

Compiler Principle

Text Book: Modern Compiler Implementation in C

Author: Andrew W. Appel

2023 Spring&Summer

实验安排 (Laboratory)

- 个人实验(Individual Experiment):
lab1或者lab2
lab3或者lab4
- 小组实验(Group Experiment):
deadline 夏学期第5周的周日5月28日22:00
小组作业会进行查重
如果有抄袭, 分数全为0
如果任何贡献都没有, 为0

Expriement Evironment

Linux环境下的编译和运行

- Linux 2.6以上版本
- GCC3.4以上版本
- Bison 2.2以上版本
- Flex 2.5.33以上版本
- 发行版可以采用Ubuntu, Gentoo, Fedora Core等。

Expriement Evironment

- Windows环境下的编译和运行
 - Visual Studio 6.0
 - Masm 6.0以上版本
 - ParseGenerator 4.0 (Lex和Yacc的集成开发包)

Task One

利用**LEX**计算文本文件的字符数等 **(5分)**

- 实验目的：了解**LEX**的基本编程方法。
- 实验要求：编写一个**LEX**输入文件，使之生成可计算文本文件的字符、单词和行数且能报告这些数字的程序。单词为不带标点或空格的字母和/数字的序列。标点和空白格不计算为单词。

Task Two

利用**LEX**进行字母的大小写转换 (5分)

- 实验目的：了解**LEX**的基本编程方法
- 实验要求：编写一个**LEX**输入文件，使之可生成将**SPL**程序注释之外的所有关键字（保留字）均大写的程序。有关**SPL**的关键字请见第二章或第八章所述。该**LEX**生成的程序要能够对**SPL**源程序进行分析，将不是大写的关键字均转换为大写。

Task Three

利用YACC生成中缀表示的计算器 (10分)

- 实验目的：了解YACC处理二义性的方法。
- 实验要求：生成如下文法表示的表达式对应的计算器
 - $\text{exp} \rightarrow \text{exp} + \text{exp} \mid \text{exp} - \text{exp}$
 - $\mid \text{exp} * \text{exp} \mid \text{exp} / \text{exp}$
 - $\mid \text{exp} ^ \text{exp} \mid - \text{exp}$
 - $\mid (\text{exp}) \mid \text{NUM}$
 - 对于输入的中缀表达式，要给出结果。如 $3 + (4 * 5)$ 结果应为23。要求能连续处理若干个数学表达式，直到输入结束或文件结束

Task Four

- 利用**YACC**生成能进行整数和实数运算的计算器 (10分)
- 实验目的: 了解YACC属性处理的基本方法。
- 实验要求: 生成如下文法表示的表达式对应的计算器
- $\text{exp} \rightarrow \text{exp} + \text{exp} \mid \text{exp} - \text{exp}$
- $\mid \text{exp} * \text{exp} \mid \text{exp} / \text{exp}$
- $\mid \text{exp} ^ \text{exp} \mid - \text{exp}$
- $\mid (\text{exp}) \mid \text{NUM}$

Task Four

实验要求:

- 对于输入的中缀表达式, 要给出结果。举例说明:
- $3 + (4 * 5) = 23$
- $3 + (4.2 * 2) = 11.4$
- $3.2 + (1/2) = 3.7$
- $3 + (1/2) = 3$
- 要求能连续处理若干个数学表达式, 直到输入结束或文件结束。

Project

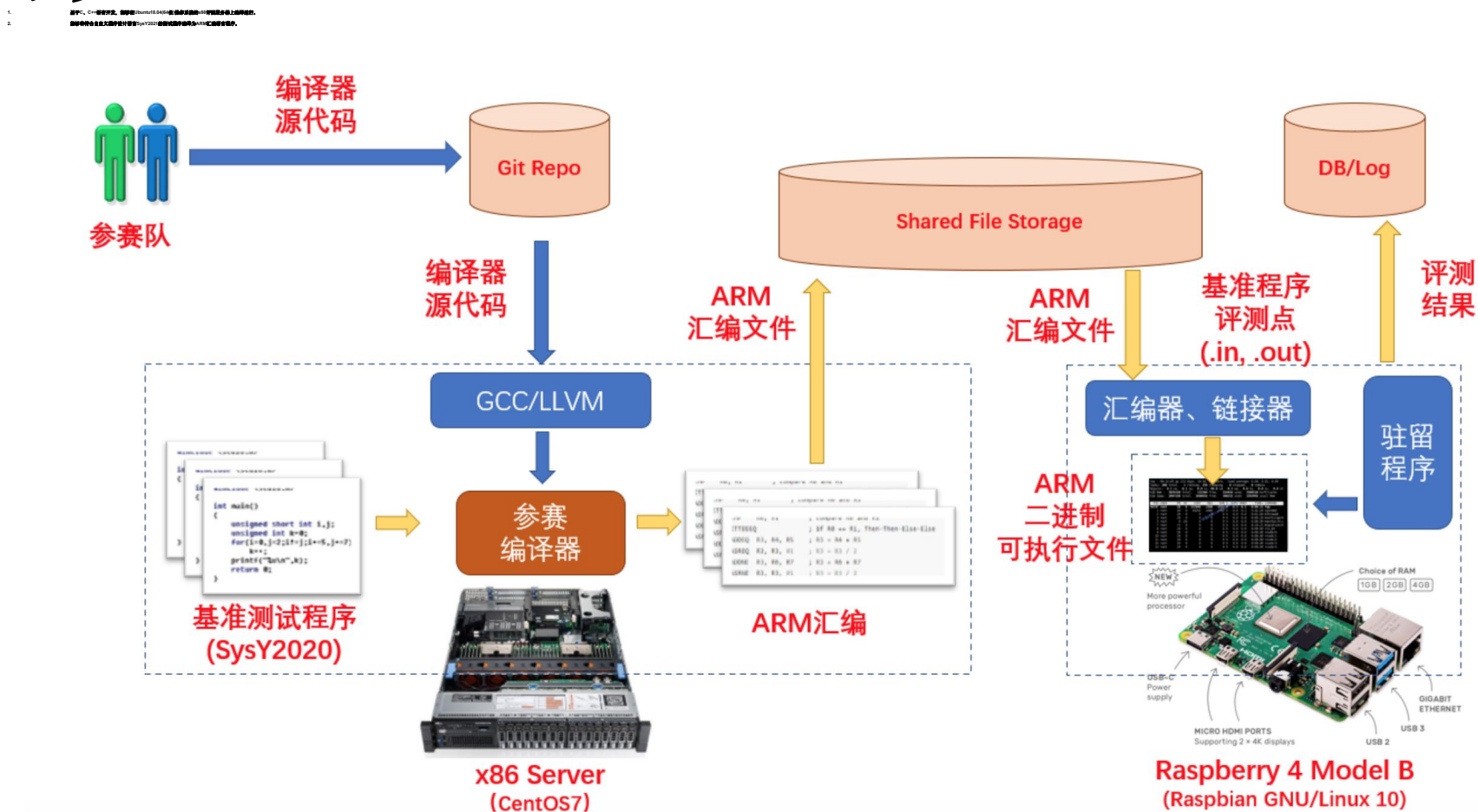
- 实现某语言的编译器；用**Lex**开发该语言的词法分析器，用**YACC**开发该语言的语法分析，并生成语法树和中间代码，生成目标代码（**ARM**或**RISC-V**）。（25分）
- 实验目的：掌握词法分析、语法分析和代码生成方法。

Project

- 实验要求:
- 1、提交实验报告
包括词法分析部分的正规表达式描述、数据结构、原理以及实现，语法分析部分的文法描述、数据结构、原理以及实现，中间代码的格式、数据结构描述以及中间代码生成、目标代码生成的实现。
- 2、提交源代码
- 3、要求结果能用可视的方法表示出来（图或者采用数据结构中学过的方法）

Project

内容：开发支持特定语言、面向**ARM**硬件平台的综合性编译系统



Project

语言: SysY

SysY 语言是本次大赛要实现的编程语言, 是 C 语言的一个子集。每个 SysY 程序的源码存储在一个扩展名为 `sy` 的文件中。该文件中有且仅有一个名为 `main` 的主函数定义, 还可以包含若干全局变量声明、常量声明和其他函数定义。SysY 语言支持 **int** 类型和元素为 `int` 类型且按行优先存储的多维数组类型, 其中 `int` 型整数为 32 位有符号数; **const** 修饰符用于声明常量。

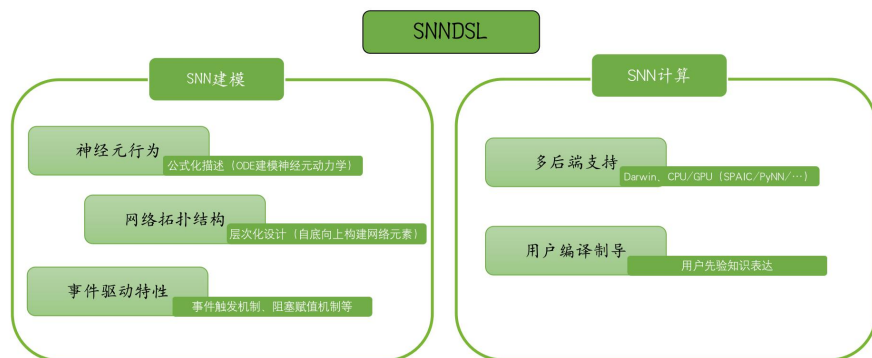
SysY 语言本身没有提供输入/输出(I/O)的语言构造, I/O 是以运行时库方式提供, 库函数可以在 SysY 程序中的函数内调用。部分 SysY 运行时库函数的参数类型会超出 SysY 支持的数据类型, 如可以为字符串。SysY 编译器需要能处理这种情况, 将 SysY 程序中这样的参数正确地传递给 SysY 运行时库。有关在 SysY 程序中可以使用哪些库函数, 请参见 SysY 运行时库文档。

Project

语言：神经元计算机编程语言SNNDSL

背景：将神经网络与逻辑推理统一到编程语言理论中，建立了同时具备神经网络和专家系统特性、融合连接机制和符号机制的智能神经元模型，实现神经计算、逻辑推理和数值计算的统一。

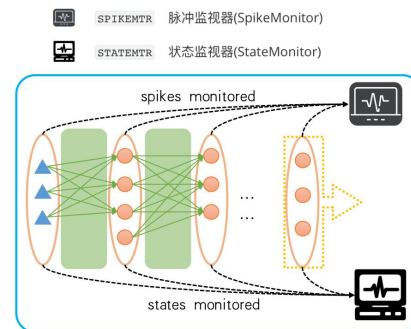
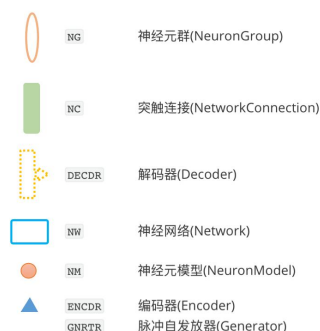
神经元编程语言 – 设计思路



设计思路：面向脉冲神经网络 **建模与计算** 的领域特定语言(DSL)

神经元编程语言 – 文法设计

• 神经网络元素



The end