Architecture

User manual

Version 1.0

**Wuhan University**

**April, 2014**

INDEX

[I. Introduction 3](#_Toc384392405)

[1.1 What is Architecture? 3](#_Toc384392406)

[1.2 Language Support 3](#_Toc384392407)

[1.3 Platform Support 3](#_Toc384392408)

[1.4 Resources Onsite 3](#_Toc384392409)

[II. Parts and Terminology 3](#_Toc384392410)

[2.1 Hypergraph 3](#_Toc384392411)

[2.2 Modularity 3](#_Toc384392412)

[2.3 About Input File 3](#_Toc384392413)

[2.4 About Output File 4](#_Toc384392414)

[III. How to use Architecture System 4](#_Toc384392415)

[3.1 Download 4](#_Toc384392416)

[3.2 Install Environment 4](#_Toc384392417)

[3.3 Starting your projects 4](#_Toc384392418)

[Windows(VS2005) 4](#_Toc384392419)

[Linux 6](#_Toc384392420)

[3.4 Another tip 6](#_Toc384392421)

[IV. System Requirement and Contact Information 6](#_Toc384392422)

# I. Introduction

## 1.1 What is Architecture?

The software architecture of a program or computing system is a depiction of the system that aids in the understanding of how the system will behave. It serves as the blueprint for both the system and the project developing it, defining the work assignments that must be carried out by design and implementation teams. The architecture is the primary carrier of system qualities such as performance, modifiability, and security. The Architecture system serves the metrics for software architecture in order to measure the reasonable of the given system.

## 1.2 Language Support

C++ present, java for future

## 1.3 Platform Support

Windows and Linux

## 1.4 Resources Onsite

Source code, download, manual, about us.

# II. Parts and Terminology

## 2.1 Hypergraph

The concept of hypergraph

## 2.2 Modularity

The definition and use of modularity

## 2.3 About Input File

|  |  |  |
| --- | --- | --- |
| Kinds of input files | Description | Format |
| Testcases files | All the source code files you want to analyze. | Folders including each source files, i.e., .h, .cpp, .cc. It may contain multi-projects files |
| Testcaese\_menu | Given the menu of your test cases including their paths | path:..//name of test cases files//the target you analyze 2 10 10 1 1 1 0 0 |

Explain the row of 2 10 10 1 1 1 0 0 here.

## 2.4 About Output File

There are following corresponding output files in txt format about each testcase: hypergraph files, partition result…

|  |  |
| --- | --- |
| Files Name | Description |
| ans.txt | Intermediate result for debugging. Presentation of the relationship among classes. |
| hypergraph.hgr | Intermediate result. The hypergraph model based on the source code. |
| partition.txt | Partition result based on the hypergraph partitioning algorithm. |
| time.txt | Recording the time-consuming in each procedure |

# III. How to use Architecture System

## 3.1 Download

You can get resources from the page <http://shaellancelot.github.io/> on our site.

## 3.2 Install Environment

Windows: Visual Studio 2005

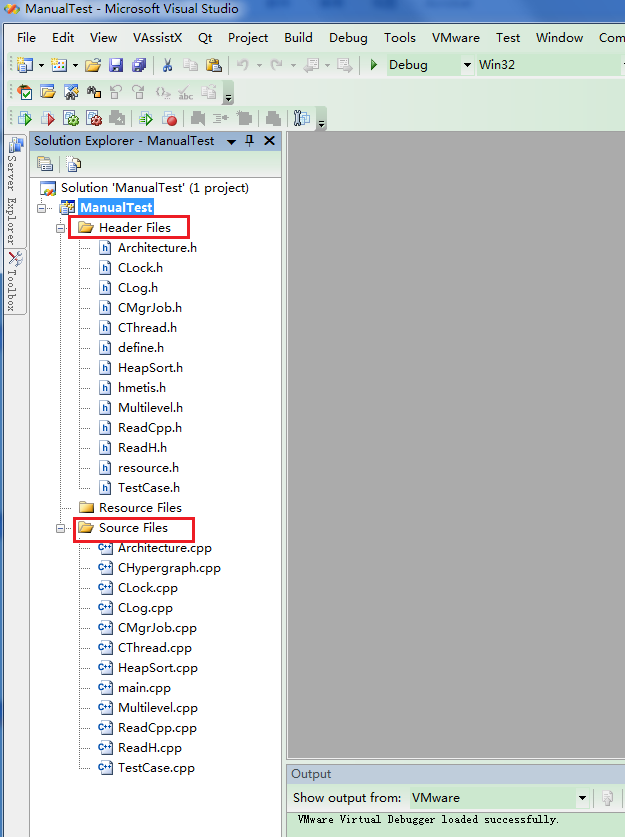
Linux: Ubuntu

## 3.3 Starting your projects

### Windows(VS2005)

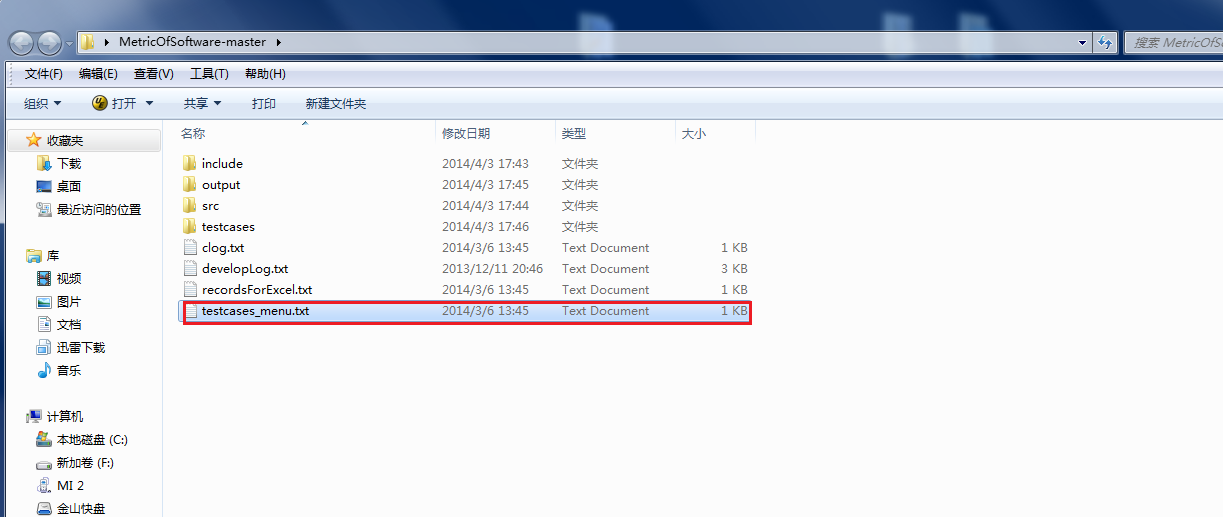
Create an empty project -> select “Header Files” with Right Click -> Select Add -> Select Existing item… -> …and the same to the Source Files

Beginning with the start page of Visual Studio 2005, create a new empty project, and add the source code (including .h and .cpp files) to your project.



Before running the project, you should give the right form of the input (see the section 2.3).

Namely, your test cases what you want to analyze should be added in the root directory of your project as well as the test case menu.



Running the project….

The first stage is parsing the source code in testcases files listed in the file of testcase\_menu.txt, and following the hypergraph modeling based on the parsing results, and the third stage is to partition hypergraph with the hypergraph algorithm. The partitioning result is shown in the file of partition.txt.

我们的界面？？

### Linux

编译执行Makefile的过程

## 3.4 Another tip

The stages in Architecture is separated that you can access our partitioning algorithm to deal with other clustering problem.

# IV. System Requirement and Contact Information

Architecture is written in C++, and it has been tested on Windows and Linux. Notwithstanding, it does not mean the Architecture is bug free. If you find any problem, please send email to ? with a brief description of the problem. Our future version will be available on <http://shaellancelot.github.io/>.