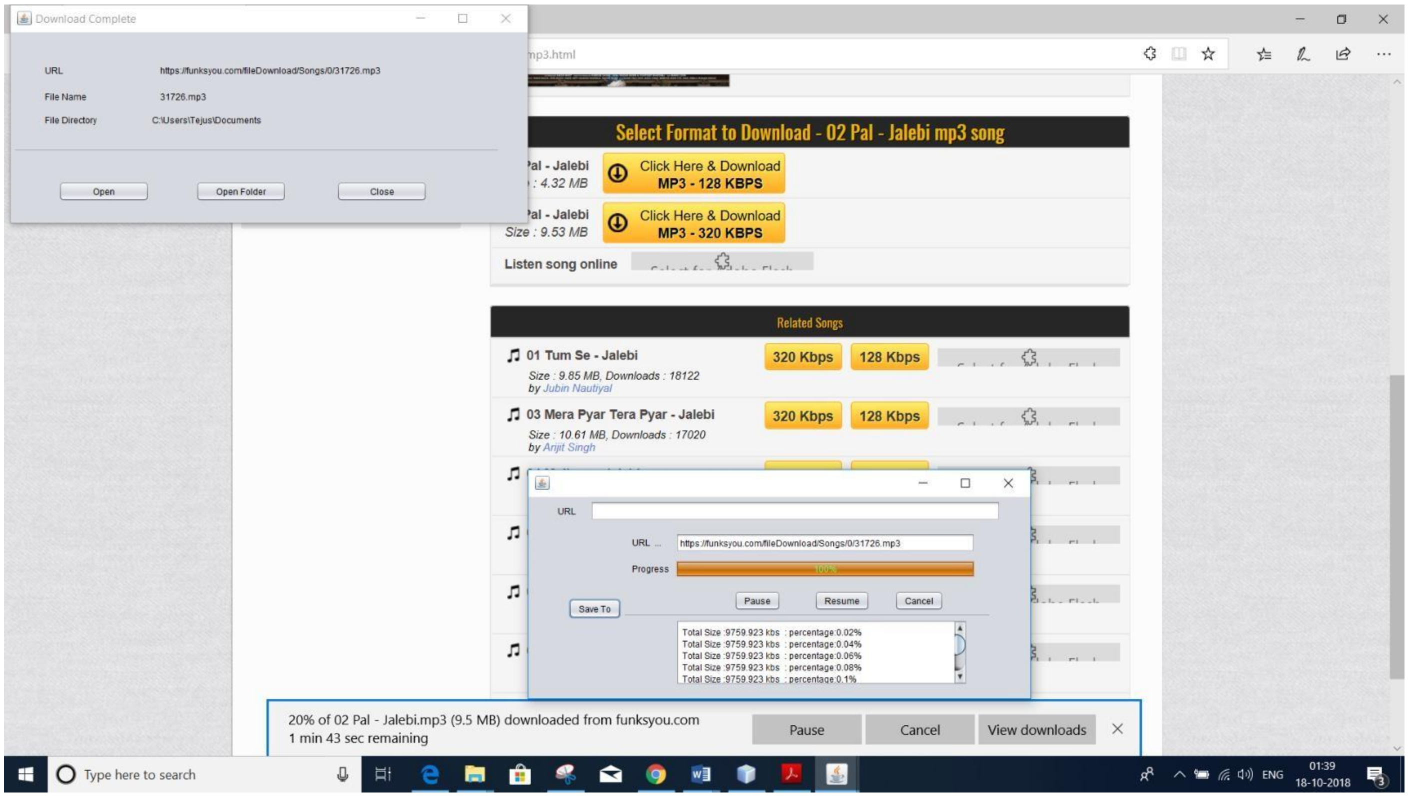
Related Songs

01 Tum Se - Jalebi Size : 9.85 MB, Downloads : 18122 by Jubin Nautiyal	320 Kbps	128 Kbps
03 Mera Pyar Tera Pyar - Jalebi Size : 10.61 MB, Downloads : 17020 by Arijit Singh	320 Kbps	128 Kbps
04 Mujhme - Jalebi	320 Kbps	128 Kbps

02 Pal - Jalebi (So....mp3
2.3/4.3 MB, 46 secs left

COMPARISON WITH MICROSOFT EDGE

When started together:


Download Complete

URL https://funksyou.com/fileDownload/Songs/0/31726.mp3

File Name 31726.mp3

File Directory C:\Users\Tejus\Documents

Open
Open Folder
Close
Select Format to Download - 02 Pal - Jalebi mp3 song
Click Here & Download MP3 - 128 KBPS
Click Here & Download MP3 - 320 KBPS
Listen song online
Related Songs
01 Tum Se - Jalebi
Size : 9.85 MB, Downloads : 18122
by Jubin Nautiyal
03 Mera Pyar Tera Pyar - Jalebi
Size : 10.61 MB, Downloads : 17020
by Arijit Singh
320 Kbps
128 Kbps

URL ... https://funksyou.com/fileDownload/Songs/0/31726.mp3

Progress 100%

Save To
Pause
Resume
Cancel

```
Total Size :9759.923 kbs  : percentage:0.02%
Total Size :9759.923 kbs  : percentage:0.04%
Total Size :9759.923 kbs  : percentage:0.06%
Total Size :9759.923 kbs  : percentage:0.08%
Total Size :9759.923 kbs  : percentage:0.1%
```

20% of 02 Pal - Jalebi.mp3 (9.5 MB) downloaded from funksyou.com
1 min 43 sec remaining

6.2 Discussion

The above Chart shows the speed up achieved on running the download using our multithreaded download manager in comparison to using Google Chrome and Microsoft Edge browser download managers.
It can be seen that a speedup can be achieved using our model. Also, in case of disconnectivity of internet, our downloader resumes the download from the point of dysconnectivity whereas other browsers cancel the download.

CHAPTER NUMBER-7

CONCLUSION

6.1 Conclusion

We have presented a parallel downloading method using multi threading in JAVA. The performance is better than serial downloading as the bandwidth is utilized more efficiently. We have compared the time taken for different download managers and came to a conclusion that multi-threaded downloading is faster. In future, we would like to add more features to the application in the future like download scheduling, bandwidth allocation etc.

6.2 References

[1] C. Gkantsidis, M. Ammar and E. Zegura. "Proceedings the Third IEEE Workshop on Internet Applications.
WIAPP 2003".
[2] S.G.M. Koo, C. Rosenberg and Dongyan Xu. "The Ninth IEEE Workshop on Future Trends of Distributed
Computing Systems, 2003. FTDCS 2003. Proceedings."
[3] Zhou Xu, Lu Xianliang, Hou Mengshu and Zhan Chaun. " ACM SIGOPS Operating Systems Review Homepage archive Volume 39 Issue 1, January 2005 Pages 63-69 "
[4] Jiantao Song, Chaofeng Sha and Hong Zhu. "Distributed Computing Systems, 2004. Proceedings."
[5] Allen Miu and Eugene Shih. "Laboratory of Computer Science Massachusetts Institute of Technology Cambridge, MA, USA"
[6] J. Funasaka, N. Nakawaki and K. Ishida. "Distributed Computing Systems Workshops, 2003. Proceedings."
[7] Atsushi Kawano, Junichi Funasaka and Kenji Ishida. "Distributed Computing Systems Workshops, 2007. ICDCSW '07."
[8] Ching-Hsien Hsu, Chia-Wei Chu and Chih-Hsun Chou. " 2009 IEEE International Symposium on Parallel
and Distributed Processing with Applications"
[9] Haitao Chen, Zhenghu Gong and Zunguo Huang. "Parallel and Distributed Computing Applications and Technologies (PDCAT'05)"
[10] Y. Higashi, S. Ata and I. Oka. "Second IEEE Consumer Communications and Networking Conference, 2005. CCNC. 2005"
[11] Jin Bo. "2012 International Conference on Computer Distributed Control and Intelligent Environmental Monitoring"
[12] G.Narasinga Rao. "International Journal of Computer Applications (0975 – 8887) Volume 3 – No.2, June
2010"