****

调研报告

|  |  |
| --- | --- |
| 题 目 | 面向C语言的编译错误增 |
|  | 强提示方法研究与实现 |
| 学生姓名 | 罗龙光 |
| 指导教师 | 李洪东 |
| 学院 | 计算机院 |
| 专业班级 | 信息安全1801 |
| 完成时间 | 2022.2.15 |

摘要

在CSUOJ作为数据集的背景下，收集并整理学生C语言代码的编译信息，提取其中的编译错误信息，并按照出现次数、错误类型等条目进行归类整理。对归类的编译信息做出分析，产出调研报告以期对教学实践有所帮助。对现有编译错误信息改进方式作总结并搜集常见的编译错误提示消息。在GCC编译信息提示的基础上，结合所收集的信息，给出一个服务以对C语言的编译错误信息进行增强提示。对于常见的错误，采用正则匹配的方式来定位错误;对于难以依靠提示而定位的错误，采用抽象语法树的方式定位错误并给出修改建议。将服务部署在平台上，前端对编译错误提示消息智能提示，后台进行数据收集。对比服务部署前后的编译错误数据，验证对C语言编译错误增强提示方法的有效性。

**关键词** C语言 编译错误 新手语法错误 调试 增强提示

目录

[第一章 设计任务及背景 1](#_Toc98075318)

[1.1 编译错误增强提示的作用及意义 1](#_Toc98075319)

[1.2 研究背景及现状 2](#_Toc98075320)

[1.2.1 国外在增强提示方面的研究 2](#_Toc98075321)

[1.2.2 国内在增强提示方面的研究 2](#_Toc98075322)

[1.3 相关知识介绍 3](#_Toc98075323)

[第二章 课题分析与设计思路 4](#_Toc98075324)

[2.1 课题目标 4](#_Toc98075325)

[2.2 需求分析 4](#_Toc98075326)

[2.3 概要设计 4](#_Toc98075327)

[2.4 工作进度 5](#_Toc98075328)

[第三章 结束语 6](#_Toc98075329)

[参考文献： 7](#_Toc98075330)

# 第一章 设计任务及背景

## 1.1 编译错误增强提示的作用及意义

计算机编程已被证明给学生带来了许多困难，它导致了学生的低积极性，低成功率。调试是学习编程过程中的一个挑战，特别是，如何学习并理解编译器的报错信息。而增强提示可以对此带来一定的帮助。

1)提高学生的编写效率以及积极性。根据研究显示，85%的学生表示发现编译器错误信息是他们挫折感的来源，92%的学生表示编译器错误信息是他们进步的障碍。64%的人说发现多个编译错误信息会让他感到困惑。某些错误难以通过编译器的所给出信息而准确定位，这需要我们在给初学者的所看到的错误信息进行增强提示。

2)提高教师教学效率。对于很多初学者来说，经常遇到编译错误，运行错误等问题而向老师求教，但是由于错误类型过多，加之部分错误较为隐秘且不常见，这给老师快速定位错误带来困难。对于编译错误的增强提示可以直接跳过阅读冗杂的代码，使教师直接看到代码的错误原因，从而快速准确的定位到出错位置。

## 1.2 研究背景及现状

编译错误提示信息作为编译器的重要组成部分，是对程序编译错误与程序员交互的一个有效手段，但由于其对编程初学者来说往往难以理解，而且经常还有可能产生对错误位置的错误定位，这些问题对初学者来说是很大的障碍。

### 1.2.1 国外在增强提示方面的研究

国外对于这方面的研究较为丰富，国外十分重视初学者的代码实践，它们做出了很多理论上的研究。

文章：爱尔兰都柏林大学团队对于编译错误集合进行了梳理，它们发现了多数情况下一个错误对应着多条编译器错误信息，其中一些错误信息是无帮助的，甚至会产生混淆。它们基于这种情况对错误信息进行了归类分析。

文章：在文章 的基础上，他们提出了同时处理多条编译器错误信息的方法，入了一个新的分类法，使研究人员和教育工作者有可能更准确地谈论这些编译错误。定义每次编译失败的第一条信息被标记为 "第一条信息"。 理论上，随后的信息可以分为三种类型：“级联信息”（虚假的错误信息，当真正的错误被修复后就会消失）；“真正的第一条信息”（对应于一个明显的错误，在编译中成为第一条信息）；以及 “被埋藏信息”（对应于一个明显的错误，但在编译后未成为第一条信息）。只考虑第一条错误信息，从全局来看是更为有效的。

文章：他们对于增强提示后的效果做出了研究，验证了增强提示的有效性， 实质性的对于调试做出了帮助。

文章、文章：对于java常见的新手错误进行了收集，对常见的错误以及教育者如何利用这些信息做出了研究，为如何将增强提示应用到新手教学中提出了指导意见。

### 1.2.2 国内在增强提示方面的研究

文章：从底层阐述了编译错误的诞生，并描述到了一个编译错误诞生的生命周期，这对于对某些相对复杂的编译错误做出个性化响应有所帮助。

字节跳动的morden.js中集成了对于node代码的编写错误的自动提示以及代码补全,这类似于Visual Studio Code上面向各种语言的代码补全插件，通过对于抽象语法树的分析发现编译错误，并通过自定义逻辑对于这些错误进行提示与补全。

## 1.3 相关知识介绍

**GCC**

原名为GNU C语言编译器（GNU C Compiler），只能处理C语言。但其很快扩展，变得可处理C++，后来又扩展为能够支持更多编程语言

GCC（GNU Compiler Collection，GNU编译器套件）是由GNU开发的编程语言译器。GNU编译器套件包括C、C++、 Objective-C、 Fortran、Java、Ada和Go语言前端，也包括了这些语言的库（如libstdc++，libgcj等。）

**抽象语法树**

在计算机科学中，抽象语法树（Abstract Syntax Tree，AST），或简称语法树（Syntax tree），是源代码语法结构的一种抽象表示。它以树状的形式表现编程语言的语法结构，树上的每个节点都表示源代码中的一种结构。

**级联错误**

级联式错误信息是由一个与先前的错误信息相对应的错误引起的虚假信息，并没有提供额外的（有用的）信息。简而言之，这些是编译器混乱的结果。这个定义与Schorsch的描述一致。“由一个错误引起的多个错误报告，都可以通过修复第一个报告的错误而得到纠正。”

# 第二章 课题分析与设计思路

## 2.1 课题目标

1. 查阅资料，了解编译错误信息的研究历程和最新研究进展；
2. 对现有编译错误信息改进方式作总结并搜集常见的编译错误提示消息；
3. 构建平台，要求前端能对编译错误提示消息智能提示，后台能展示所有编译数据并归类展示。
4. 对设计平台进行实际使用分析，发现其存在的问题，分析并提出下一步工作展望。

## 2.2 需求分析

首先需要从数据库中获得编译错误信息。获取信息后进行分析，主要分为两个方面，第一是错误频率分析，统计不同错误的出现频率；第二是级联错误分析，一次编译中可能产生了多条错误信息，尝试提取与错误最相关的那一条。最后可以做个对照分析，只考虑每次编译的第一条错误信息和考虑所有的错误信息会在频率上面有什么不同。

做完数据处理之后需要对于错误人工添加增强提示。第一种是较为简单的错误，直接对提示进行翻译；第二种是复杂错误，复杂错误的有几种类型:一个错误对应着多种导致错误的原因，需要一一列举；一个错误是级联式错误，对于调试没有直接帮助，需要具体错误具体分析，并给出增强提示。

然后需要编写服务，从上游接受一次编译所给出的信息，返回增强提示。对于只有一条错误的情况下，直接返回之前人工做好的增强提示即可。对于多条错误，需要按例分析，编写多个逻辑,处理级联错误、具有混淆性的错误等多种情况。

最后将服务部署于CSUOJ后台，改写前端页面，以在前端对学生编译错误做出增强提示。

## 2.3 概要设计

本次设计主要是编写一个服务，对于编译错误增强提示响应。大致上可以分为三个阶段：数据提取，服务编写，环境部署。

数据提取： 利用Navicat Premium 16辅助sql查询，从表solution、compileinfo中利用正则匹配提取编译错误，从表source\_code\_user中提取每种编译错误所对应的源码样例。将错误归类，制作高频错误表单，利用正则匹配将同类错误收敛，制作按照出现频率排序去重的编译错误表单。

服务编写：人工完成简单错误的提示并直接响应，对于复杂且高频的错误编写多个子服务以做出相应。再加入中间层进行A/B实验，统计增强提示与否对于学生代码产生编译错误的影响，以及验证某个具体的增强提示的效果。

环境部署： 需要对于CSUOJ的前端页面进行改写。在提交界面增加富文本编辑框以展示增强提示信息。需要将本地编写的node服务与生产环境进行兼容，搭建中间层来保证服务的可拓展性。

## 2.4 工作进度

|  |  |  |
| --- | --- | --- |
| 阶段 | 工作任务内容 | 起止时间 |
| 一 | 查阅资料、搭建开发环境、完成翻译和调研报告 | 第1~4周 |
| 二 | 确定系统实现方案 | 第5~6周 |
| 三 | 进行编码、测试及迭代 | 第7~12周 |
| 四 | 完成毕业论文，准备答辩 | 第13~14周 |
| 五 | 论文答辩及完善 | 第15~16周 |

# 第三章 结束语

随着计算机科学的发展，对于编译信息的分析与理解越来越重要。如何让初学者能更快的理解这些信息也就成为了重中之重。编译结果的形象化表示，将大大提高初学者调试程序的效率。如何做出增强提示在业内已有了很多的研究，针对我校样本的数据集，我们可以在之前研究的基础上做出本地化的结果，从而帮助教学工作的开展，在提效方面的成果值得期待。加之计算机技术的快速发展，在本次研究的基础上，未来还可以引入机器学习的方式，对于某个具体同学或者某种具体错误来做出针对的响应。我很荣幸能参加本次课题的设计，这对于我来说是一个很好的磨练，期望我能够顺利的完成这次的毕业设计。

# 参考文献：

[1] http://vlab.csu.edu.cn/oj/

[2] Becker B A, Denny P, Pettit R, et al. Compiler error messages considered unhelpful: The landscape of text-based programming error message research[M]//Proceedings of the working group reports on innovation and technology in computer science education.

[3] 高云云. C语言编译系统的研究与实现[D].南京邮电大学,2019.

[4] Becker B A, Murray C, Tao T, et al. Fix the first, ignore the rest: Dealing with multiple compiler error messages[C]//Proceedings of the 49th ACM technical symposium on computer science education. 2018: 634-639.

[5] Becker B A, Goslin K, Glanville G. The effects of enhanced compiler error messages on a syntax error debugging test[C]//Proceedings of the 49th ACM Technical Symposium on Computer Science Education. 2018: 640-645.

[6] Becker B A, Goslin K, Glanville G. The effects of enhanced compiler error messages on a syntax error debugging test[C]//Proceedings of the 49th ACM Technical Symposium on Computer Science Education. 2018: 640-645.

[7] Becker B A, Goslin K, Glanville G. The effects of enhanced compiler error messages on a syntax error debugging test[C]//Proceedings of the 49th ACM Technical Symposium on Computer Science Education. 2018: 640-645.