

# 一种支持编译器构造全过程分阶段自动评判的实验教学案例

## 1. 案例概述

编译技术是一门理论与实践紧密结合的课程，学生在学习理论知识的同时，同步开展编程实践，不仅有助于巩固理论知识，也能将一个复杂的完整编译器的构造分解为多个阶段任务，让同学有充足的时间进行设计和编码，“分而治之”，有利于独立思考和独立完成。

本案例介绍一种在编译技术实验环节中，将编译器构建的全过程划分为若干阶段，每个阶段都能实现自动评判的实验教学方案，在信息化的云端教学平台(judge.buaa.edu.cn)的辅助下，为学生随时随地开展课程实践、实现精细化过程管理和辅导提供有力支持。本实验教学方案将自学与辅导相结合，培养学生独立解决问题的能力。首先通过阅读理解文法的定义，写出符合文法规则并覆盖所有规则的程序，再分阶段、增量式实现编译器的词法分析、语法分析、错误处理、代码生成、代码优化功能，每个阶段提交的作业，都要求输出一种可进行自动评判的数据；在这个过程中，一方面引导学生阅读经典的教学案例 Pascal-S 编译器源代码，分阶段进行设计；另一方面用学生提交的测试程序，构建公共测试程序库，用于学生的调试并完善编译器；通过上述步骤督促学生一步步完成整个编译系统的设计、编码和测试，并挑战优化的竞速排名。

本案例给出一个文法的定义，分阶段构造编译器并进行自动评判的实验步骤、以及针对该文法的各阶段作业的测试程序及预期结果（见附件）。本案例的步骤和测试数据都可以在教学平台上进行展示。使用此案例的学校还可以自定义文法，

按照此本说明书中的步骤构造相应的测试程序以及输入输出数据即可进行评判。

本案例包含的文件为([下载链接](#)):

- (1) 案例说明书
- (2) 文法定义
- (3) 各阶段作业的测试程序及预期结果

## 2. 案例教学目标

本案例通过将编译器的构建过程分解为各个步骤，分阶段、增量式开发完成编译器，加强过程培养、过程考核，渐进式培养学生的能力，通过与理论课堂学习同步进行，自动评判及时发现学生在学习中存在的问题。

## 3. 案例准备

(1) 支撑平台：计算机专业课一体化平台 (judge.buaa.edu.cn, 公网地址 <https://course.educg.net/>) 能支持本案例描述的分阶段构造编译器并进行自动评判的实验方案;

计算机专业课一体化平台

比赛 OnlineJudge 教师登录

最新公告

- [编译技术] 关于目标代码 1 的进一步说明 2020/11/07
- [高级语言程序设计 (高工)] 作业评判反馈信息更改通知 2020/11/07
- [高级语言程序设计 (高工)] 20级第一次作业填空选择题答案 2020/10/23
- [编译技术] 编译实验模拟上机考试及期中考试通知 2020/10/17
- [高级语言程序设计 (一)] 欢迎参加2021级推免研究生上机测试, 请阅读考试须知! 2020/10/09

希望

在线用户数  
Number of online users  
90  
在线用户

学生入口

学号

密码

登录

忘记密码

北京航空航天大学 计算机学院  
若重置密码, 请与当前的任课教师联系

请注意完善您的个人信息，特别是Email地址，可以帮助您找回密码。

密码强度不够，为保证您的信息安全，务必修改密码。

最新公告	
关于目标代码的进一步说明	2020/11/07
实验作业生成的目标代码选择CODE代码和MIPS代码的其中一种完成即可，如选择生成MIPS代码，需完成所给的真值树作业。关于CODE代码和MIPS代码的进一步说明如下：...	
关于编译实验成绩组成的说明	2020/11/04
正常情况下，实验作业及考试成绩所占的比例如下：文法解析、语法分析、语义分析、编译原理、代码生成作业各占3%，代码生成占20%。...	
编译实验模拟上机考试及期中考试通知	2020/10/17
2020年10月17日将进行编译实验的期中上机考试。为了让同学们更好地完成期中考试，10月20日将进行一次模拟考试。请同学们仔细阅读以下要求或指南：...	
编译实验作业的热点问题解答	2020/09/30
在今天实验课的上机过程中，发现同学们遇到以下问题关注得比较多，现统一做解答：在新去楼机房开机后，在“管理虚拟机”界面，高德群Comput.College实验课的作业都在“...	
新版开发环境及部署说明发布	2020/09/24
由于VS2019和Win10 19H2存在兼容性问题，现提供另外一种新的部署方法，更新了开发环境及部署说明文档，请参照指定“课程说明”...	
发布词法分析测试程序的通知	2020/09/23
为便于同学们进行测试，已将词法分析公共测试程序中的测试程序上传至平台，请上传各个文件并分别命名为“2020词法分析测试程序”的相应作业名称。...	
布置词法分析作业及上机注意事项说明	2020/09/22
今天布置词法分析作业，请大家登录ojp平台，在“在线作业”中查看，在每周二上午9:00（5:50-12:15）的课上课前完成。...	

正在进行的作业	
2020代码生成测试作业 (不计入成绩)	【34天4小时后截止】
2020代码生成阶段设计文档	【34天4小时后截止】
2020代码生成作业一	【6天4小时后截止】
2020年文法定义(用于查看文法, 不计入成绩)	【41天4小时后截止】

正在进行的在线实验	
2020实验作业测试	【50天4小时后截止】

(2) 文法定义：为实现编译器，首先要定义描述源语言的文法，目前平台定义了两类 C 语言文法，具有常量、变量、整数、字符、字符串、整型、字符型、一维/二维数组、函数（带参数）、赋值语句、if 语句、while 语句、for 语句、switch-case 语句、函数调用、复合语句、读语句、写语句（带字符串）；提供对这两种文法编译器的自动评判。如果教学中采用自定义的文法，需要自行构造符合文法的测试程序，并为各阶段产生测试程序的相应的预期结果，作为自动评判的依据。

(3) 预备知识：学生需要通过编译技术理论课的概论部分，了解编译的全过程；并通过阅读经典编译器源代码，了解编译器的典型组成部分和各组成部分之间的接口和传递的数据，以便于在分阶段、增量式完成编译器的开发时，能够设计松耦合、高内聚的模块，便于修改和调试。

## 4. 案例教学要点

(1) 通过让学生构造一个完整编译器，1) 让学生掌握编译系统的功能、原理和构造方法（不局限于特定的程序设计语言和目标机）；理解高级程序设计语言的运行机理；2) 具备完整小型编译系统的设计、实现和测试的能力；从而理

解将一种程序设计语言程序到另一种程序等价转化的理论和方法；3) 针对特定硬件平台进行编译优化能力；理解程序的安全问题。

(2) 完成一个完整编译器是一个复杂的过程，针对不同学习程度的同学，设计了 PCODE 和 MIPS 汇编两种形式不同难度的目标代码、不同的中间代码要求和代码优化要求，让同学既能感受到构造一个完整编译器的获得感，又不至于因为难度太大而放弃或者抄袭。

(3) 学生在完成词法分析、语法分析、错误处理功能之后，可以选择后续阶段生成的目标代码，若选择生成 PCODE 代码，则需要完成解释执行程序，解释执行产生的 PCODE；若选择生成 MIPS 汇编，则要生成四元式中间代码，再生成 MIPS 汇编，并进行代码优化。生成的 MIPS 汇编在模拟器 Mars 上运行，得到运行结果。

## 5. 案例教学组织方式

本案例展示的实验过程包括以下步骤，学生在每个步骤都需要按要求完成编译器相应功能的开发，并按要求输出编译器产生的过程数据到指定的文件中。所有步骤的评判均可以在教学平台上自动完成，教学平台中展示的学生的提交行为数据和评判结果数据有助于帮助师生及时发现学生在学习中的问题，迅速做出改进。

### (一) 文法解读

**目标和要求：**根据具体文法理解课堂教学内容，考核学生对文法基础知识的掌握情况，培养学生编写符合文法规则的测试程序的能力。

学完“第 2 章文法和语言的概念和表示”开始完成此项作业，学生在分析理解文法的基础上编写符合该文法的测试程序，要求测试程序覆盖文法中所有的语法

成分，以及语法成分的常见组合情况。

2020文法解读作业 / 通用评测题 / 1. 2020文法解读作业题目

本课程的实验，是根据给定的文法实现一个小型编译器。该编译器的实现分成几个阶段来完成，请根据每次作业提示，逐步增加该编译器的模块，最终形成一个生成特定目标代码的完整编译器。

开始编写编译器代码之前，需要仔细阅读和分析给定的文法、设计编译器的总体结构和各模块之间的接口、考虑编译的每个阶段可能出现的错误类型。为此，第一项作业从文法的解读开始。

请仔细阅读下面的文法，要求对文法中每条规则所定义的语法成分进行分析，了解其作用、限定条件、组合情况和可能产生的句子，在此基础上，编写4-6个测试程序，要求所有测试程序能共同覆盖所有语法规则及每条规则内的常见组合情况。每个测试程序有且仅有10条写语句（每个测试程序的第一条写语句用printf("%c", <字符>); 输出自己的学号，其他写语句尽量反映由程序定义的数据及其运算结果，以便在后续阶段测试各种语法成分翻译的正确性）。请提供每个测试程序的输入数据（有<读语句>则提供，否则无需提供），输出数据（输出数据必须提供，若输入输出数据没有正确提供，评测时会报错），放到文件中，按下述要求为文件命名：

测试程序及对应的输入输出数据文件分别为

testfile1.txt	testin1.txt	testout1.txt
testfile2.txt	testin2.txt	testout2.txt
...		
testfilen.txt	testinn.txt	testoutn.txt

将以上文件打包为rar或zip文件后上传（请直接打包文件，不用置于文件夹中打包）。

注：本学期实验的所有作业都针对此文法，后续作业需在上一次作业的基础上进行增量式开发

```
<加法运算符> ::= + | - /*测试程序需出现2种符号*/
<乘法运算符> ::= * | / /*测试程序需出现2种符号*/
<关系运算符> ::= < | <= | > | >= | != /*测试程序需出现6种符号*/
<字母> ::= _ | a | ... | z | A | ... | Z /*测试程序需出现下划线、小写字母、大写字母3种情况*/
<数字> ::= 0 | 1 | ... | 9 /*测试程序至少出现1次数字*/
<字符> ::= '<加法运算符>' | '<乘法运算符>' | '<字母>' | '<数字>' /*测试程序至少出现4种类型的字符*/
<字符串> ::= " {十进制编码为32,33,35-126的ASCII字符} " /*字符串中要求至少有一个字符*/
```

需独立编写 4-6 个测试程序，通过评判在后续的阶段作业中，将用来建立公共测试程序库，将自动评判结果开放给学生，用于调试。

提交内容：4-6 个测试程序及配套的输入输出文件

输入输出：测试程序及对应的输入输出数据文件分别为

testfile1.txt	testin1.txt	testout1.txt
testfile2.txt	testin2.txt	testout2.txt
...		
testfilen.txt	testinn.txt	testoutn.txt

评判过程：学生提交的测试程序传入教学平台进行自动评判，若测试程序不符合语法规则，则不能得分；对于符合语法规则的测试程序，平台给出每个测试程序对语法规则的覆盖率，以及该学生提交的所有测试程序对语法规则的累积覆盖率。根据累积覆盖率给出得分率。

完成周期：1 周（第 2-3 周）

## (二) 词法分析

**目标和要求：**根据具体文法编写词法分析程序，考核学生对词法分析方法的掌握情况，培养学生编写符合词法规则的词法分析程序的能力。

2020词法分析作业 / 通用评测题 / 1. 2020词法分析作业题目

【问题描述】

请根据给定的文法设计并实现词法分析程序，从源程序中识别出单词，记录其单词类别和单词值，输入输出及处理要求如下：

(1) 数据结构和与语法分析程序的接口请自行定义；类别码需按下表格式统一定义；

(2) 为了方便进行自动评测，输入的被编译源文件统一命名为testfile.txt (注意不要写错文件名)；输出的结果文件统一命名为output.txt (注意不要写错文件名)，结果文件中每行按如下方式组织：

单词类别码 单词的字符/字符串形式(中间仅用一个空格间隔)

单词的类别码请统一按如下形式定义：

单词名称	类别码	单词名称	类别码	单词名称	类别码	单词名称	类别码
标识符	IDENFR	else	ELSETK	MINU	=	ASSIGN	
整形常量	INTCON	switch	SWITCHTK *	MULT	;	SEMICN	
字符常量	CHARCON	case	CASETK /	DIV	,	COMMA	
字符串	STRCON	default	DEFAULTTK <	LSS	{	LPARENT	
const	CONSTTK	while	WHILETK <=	LEQ	}	RPARENT	
int	INTTK	for	FORTK >	GRE	[	LBRACK	
char	CHARTK	scanf	SCANFTK >=	GEQ	]	RBRACK	
void	VOIDTK	printf	PRINTFTK ==	EQL	{	LBRACE	
main	MAINTK	return	RETURNTK !=	NEQ	}	RBRACE	
if	IFTK	+	PLUS :	COLON			

【输入形式】 testfile.txt中的符合文法要求的测试程序。

【输出形式】 要求将词法分析结果输出至output.txt中。

【特别提醒】 (1) 读取的字符串要原样保留着便于输出，特别是数字，这里输出的并不是真正的单词值，其实是读入的字符串，单词值需另行记录。

(2) 本次作业只考核对正确程序的处理，但需要为今后可能出现的错误情况预留接口。

学完“第 3 章 词法分析”开始完成此项作业，学生在理论课所学词法分析方法的基础上，设计实现词法分析程序，并按规定的格式输出单词信息。

**提交内容：**词法分析程序源代码

**输入输出：**为了方便进行自动评测，输入的被编译源文件统一命名为testfile.txt；输出的结果文件统一命名为 output.txt，结果文件中每行按如下方式组织：

单词类别码 单词的字符/字符串形式(中间仅用一个空格间隔)

**评判过程：**学生提交的词法分析程序传入教学平台，平台将 5 个测试程序依次输入学生的词法分析程序，分别运行得到输出结果，根据输出结果与预期结果一致部分所占的比例给分。

**公共测试库：**词法分析测试程序库，公布了 60 个测试程序及其词法分析程

## 序输出结果

完成周期：2周（第3-5周）

### （三）语法分析

**目标和要求：根据具体文法编写语法分析程序，考核学生对语法分析方法的掌握情况，培养学生编写符合文法的自顶向下语法分析程序的能力。**

2020语法分析作业 / 通用评测题 / 1.2020语法分析作业题目

#### 【问题描述】

请根据给定的文法设计并实现语法分析程序，能基于上次作业的词法分析程序所识别出的单词，识别出各类语法成分。输入输出及处理要求如下：

(1) 需按文法规则，用递归子程序法对文法中定义的所有种语法成分进行分析；

(2) 为了方便进行自动评测，输入的被编译源文件统一命名为testfile.txt（注意不要写错文件名）；输出的结果文件统一命名为output.txt（注意不要写错文件名）；结果文件中包含如下两种信息：

1) 按词法分析识别单词的顺序，按行输出每个单词的信息（要求同词法分析作业，对于预读的情况不能输出）。

2) 在文法中高亮显示（见2020文法定义）的语法分析成分分析结束前，另起一行输出当前语法成分的名字，形如“<常量说明>”（注：未要求输出的语法成分仍需要分析）

【输入形式】 testfile.txt中的符合文法要求的测试程序。

【输出形式】 按如上要求将语法分析结果输出至output.txt中，中文字符的编码格式要求是UTF-8。

【特别提醒】 (1) 本次作业只考核对正确程序的处理，但需要为今后可能出现的错误情况预留接口。

(2) 当前要求的输出只是为了便于评测，完成编译器中无需出现这些信息，请设计为方便打开/关闭这些输出的方案。

【文法定义】 请到“2020年文法定义(用于查看文法, 不计入成绩)”作业中查看文法

#### 【样例输入】

```
const int cOnst1 = 001, const2 = -100;
const char const3 = '_';
int change1;
char change2;
int gets1(int var1, int var2){
    change1 = var1 + var2;
    return (change1);
}
void main(){
    printf("Hello World");
    printf(gets1(10, 20));
}
```

#### 【样例输出】

```
CONSTTK const
INTTK int
IDENFR cOnst1
ASSTGN =
INTCON 001
<无符号整数>
```

学完“第4章 语法分析”开始完成此项作业，学生在理论课所学语法分析方法基础上，设计实现语法分析程序，并按规定格式输出单词信息（沿用词法分析程序的输出）和语法成分名称。

提交内容：语法分析程序源代码

输入输出：为了方便进行自动评测，输入的被编译源文件统一命名为testfile.txt；输出的结果文件统一命名为output.txt；结果文件中包含如下两种信息：

1) 按词法分析识别单词的顺序，按行输出每个单词的信息（要求同词法分

析作业，对于预读的情况不能输出)。

2) 在文法中高亮显示 (见 2020 文法定义) 的语法分析成分分析结束前，另起一行输出当前语法成分的名字，形如“<常量说明>” (注：未要求输出的语法成分仍需要分析)

评判过程：学生提交的语法分析程序传入教学平台，平台将 5 个测试程序依次输入学生的语法分析程序，分别运行得到输出结果，根据输出结果与预期结果一致部分所占的比例给分。

公共测试库：语法分析测试程序库，公布了 60 个测试程序及其语法分析程序输出结果

完成周期：3 周 (第 4-7 周)

#### **(四) 错误处理**

**目标和要求：**针对常见的错误分类编写错误处理程序，考核学生对错误处理方法的掌握情况，培养学生编写错误处理程序的能力。

学生需分析编译过程的每个阶段可能出现的错误，学完“第 8 章 错误处理”后，对各阶段的错误进行错误局部化处理，并进行补充和完善，按要求输出错误信息。

提交内容：含错误处理的语法分析程序源代码

输入输出：为了方便自动评测，输入的被编译源文件统一命名为 testfile.txt；  
错误信息输出到命名为 error.txt 的结果文件中；

结果文件中包含如下两种信息：错误所在的行号 错误的类别码



## 【问题描述】

请根据给定的文法设计并实现错误处理程序，能诊断出常见的语法和语义错误，进行错误局部化处理，并输出错误信息。为了方便自动评测，输入输出及处理要求如下：

- (1) 输入的被编译源文件统一命名为`testfile.txt`；错误信息输出到命名为`error.txt`的结果文件中；
- (2) 结果文件中包含如下两种信息：错误所在的行号 错误的类别码（行号与类别码之间只有一个空格，类别码严格按照表格中的小写英文字母）

其中错误类别码按下表中的定义输出，行号从1开始计数：

错误类型	错误类别码	解释及举例
非法符号或不符词法	a	例如字符与字符串中出现非法的符号，字符串中无任何符号
名字重定义	b	同一个作用域内出现相同的名字（不区分大小写）
未定义的名字	c	引用未定义的名字
函数参数个数不匹配	d	函数调用时实参数大于或小于形参数
函数参数类型不匹配	e	函数调用时形参为整型，实参为字符型；或形参为字符型，实参为整型
条件判断中出现不合法的类型	f	条件判断的左右表达式只能为整型，其中任一表达式为字符型即报错，例如 <code>a==1</code>
无返回值的函数存在不匹配的return语句	g	无返回值的函数中可以没有return语句，也可以有形如return;的语句，若出现了形如return(表达式);或return();的语句均报此错误
有返回值的函数缺少return语句或存在不匹配的return语句	h	例如有返回值的函数无任何返回语句；或有形如return;的语句；或有形如return();的语句；或return语句中表达式类型与返回值类型不一致
数组元素的下标只能是整型表达式	i	数组元素的下标不能是字符型
不能改变常量的值	j	这里的常量指的是声明为const的标识符。例如 <code>const int a=1;</code> 在后续代码中如果出现了修改a值的代码，如给a赋值或用scanf获取a的值，则报错。
应为分号	k	应该出现分号的地方没有分号，例如 <code>int x=1</code> 缺少分号（7种语句末尾，for语句中，常量定义末尾，变量定义末尾）
应为右小括号)	l	应该出现右小括号的地方没有右小括号，例如 <code>fun(a,b;</code> ，缺少右小括号（有/无参数函数定义，主函数，带括号的表达式，if, while, for, switch, 有/无参数函数调用，读、写、return）

评判过程：学生提交的错误处理程序传入教学平台，平台将 10 个包含错误的测试程序依次输入学生的错误处理程序，分别运行得到输出结果，若第一个错误报告正确得 60%的分数，其余错误则按实报错误占应报错误的比例得分。

公共测试库：错误处理测试程序库，公布了 20 个测试程序及其语法分析程序输出结果

完成周期：2 周（第 7-9 周）

## （五）期中考核

**目标和要求：**检查学生对词法分析方法和语法分析方法的掌握情况，要求学生根据文法的变化修改代码，使词法分析程序和语法分析程序符合新的文法规则。

对作业文法的规则进行少量修改，学生需要在规定的时间内，独立完成对前期作业代码的修改，以符合修改后的文法规则。考核统一在机房上机完成。

考核时间：第 8 周

## (六) 代码生成

**目标和要求：**根据具体文法生成指定代码的目标程序，考核学生对语义分析及代码生成方法的掌握情况，培养学生针对给定文法和语义生成指定目标代码的能力。

开始学习“第 10 章 语义分析和代码生成”后，可逐步完成此项作业。学生在理论课所学语义分析方法的基础上，设计实现一个输出指定目标代码的编译器。

目标代码分为 PCODE、MIPS 汇编两种，二者任选其一。若选择生成 PCODE 代码，需同时实现能对该代码进行解释执行的程序，根据解释执行的结果进行考核。若选择生成 MIPS 汇编，建议先生成四元式中间代码，再从中间代码生成 MIPS 汇编代码，并合理利用临时寄存器（临时寄存器池）生成较高质量的目标代码，以获得较高的代码执行效率。根据汇编代码在 Mars 模拟器上的运行结果进行考核。

2020代码生成作业一 / 通用评测题 / 1. 2020代码生成作业一

### 【问题描述】

请在词法分析、语法分析及错误处理作业的基础上，为编译器实现语义分析、代码生成功能。输入输出及处理要求如下：

(1) 需根据语法规则及语义约定，采用自顶向下的语法制导翻译技术，进行语义分析并生成目标代码（任选其一）；

(2) 对于选择生成**MIPS**的编译器，约定如下：

完成编译器，将源文件（统一命名为testfile.txt）编译生成MIPS汇编并输出到文件（统一命名为mips.txt），具体要求包括：

a) 需自行设计四元式中间代码，再从中间代码生成MIPS汇编，请设计实现输出中间代码的有关函数，本次作业不考核，后续会有优化前后中间代码的输出及评判（输出文件命名为学号\_姓名\_优化前/后中间代码.txt）。

b) 若选择此项任务，后续的作业需参加竞速排序，请提前预留代码优化有关的接口，并设计方便切换开启/关闭优化的模式

c) 自行调试时，可使用Mars仿真器（使用方法见“Mars仿真器使用说明.docx”），提交到平台的编译器只需要能按统一的要求生成MIPS汇编代码文件即可

d) **此类提交的编译器请仅读取testfile.txt文件并生成相应的MIPS代码，编译器自身不要读入标准输入中的内容。**

(3) 对于选择生成**PCODE**的编译器，约定如下：

完成编译器和解释执行程序，将源文件（统一命名为testfile.txt）编译生成PCODE目标代码并解释执行，得到解释执行的结果（以输出至“pcoderesult.txt”文件中的形式给出），具体要求包括：

a) 需自行设计PCODE，可参考PASCAL-S编译器的处理

b) 将生成的PCODE直接进行解释执行，在提交的作业中不需要输出PCODE

c) 执行PCODE时，输入从标准输入中读取；输出至pcoderesult.txt文件中

【输入形式】 testfile.txt为符合文法要求的测试程序，另外可能存在来自于标准输入的数据。

【输出形式】 按照选择的不同难度分为两类：

1) 生成MIPS的编译器

按如上要求将目标代码生成结果输出至mips.txt中，中文字符的编码格式要求是UTF-8。

代码生成作业的考核分两次进行，第一次用一个简单的测试程序进行考核，

第二次用 10 个测试程序进行考核。

提交内容：能生成目标代码的编译器源代码

输入输出：对于选择生成 MIPS 的编译器，将源文件（统一命名为 testfile.txt）

编译生成 MIPS 汇编并输出到文件（统一命名为 mips.txt）；

对于选择生成 PCODE 的编译器，完成编译器和解释执行程序，将源文件（统一命名为 testfile.txt）编译生成 PCODE 目标代码并解释执行，得到解释执行的结果（以输出至"pcoderesult.txt"文件中的形式给出），执行 PCODE 时，输入从标准输入中读取；输出至 pcoderesult.txt 文件中

评判过程：作业提交至教学平台，运行学生的编译器代码，输入测试程序，得到目标程序；分别在解释执行程序上运行目标代码或在 Mars 平台上运行产生的 MIPS 汇编代码，得到运行结果，按运行结果与预期结果不一致的行数扣分，每项扣 10%。

公共测试库：代码生成测试程序库，公布了 60 个测试程序及其目标代码的运行结果

完成周期：第一次作业 2 周（第 9-11 周）

第二次作业 4 周（第 11-15 周）

## （七）代码优化

**目标和要求：**在编译器产生的四元式中间代码基础上完成几种优化算法。考核学生对代码优化算法的掌握情况，培养学生开发具有代码优化功能的编译器的能力。

开始学习“第 11 章 代码优化”后，可逐步完成此项作业。选择生成 PCODE 代码的同学无需完成此项作业。

(1) 完成如下代码优化算法：基本块内部的公共子表达式删除（DAG 图）；全局寄存器分配（引用计数或着色算法）；数据流分析（通过活跃变量分析，或利用定义-使用链建网等方法建立冲突图）；还可自选其他优化算法。优化后的 MIPS 代码用 1-3 个测试程序进行测试，获取在 Mars 模拟器上的运行结果，与预期结果进行自动比对，在运行结果正确的基础上参加竞速排序

(2) 竞速排序：根据测试程序在学生提交的优化编译器上产生的指令统计数据数据进行竞速排序，根据排序先后给分。

提交内容：具有代码优化功能的编译器源代码

输入输出：将源文件（统一命名为 testfile.txt）编译生成 MIPS 汇编并输出到文件中（统一命名为 mips.txt）

评判过程：作业提交至教学平台，运行学生的编译器代码，输入测试程序，得到 MIPS 汇编代码，在 Mars 平台上运行汇编代码，得到运行结果，在运行结果正确的基础上，根据  $FinalCycle = ALU * 1 + Jump * 2 + Branch * 2 + Memory * 2 + Other * 1$  的值排名，FinalCycle 越小排名越靠前。

排行榜

#	学号	姓名	Branch	Memory	ALU	Other	Jump	rank
1	16191051	朱雨聪	12708	329	61743	162	18	88015.00
2	17373028	尹子鑫	20257	566	63874	162	14	105710.00
3	17373331	何岱岚	24466	549	57721	163	15	107944.00
4	17231181	唐嘉明	25516	555	57619	162	272	110467.00
5	17373423	王少布	25515	550	57754	162	268	110582.00
6	17373321	游子诺	25516	4215	59682	163	12	119331.00
7	17373381	何逸宸	25516	580	80871	162	279	133783.00
8	17373380	蝶天香	25516	570	81581	162	275	134465.00
9	17373323	罗钧宇	25516	700	81938	162	11	134554.00
10	17231173	王濛汀	25513	562	82316	163	12	134653.00
11	17373457	叶苏鹏	25516	832	82996	163	15	135885.00
12	17373451	赵博	25516	589	83928	162	13	136326.00
13	17373552	莫策	25516	550	84776	163	14	137099.00
14	17376108	王慎执	25516	2601	83062	154	4	139458.00
15	17373452	单彦博	25516	548	97465	154	12	149771.00
16	17373434	伦泽标	25516	554	73085	163	12374	150136.00

公共测试库：代码优化测试程序库，公布了 20 个测试程序及其目标代码的

运行结果

完成周期: 4 周 (第 12-16 周)

### (八) 期末考核

**目标和要求:** 检查学生对语义分析、代码生成、代码优化方法的掌握情况, 要求学生根据文法的变化修改代码, 使编译器符合新的文法规则。

(1) 对作业文法的规则进行少量修改, 学生需要在规定的时间内, 独立完成对前期作业代码的修改, 以符合修改后的文法规则, 根据目标代码运行结果评分。

(2) 竞速排序: 根据测试程序在学生提交的优化编译器上产生的指令统计数据竞速排序, 根据排序先后给分。选择生成 PCODE 代码的同学无需进行此项考核。

考核统一在机房上机完成。

考核时间: 第 17 周

## 6. 开放性问题

(1) 在生成中间代码时, 引导学生去了解其他形式的中间代码, 比较与四元式的相同和不同之处;

(2) 除了在理论课中学习到的公共子表达式删除、死代码删除、函数内联、常量传播等优化算法之外, 考虑如何把优化的思想应用到其他场景, 比如函数调用、数组操作等;

(3) 对于有多种优化算法时, 如何进行架构设计, 使各种优化算法之间最大程度上解耦, 且方便后续优化算法的实现与纠错。