



BITMiniCC简介

计卫星，李侃，王贵珍

北京理工大学计算机学院



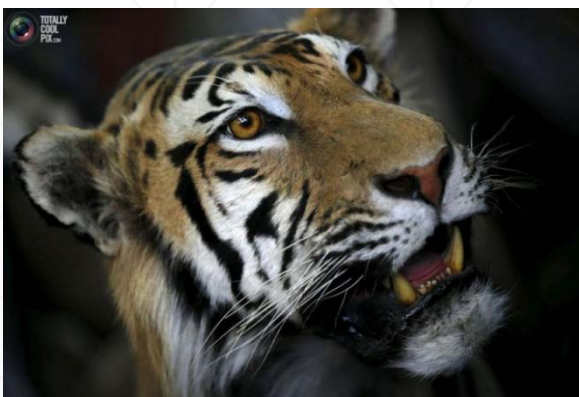
提纲

1. BITMiniCC简介
 2. 框架介绍
 3. 框架扩展方法
 4. 框架展示
 5. 课程实践作业要求
 6. Q&A
-



BITMiniCC简介

- 课程实验存在的问题
 - 从理论到实践的距离：我听明白了，但是还是不知道怎么实现
 - 从前端到后端的距离：词法分析实现了，但是不怎么好
 - 从理想到现实的距离：老师，我这学期有5门课。。。



大作业A？

大作业B？



大作业C？

竞赛、实验室项目？



BITMiniCC简介

- BITMiniCC = **BIT Mini C** Compiler
 - 一个迷你C语言编译器
 - 源语言 : C语言子集
 - 目标语言 : MIPS机器语言 , X86机器语言
 - 宿主语言 : Java / Java+C / Java+C# / Java+Python
 - 开发现状
 - 2014年具有初步想法
 - 2015年12起步
 - 目前仍在建设之中...
-



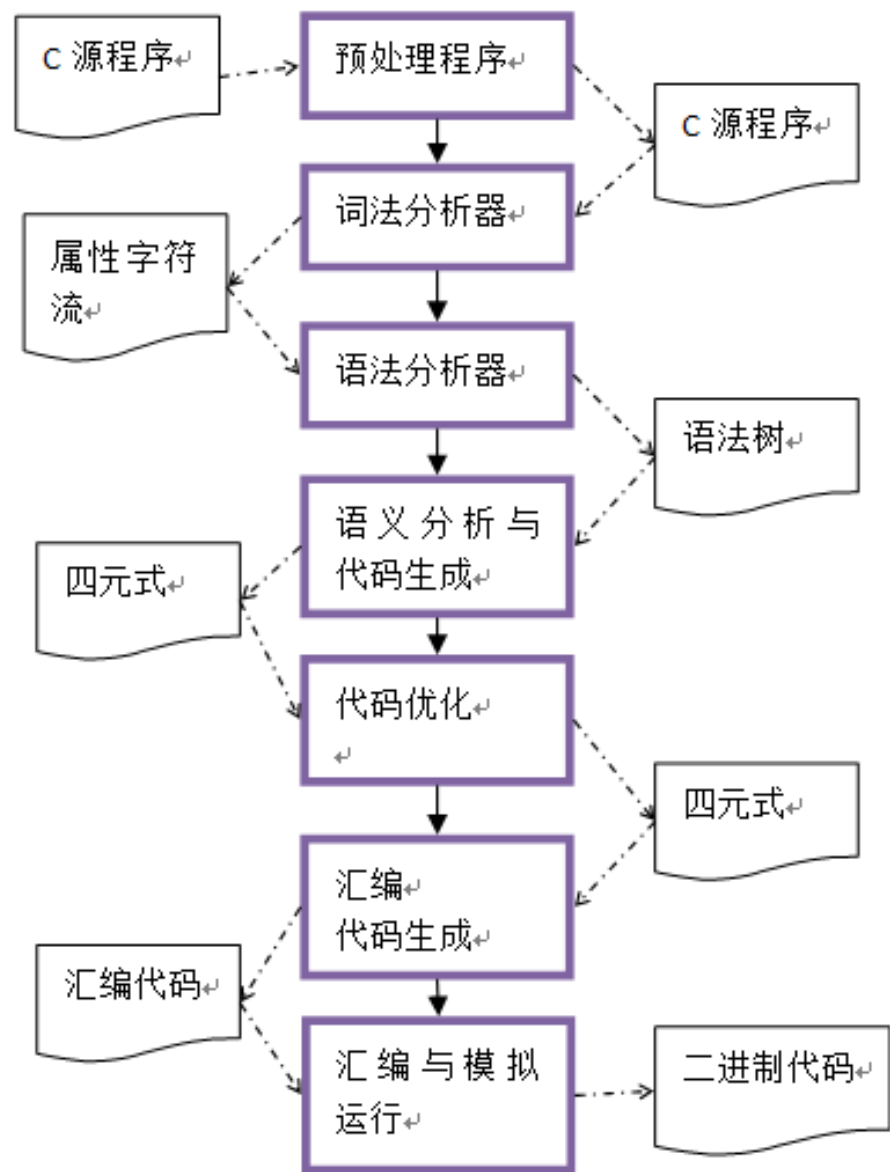
BITMiniCC简介

- 设计目标
 - 提供一个可参考的运行实例
 - 采用熟悉的语言实现：C/C++、Java、C#和Python等
 - 内部集成了各个部分的实现
 - 中间处理结果可见：标准XML
 - 每个过程可替换：前面做的不好没关系，可以用现有的
 - 除此之外
 - 框架源码公开，但是内部实现不对外公开
 - 代码复制检测：框架能极大降低检测范围
 - 最后结果测试自动化：标准输入输出使得自动测试成为可能
-



框架介绍

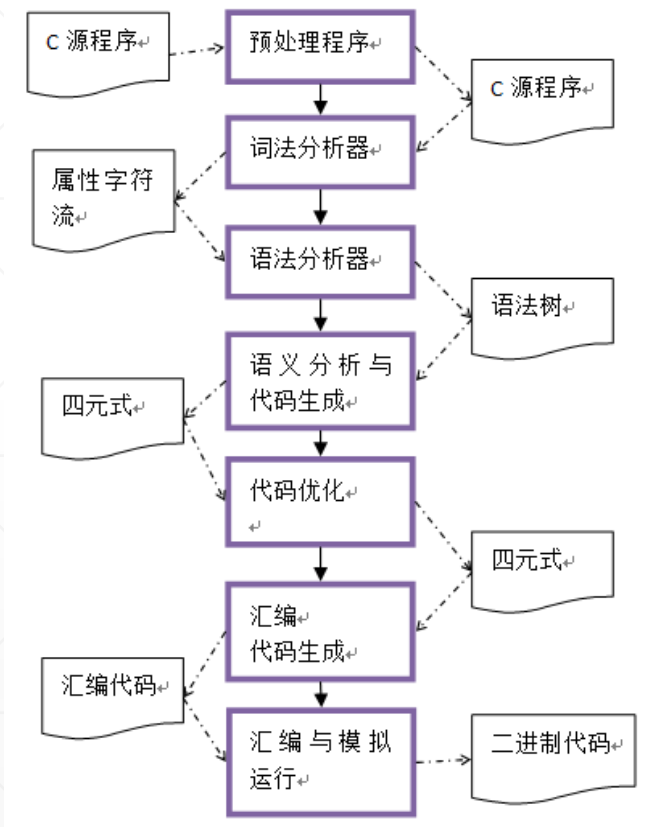
- 整体结构





框架介绍

- 特点
 - 输入输出标准化
 - 单个模块可替换
 - 只需关注单个模块的设计
 - 内部模块集成实现
 - 内部实现和自主实现可组合



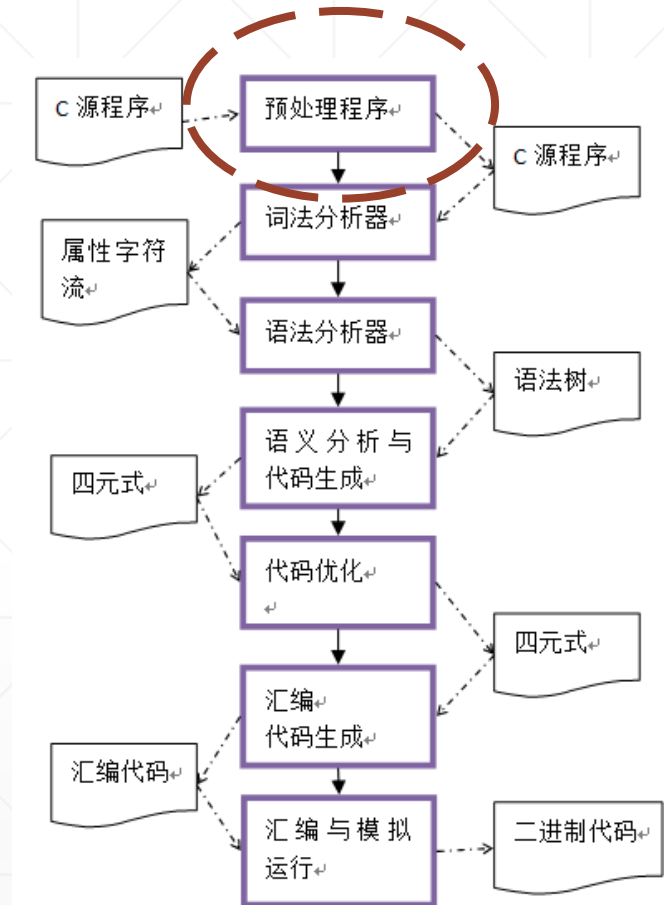


框架介绍

- 预处理
 - 输入：源程序
 - 输出：处理后的源程序
 - 功能
 - 文件包含
 - 宏替换
 - 删除注释
 - 无用空白删除

```
int main(int a, int b){ //main function
    return a + b;
}
```

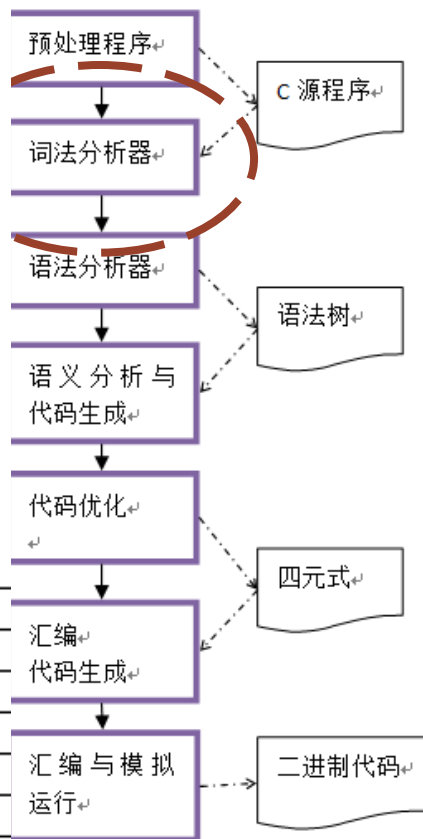
 `int main(int a, int b){ return a + b; }`





- 输入：清理后的源程序
- 输出：属性字符流
- 功能：根据词法规则识别
输出相应的属性字

```
int main(int a, int b) {
```

[illegible]



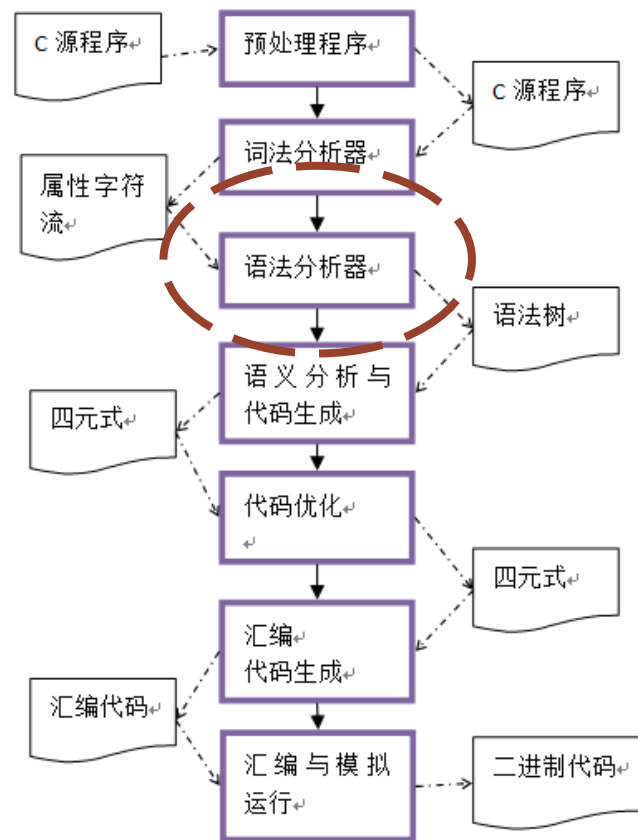
框架介绍

- 语法分析
 - 输入：属性字符流
 - 输出：语法树
 - 功能：根据C语言的语法规则，识别输出程序的结构，
 - 输出语法错误或者语法树

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="test.1">
  <tokens>
    <token>
      <number>1</number>
      <value>int</value>
      <type>keyword</type>
      <line>1</line>
      <valid>true</valid>
    </token>
    <token>
      <number>2</number>
      <value>main</value>
      <type>identifier</type>
      <line>1</line>
      <valid>true</valid>
    </token>
  </tokens>
</project>
```



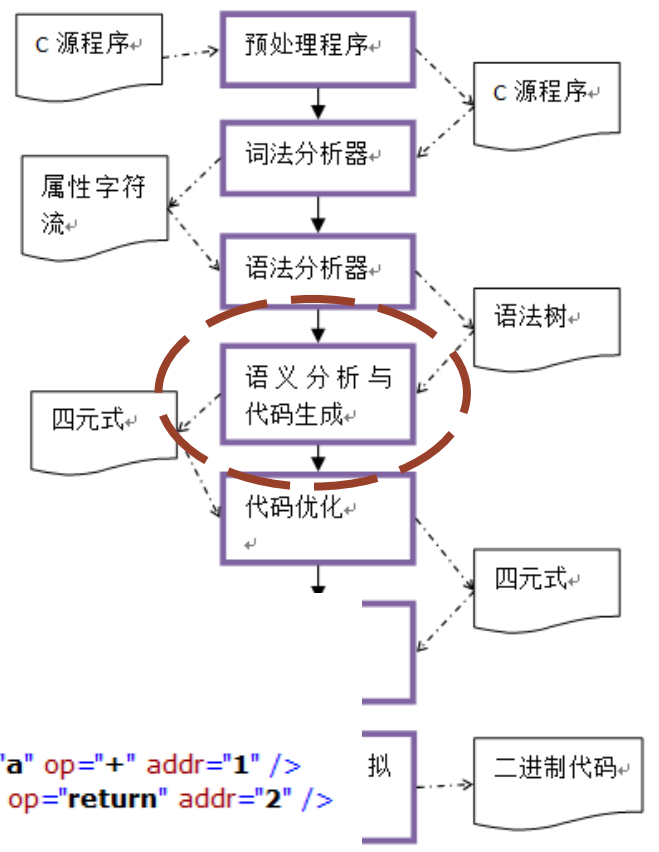
```
<?xml version="1.0" encoding="UTF-8"?>
<ParserTree name="test.tree.xml">
  <PROGRAM>
    <FUNCTIONS>
      <FUNCTION>
        <TYPE>
          <identifier>main</identifier>
          <separator>(</separator>
            <ARGS>
              <FARGS>
                <ALIST>
                  </ARGS>
                <separator>)</separator>
              <FUNC_BODY>
                <PLUS1 />
              </FUNCTION>
            </FUNCTIONS>
          </PROGRAM>
        </ParserTree>
```





框架介绍

- 语义分析
 - 输入：属性字符流
 - 输出：语法树
 - 功能：根据C语言的语义规则，识别输出程序的结构，
 - 输出语法错误或者语法树



```
<?xml version="1.0" encoding="UTF-8"?>
<ParserTree name="test.tree.xml">
  <PROGRAM>
    <FUNCTIONS>
      <FUNCTION>
        <TYPE>
          <identifier>main</identifier>
        <separator></separator>
        <ARGS>
          <FARGS>
          </FARGS>
          <ALIST>
          </ALIST>
          <separator></separator>
        <FUNC_BODY>
          </FUNCTION>
          <PLUS1 />
        </FUNCTIONS>
      </PROGRAM>
    </ParserTree>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
- <IC name="test.ic.xml">
- <functions>
- <function>
  <quaternion result="T1" arg2="b" arg1="a" op="+" addr="1" />
  <quaternion result="" arg2="T1" arg1="" op="return" addr="2" />
</function>
</functions>
</IC>
```



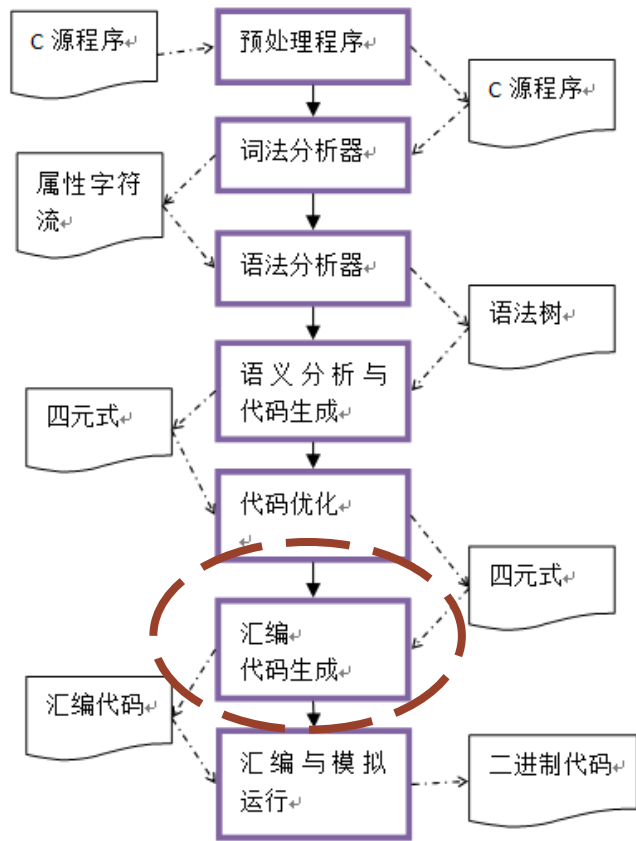
框架介绍

- 代码生成
 - 生成目标机汇编代码

```
<?xml version="1.0" encoding="UTF-8" ?>
- <IC name="test.ic.xml">
- <functions>
- <function>
  <quaternion result="T1" arg2="b" arg1="a" op="+" addr="1" />
  <quaternion result="" arg2="T1" arg1="" op="return" addr="2" />
</function>
</functions>
</IC>
```



```
Edit Execute
test.code.s
1 .data
2 T1: .word
3 a: .word
4 b: .word
5 .text
6 .globl main
7 main:
8     la $a0, T1
9     la $t1, a
10    la $t2, b
11    add $t3, $t1, $t2
12    sw $t3, 0($a0)
13
```



框架介绍

- 模拟运行：MARS

```
D:\projects\bit-minic-clean\input\test.code.s - MARS 4.5
File Edit Run Settings Tools Help
Run speed
test.code.s
1 .data
2 T1: .word
3 a: .word
4 b: .word
5 .text
6 .globl main
7 main:
8     la $a0, T1
9     la $t1, a
10    la $t2, b
11    add $t3, $t1, $t2
12    sw $t3, 0($a0)
13
```

```
test.code.s
1 .data
2 T1: .word
3 a: .word
4 b: .word
5 .text
6 .globl main
7 main:
8
9
10
11
12
13
```

EditExecute

Text Segment

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x3c011001	lui \$1, 0x00001001	8: la \$a0, T1
<input type="checkbox"/>	0x00400004	0x34240000	ori \$4, \$1, 0x00000000	
<input type="checkbox"/>	0x00400008	0x3c011001	lui \$1, 0x00001001	9: la \$t1, a
<input type="checkbox"/>	0x0040000c	0x34290000	ori \$9, \$1, 0x00000000	
<input type="checkbox"/>	0x00400010	0x3c011001	lui \$1, 0x00001001	10: la \$t2, b
<input type="checkbox"/>	0x00400014	0x342a0000	ori \$10, \$1, 0x00000000	
<input type="checkbox"/>	0x00400018	0x012a5820	add \$t1, \$9, \$10	11: add \$t3, \$t1, \$t2
<input type="checkbox"/>	0x0040001c	0xac8b0000	sw \$t1, 0x00000000 (\$4)	12: sw \$t3, 0(\$a0)

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

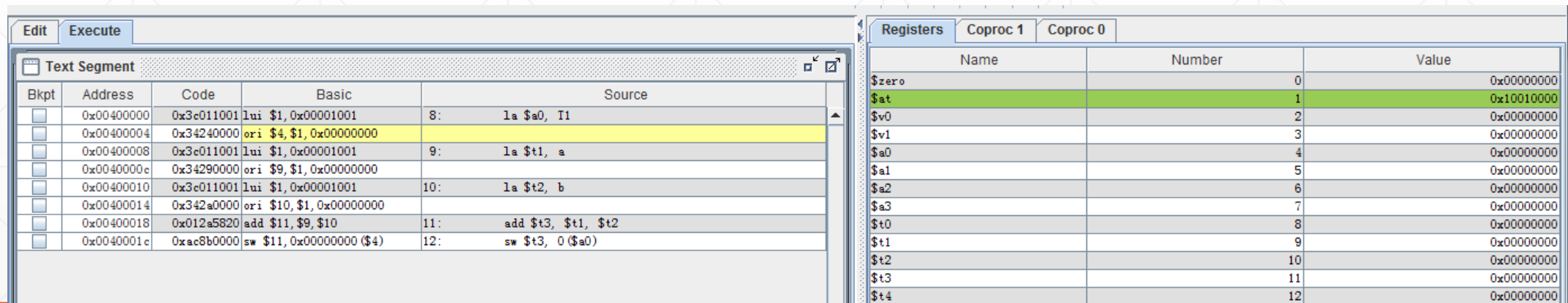
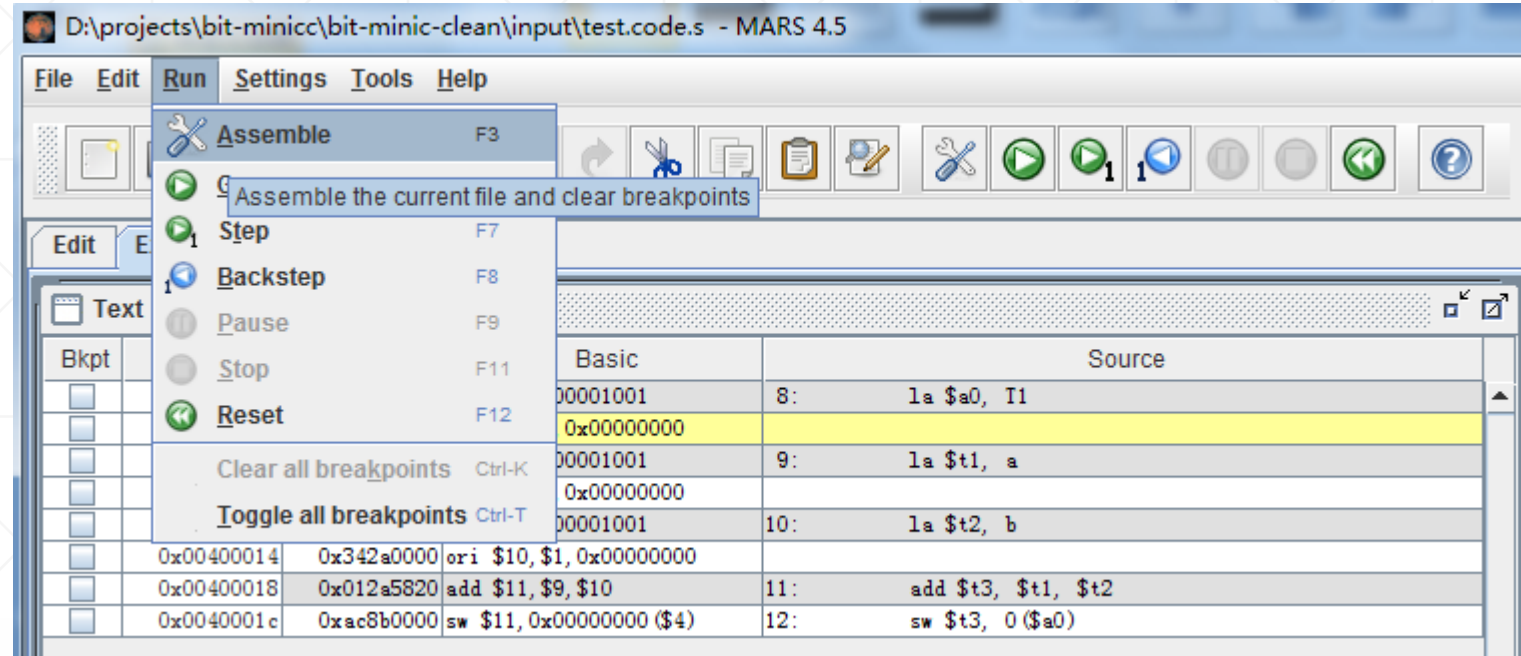
0x10010000 (.data)

☒ Hexadecimal Addresses☒ Hexadecimal Values☐ ASCII



框架介绍

- 模拟运行





框架介绍

- MARS相关资料
 - <http://courses.missouristate.edu/KenVollmar/MARS/>

The screenshot shows the Missouri State University website. At the top is a dark red header with the university's logo and name. Below the header is a search bar and a list of letters. The main content area features a large image of the planet Mars on the left. To the right of the image is the title "MARS (MIPS Assembler and Runtime Simulator)" and a subtitle "An IDE for MIPS Assembly Language Programming". Below the subtitle is a paragraph describing MARS as a lightweight interactive development environment (IDE) for programming in MIPS assembly language. To the left of this paragraph is a vertical navigation menu with buttons for Home, Features, Download, License, Papers, Help & Info, and Contact Us. To the right of the paragraph is a Softpedia certification badge stating "100% FREE" and "NO SPYWARE", "NO ADWARE", "NO VIRUSES". Below the badge is a quote from Softpedia dated Feb. 2013, praising MARS for being clean of malware. At the bottom of the page is a link to download MARS from Softpedia, with a note that the version on Softpedia may lag behind the version on the page.

Missouri State
UNIVERSITY

Search
a b c d e f g h i j k l m
n o p q r s t u v w x y z

MARS (MIPS Assembler and Runtime Simulator)
An IDE for MIPS Assembly Language Programming

MARS is a lightweight interactive development environment (IDE) for programming in MIPS assembly language, intended for educational-level use with Patterson and Hennessy's *Computer Organization and Design*.

100% FREE
NO SPYWARE
NO ADWARE
NO VIRUSES
SOFTPEDIA™
certified by www.softpedia.com

Feb. 2013: "MARS has been tested in the Softpedia labs using several industry-leading security solutions and found to be completely clean of adware/spyware components. ... Softpedia guarantees that MARS 4.3 is 100% FREE, which means it does not contain any form of malware, including spyware, viruses, trojans and backdoors."

[Download MARS from Softpedia](#) (version on Softpedia may lag behind the version on this page).

Home
Features
Download
License
Papers
Help & Info
Contact Us



框架介绍

- 运行环境
 - 需要java JRE ≥ 1.7
- 运行参数
 - `java -jar BITMiniCC.jar xxx.c`
 - `run.bat xxx.c`



```
D:\projects\bit-minicc\bit-minic-clean\run>run.bat D:\projects\bit-minicc\bit-minic-clean\input\test.c

D:\projects\bit-minicc\bit-minic-clean\run>java -jar BITMiniCC.jar D:\projects\bit-minicc\bit-minic-clean\input\test.c
Start to compile ...
1. PreProcess finished!
2. LexAnalyse finished!
3. Parse finished!
4. Semantic finished!
5. Intermediate code generate not finished!
6. Optimize not finished!
OP: return
7. Code generate finished!
8. Simulate not finished!
Compiling completed!
```




框架介绍

- 常见运行问题

```
D:\projects\bit-minicc\bit-minic-clean\run>java -jar BITMiniCC.jar input.c
Start to compile ...
Exception in thread "main" java.lang.UnsupportedClassVersionError: bit/minisys/minicc/pp/MiniCCPrePr
ocessor : Unsupported major.minor version 52.0
    at java.lang.ClassLoader.defineClass1(Native Method)
    at java.lang.ClassLoader.defineClass(Unknown Source)
    at java.security.SecureClassLoader.defineClass(Unknown Source)
    at java.net.URLClassLoader.defineClass(Unknown Source)
    at java.net.URLClassLoader.access$100(Unknown Source)
    at java.net.URLClassLoader$1.run(Unknown Source)
    at java.net.URLClassLoader$1.run(Unknown Source)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.net.URLClassLoader.findClass(Unknown Source)
    at java.lang.ClassLoader.loadClass(Unknown Source)
    at sun.misc.Launcher$AppClassLoader.loadClass(Unknown Source)
    at java.lang.ClassLoader.loadClass(Unknown Source)
    at bit.minisys.minicc.MinicCompiler.run(MinicCompiler.java:94)
    at bit.minisys.minicc.BITMiniCC.main(BITMiniCC.java:31)
```



框架介绍

- 基本配置: config.xml
 - skip : 是否跳过该阶段运行
 - type : 模块实现方法
 - java/binary/python
 - path : 路径
 - Binary
 - simulator
 - name : 阶段名称

```
<?xml version="1.0" encoding="UTF-8" ?>
- <config name="config.xml">
- <phases>
- <phase>
    <phase skip="false" type="java" path="" name="pp" />
    <phase skip="false" type="java" path="" name="scanning" />
    <phase skip="false" type="java" path="" name="parsing" />
    <phase skip="false" type="java" path="" name="semantic" />
    <phase skip="false" type="java" path="" name="icgen" />
    <phase skip="false" type="java" path="" name="optimizing" />
    <phase skip="false" type="java" path="" name="codegen" />
    <phase skip="false" type="java" path="" name="simulating" />
  </phase>
</phases>
</config>
```



框架介绍

- 内部集成的功能
 - 预处理：注释和宏替换
 - 词法分析：所有单词
 - 语法：
 - 语义：暂无
 - 中间代码：四元式
 - 优化：暂无
 - 代码生成：MIPS

```
PROGRAM → → → ·FUNCTIONS↵
FUNCTIONS → → → ·FUNCTION ·FLIST↵
FLIST → → → → ·FUNCTION ·FLIST · | · ε↵
FUNCTION → → → → ·TYPE ·TKN_ID ·TKN_LP ·ARGS ·TKN_RP ·FUNC_BODY↵
ARGS → → → → → ·FARGS ·ALIST · | · ε↵
ALIST → → → → → ·TKN_COMMA ·FARGS ·ALIST · | · ε↵
FARGS → → → → → ·TYPE ·TKN_ID↵
FUNC_BODY → → → → → ·TKN_LB ·STMTS ·TKN_RB↵
STMTS → → → → → ·STMT ·STMTS · | · ε↵
STMT → → → → → ·EXPR_STMT · | ·RET_STMT↵
EXPR_STMT → → → → → ·EXPR ·TKN_SEMICOLON↵
RET_STMT → → → → → ·TKN_KW_RET ·EXPR_STMT↵
EXPR → → → → → ·TERM ·TLIST↵
TLIST → → → → → ·TKN_PLUS ·TERM ·TLIST · | · ε↵
TERM → → → → → ·FACTOR ·FLIST↵
FLIST → → → → → ·TKN_MUL ·FACTOR ·FLIST · | · ε↵
FACTOR → → → → → ·TKN_LP ·EXPR ·TKN_RP · | ·TKN_ID↵
TYPE → → → → → ·TKN_INT · | ·TKN_FLOAT↵
```



扩展方法

- 从github下载框架
 - <https://github.com/jiweixing/bit-minic-compiler>

jiweixing / bit-minic-compiler

Unwatch 2 Star 2 Fork 2

Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings

A C compiler framework in Java — Edit

3 commits 1 branch 0 releases 1 contributor

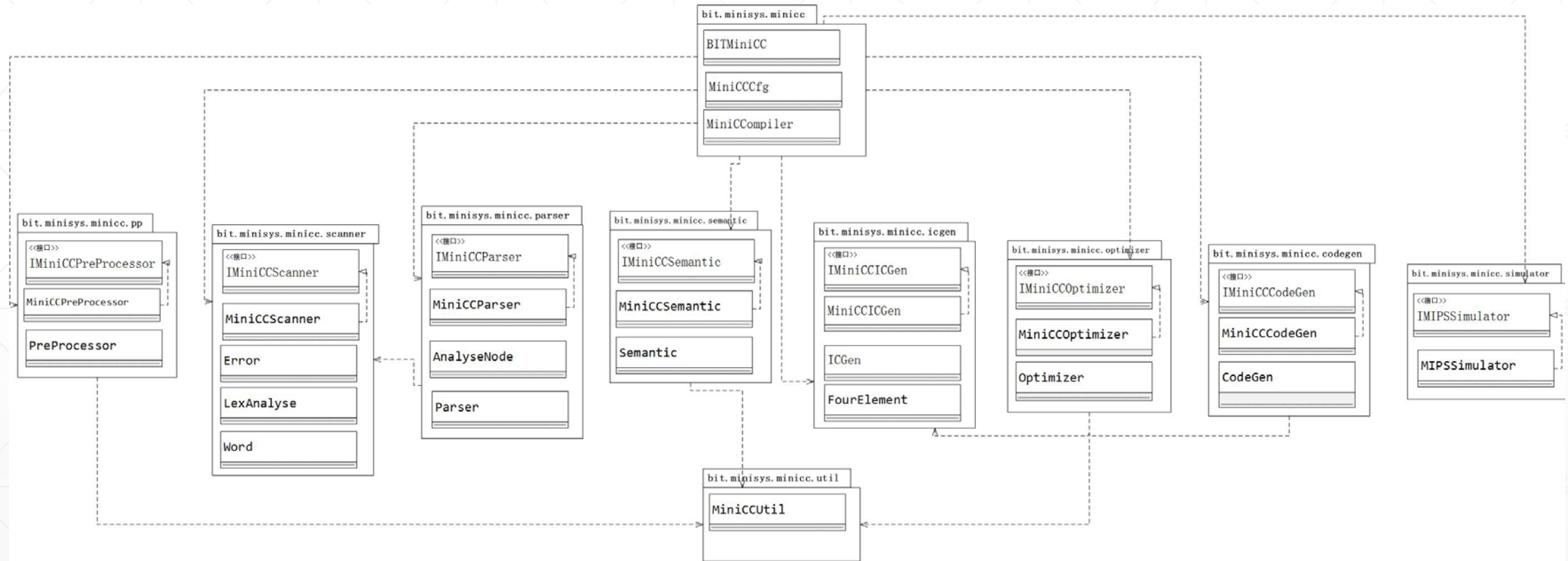
Branch: master New pull request New file Upload files Find file HTTPS https://github.com/jiweixing/ Download ZIP

jiweixing update libraries Latest commit 0aefb17 11 days ago		
bin	first commit	14 days ago
input	first commit	14 days ago
lib	update libraries	11 days ago
output	first commit	14 days ago
src/bit/minisys/minicc	add interface	11 days ago
.classpath	first commit	14 days ago
.project	first commit	14 days ago

Help people interested in this repository understand your project by adding a README. Add a README



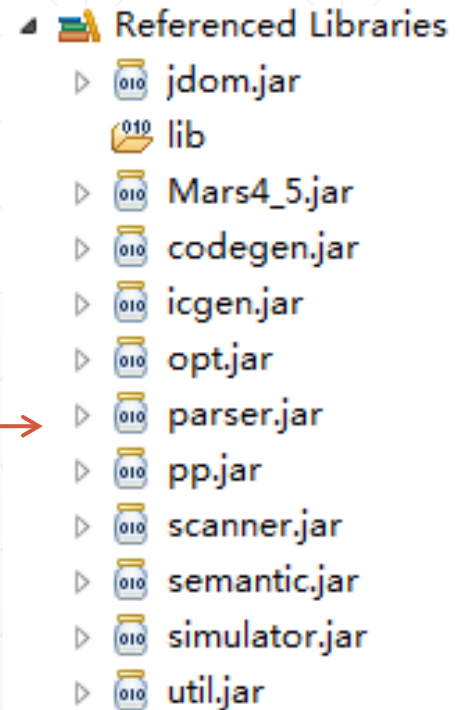
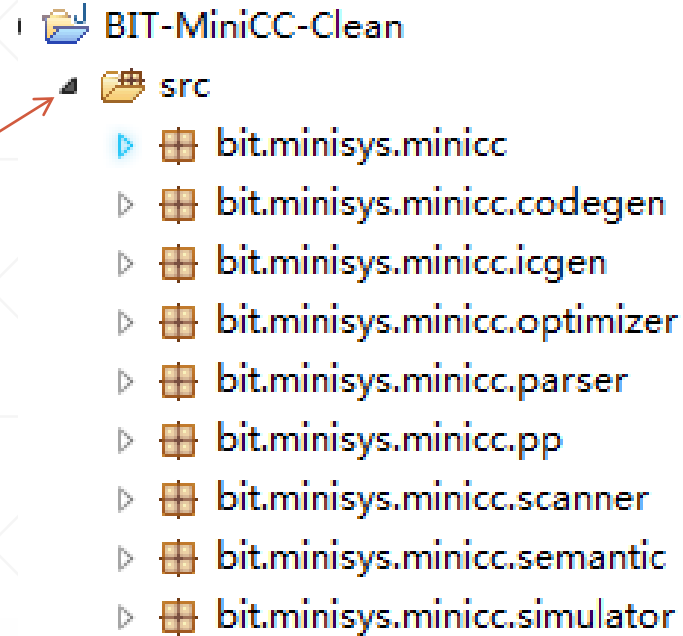
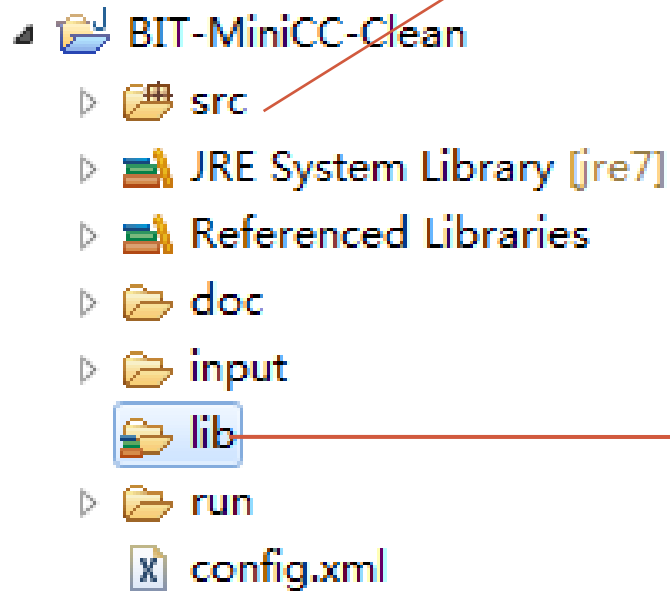
扩展方法





扩展方法

- 源代码树





扩展方法

- Java语言
 - 以词法分析器为例

```
package bit.minisys.minicc.scanner;

import java.io.IOException;

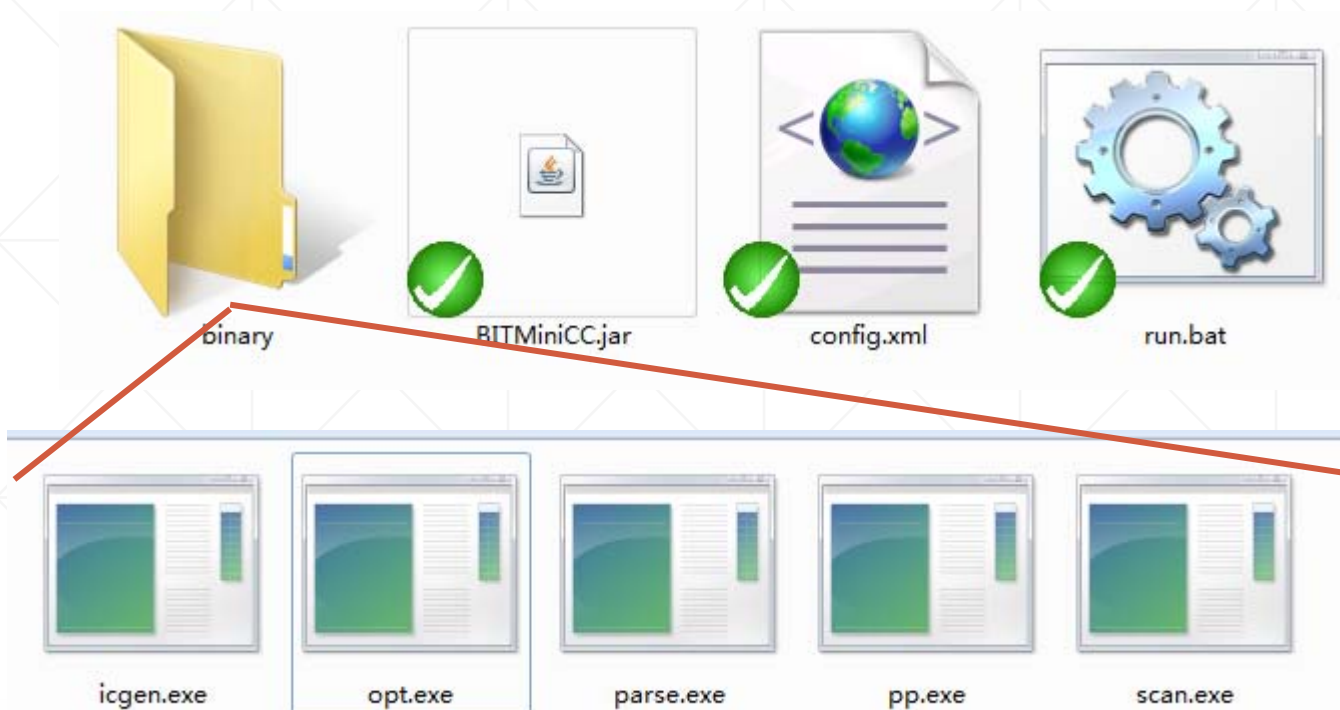
public class MiniCCScanner implements IMiniCCScanner{

    public void run(String iFile, String oFile) throws IOException{
        LexAnalyse lex = new LexAnalyse();
        lex.lexAnalyse1(iFile);
        lex.outputWordList(oFile);
        System.out.println("2. LexAnalyse finished!");
    }
}
```



扩展方法

- C/C++/C#



```
<?xml version="1.0" encoding="UTF-8" ?>
- <config name="config.xml">
-   <phases>
-     <phase>
-       <phase skip="false" type="binary" path="D:\projects\bit-minicc\bit-minic-clean\run\binary\pp.exe" name="pp" />
-       <phase skip="false" type="binary" path="D:\projects\bit-minicc\bit-minic-clean\run\binary\scan.exe" name="scanning" />
-       <phase skip="false" type="java" path="" name="parsing" />
-       <phase skip="false" type="java" path="" name="semantic" />
-       <phase skip="false" type="java" path="" name="icgen" />
-       <phase skip="false" type="java" path="" name="optimizing" />
-       <phase skip="false" type="java" path="" name="codegen" />
-       <phase skip="false" type="java" path="" name="simulating" />
-     </phase>
-   </phases>
- </config>
```




扩展方法

- Python
 - 正在建设当中





课程实践作业具体要求

- 阶段工作
 - 词法分析、语法分析、中间代码生成必选
 - 其他阶段可选
- 词法分析
 - 支持源语言所有单词类：标示符、关键词、运算符、常量和分隔符
 - 输入输出按照框架进行
- 语法分析
 - 支持语句：赋值语句、返回语句、一种分支语句和一种循环语句
 - 运算符：+、-、*、/

具体要求按照教学班进行



课程实践作业具体要求

- 提交内容
 - 公共：实验报告（目标，过程，方法，结果和心得）
 - 语言相关
 - Java：项目打包
 - C/C++：源代码，框架的run目录（binary下需要有已经编译后的结果）
 - 个人打包命名
 - 目录结构：src，bin，doc
 - 最终实验提交：学号_姓名.zip
 - 阶段性提交：学号_姓名_预处理.zip
 - **提交纸质版实验报告**
-



2016奖励计划

- 以教学班为单位
 - 最佳阶段实现奖若干名：+5分
 - 自主参评
 - 教学班最佳，超过已有的实现
 - 可空缺
 - 最佳框架改进奖1名：+3分
 - 自主参评，全年级范围内评选
 - 对现有框架进行改造，更加符合教学的实际情况
 - 可空缺





Q & A ?

