

新概念C语言

NCCL – New Concept C Language

@亚嵌李明老师

limingth@gmail.com

2012-12-21

■ <https://github.com/limingth/NCCL/>

Outline

- C 语言 Language
 - 高级语言 High-level Programming Language
 - 汇编语言 Assembly Language
 - 机器指令 Instructions
- 编译器 Compiler
 - 汇编器 Assembler
- 操作系统 OS
 - 加载器 Loader
 - 加载地址和执行地址 LoadAddr vs ExecAddr
- 程序的执行 Program Execution
 - 进程的概念 Process
 - 执行流程 Execution Sequence

Lesson 1 What is a simplest C program?

最简单的C程序

```
int main(void)
{
    return 0;
}
```

知识点

- 数据类型 `int`
- 函数名 `main`
- 参数(列表) `void`
- 返回值 `return`

C语言 BNF 范式分析

```
int main(void)
{
    return 0;
}
```

类型声明 声明符
int main(void)

复合语句: { 语句 }
{ return 0; }

-----> translation_unit 翻译单元

C语言 BNF 范式分析

translation_unit : external_decl
翻译单元 外部声明

external_decl : function_definition
外部声明 函数定义

function_definition : decl_specs declarator compound_stat
函数定义 声明说明符 声明符 复合语句

decl_specs : type_spec
声明说明符 类型说明符

type_spec : void | char | short | int | long | float
类型说明符

C语言 BNF 范式分析

declarator : direct_declarator
声明符 直接声明符

direct_declarator : id 标识符
直接声明符 | (declarator) 声明符
| direct_declarator [const_exp] 直接声明符 [常量表]
| direct_declarator [] 直接声明符 []
| direct_declarator (param_type_list) 直接声明符 (形式参数表)
| direct_declarator (id_list) 直接声明符 (标识符表)
| direct_declarator () 直接声明符 ()

-----> main(param_type_list) 函数声明符

C语言 BNF 范式分析

param_type_list : param_list 形式参数表
形式参数类型表 | param_list , ... 形式参数表, ...
;

param_list : param_decl 形式参数声明
形式参数表 | param_list , param_decl 形式参数表, 形式参数声明

param_decl : decl_specs declarator 声明说明符 声明符
形式参数声明 | decl_specs abstract_declarator 声明说明符 抽象声明符
| decl_specs 声明说明符
;

decl_specs : storage_class_spec decl_specs 存储类说明符 声明说明符
声明说明符 | storage_class_spec 存储类说明符

C语言 BNF 范式分析

`compound_stat` : { `decl_list` `stat_list` } { 声明表 语句表 }
复合语句

-----> { `return 0;` } 复合语句

C语言 BNF 范式分析

stat_list : stat 语句
语句表 | stat_list stat 语句表 语句

stat : labeled_stat 带标号语句
语句 | exp_stat 表达式语句
| compound_stat 复合语句
| selection_stat 选择语句
| iteration_stat 循环语句
| jump_stat 跳转语句

jump_stat : goto id ; goto 标识符;
跳转语句 | continue ; continue;
| break ; break;
| return exp ; return 表达式;

扩展

```
/* this is a simplest c program */  
  
int global = 1;  
  
int main(void)  
{  
    int local = 2;  
  
    // we return these two variable summary  
    return local + global;  
}
```

知识点

- 变量 variable
 - 局部变量 local
 - 全局变量 global
- 运算符 operator
 - 双目运算符 +, -, *, /
 - 赋值运算符 =
- 注释 comment
 - 不能嵌套 /* */
 - 能嵌套 //
- 基本数据类型
 - 字符 char
 - 短整型 short
 - 浮点 float
 - 双精度浮点 double

课堂讨论

- `main` 函数名是 C 语言的关键字吗？
- 注释可以写在某一行代码的里面吗？
- 全局变量和局部变量取名可以重名吗？
- 所有代码可以写在一行里面吗？
- 把源程序中的空格去掉可以吗？

课后练习



修改代码，使得编译不通过，并举出常见的错误提示和出错原因，

参考资料

■ B语言 C语言的前身

- B语言 <http://zh.wikipedia.org/wiki/B语言>
- THE PROGRAMMING LANGUAGE B
<http://cm.bell-labs.com/cm/cs/who/dmr/bintro.html>
- A TUTORIAL INTRODUCTION TO THE LANGUAGE B
<http://cm.bell-labs.com/cm/cs/who/dmr/btut.html>
- Users' Reference to B
<http://cm.bell-labs.com/cm/cs/who/dmr/kbman.html>

■ C语言

- C语言 <http://zh.wikipedia.org/wiki/C语言>
- C语言之父 Dennis Ritchie
<http://zh.wikipedia.org/wiki/丹尼斯·里奇>
- C语言 BNF 范式
http://www.cs.man.ac.uk/~pjj/bnf/c_syntax.bnf

参考资料

■ BNF 范式

- <http://www.cs.man.ac.uk/~pjj/bnf/bnf.html>
- <http://zh.wikipedia.org/wiki/巴科斯范式>

■ glibc 库

- <http://zh.wikipedia.org/wiki/Glibc>
- <http://ftp.gnu.org/gnu/glibc>
- <http://www.oschina.net/code/explore/glibc-2.9>

■ gcc 编译器

- <http://ftp.gnu.org/gnu/gcc/>
- <http://www.oschina.net/code/explore/gcc-4.5.2>

名人名言

- Dennis Ritchie (C语言之父)
 - “Many of the improvements I introduced when developing C simply looked like a good thing to do.”