

C 语言概述与云环境部署

本项目通过具体的示例介绍，使学习者对计算机与程序设计语言、C 语言的发展和特点、C 语言的基本结构、C 程序的工作原理等方面有深入的认识；并通过具体案例，使学习者掌握，在腾讯云中搭建 C 语言的开发环境。

课堂学习目标

- 计算机与程序设计语言
- C 语言的发展和特点
- C 语言的基本结构
- C 程序的工作原理
- 腾讯云 C 语言开发环境搭建



任务 1-1 计算机与程序设计语言

任务要求

通过本任务深入了解计算机和程序设计语言的发展与状况；并根据各个发展阶段的突出特点。掌握计算机与程序设计语言之间的关系。

相关知识

(一) 计算机

计算机（computer）俗称电脑，是现代一种用于高速计算的电子计算机器，可以进行数值计算，又可以进行逻辑计算，还具有存储记忆功能。是能够按照程序运行，自动、高速处理海量数据的现代化智能电子设备。

计算机的发展分四代

- 第 1 代：电子管数字机（1946—1958 年）

硬件方面，逻辑元件采用的是真空电子管，主存储器采用汞延迟线、阴极射线示波管静电存储器、磁鼓、磁芯；外存储器采用的是磁带。

- 第 2 代：晶体管数字机（1958—1964 年）

软件方面的操作系统、高级语言及其编译程序应用领域以科学计算和事务处理为主，并开始进入工业控制领域。

- 第 3 代：集成电路数字机（1964—1970 年）

硬件方面，逻辑元件采用中、小规模集成电路（MSI、SSI），主存储器仍采用磁芯。软件方面出现了分时操作系统以及结构化、规模化程序设计方法。

- 第 4 代：大规模集成电路计算机（1970 年至今）

硬件方面，逻辑元件采用大规模和超大规模集成电路（LSI 和 VLSI）。软件方面出现了数据库管理系统、网络管理系统和面向对象语言等。

(二) 程序设计语言

计算机编程语言是程序设计的最重要的工具，它是指计算机能够接受和处理的、具有一定语法规则的语言。

根据程序设计语言的发展历程，可以分为四代

- 第一代机器语言

机器语言是由二进制 0、1 代码指令构成，不同的 CPU 具有不同的指令系统。

- 第二代汇编语言

汇编语言指令是机器指令的符号化，与机器指令存在着直接的对应关系，所以汇编语言同样存在着难学难用、容易出错、维护困难等缺点。

- 第三代高级语言

高级语言是面向用户的、基本上独立于计算机种类和结构的语言。

- 第四代非过程化语言
非过程化语言，编码时只需说明“做什么”，不需描述算法细节。

(三) 计算机与程序设计语言

根据计算机、程序设计语言的发展过程，和几代程序设计语言的区别；不难看出，计算机只能识别和执行 0、1 组成的二进制指令的机器语言；并不能直接识别和执行，用高级语言编写出的代码源程序。

为了让计算机能识别与执行高级语言编写的源程序代码；必须要经过转换，把高级语言编写出的源程序，“翻译”成计算机能直接执行的机器语言。

在计算机中，把高级语言“翻译”成机器语言的过程成为编译。

任务 1-2 C 语言的发展和特点

任务要求

通过本任务了解 C 语言的发展历史，并掌握 C 语言的特点与优势。

相关知识

(一) C 语言的发展

从程序语言的发展过程看到，操作系统等系统软件，主要是用汇编语言编写，但由于汇编语言依赖于计算机硬件，程序的可读性和可移植不佳。为了提高程序语言的可读性和可移植性；人们不断寻找一种语言，这种语言应具有高级语言的易读特性，又有低级语言运行速度快的特点。因此，C 语言便应运而生了。

C 语言是在 BCPL 语言（简称 B 语言）的基础上发展和完善起来的，而 B 语言是由 UNIX 的研制者丹尼斯·里奇（Dennis Ritchie）和肯·汤普逊（Ken Thompson）于 1970 年研制出来的。20 世纪 70 年代初期，AT&T Bell 实验室的程序员丹尼斯·里奇第一次把 B 语言改为 C 语言。

(二) C 语言的特点

C 语言是一门面向过程的计算机编程语言，C 语言的设计目标是提供一种能以简易的方式编译、处理低级存储器、仅产生少量的机器码以及不需要任何运行环境支持便能运行的编程语言。同时具有高级语言和汇编语言的优点。C 语言可以广泛应用于不同的操作系统，如 UNIX、MS-DOS、Microsoft Windows 及 Linux 等。

在当前常用的编程语言中，C 语言是使用时间最长的一种语言类型，也是使用较为广泛的一种通用语言。

C 语言之所以在软件开发行业中具有强大生命力，主要是因其具有以下三个特点。

- 良好的使用性能。

在软件编程中，C 语言具有良好的使用功能，如语言简短、通用性强等，因此适用于这类计算机系统使用。

- 计算机编程的基础语言。

作为一种使用时间最长的编程语言，C 语言在编程中较之其他高级语言与计算机系统更加接近，因此其也成为了软件编程的基础。在其他语言编程遇到困难的情况下，利用 C 语言编程可以解决问题，实现良好的编程过程。

- 具有一定的专业性优势。

当前 C 语言在部分专用计算机领域，如工业单片机控制程序的编制中，具有一定的专业优势。这种技术优势的存在，对于 C 语言发展起到了重要的促进作用。

任务 1-3 C 语言的基本结构

任务要求

通过本任务具体的应用案例，理解与掌握 C 语言程序的基本结构各部分的具体作用。

相关知识

C 语言的基本结构

1. 文件包含

`#include <stdio.h>` 语句是文件包含，通用的语法方式是：`#include <文件名>`，是属于一种预处理命令。

文件包含的作用，是将程序编译时所需要的文件，复制到本文件，再对合并后的文件进行编译。

【实训 1-3-1】中所使用的输入/输出函数 `printf()`、`scanf()`，需要在源程序的开头，写上 `#include <stdio.h>` 语句

【实训 1-3-2】，若编写程序时，用到数学函数，如：`sin()`、`cos()`、`abs()`时，则在文件头部，必须包含 `#include <math.h>` 语句

2. 主函数

`main()`为程序的主函数，是系统提供的特殊函数，每个 C 语言程序，有且只有一个 `main()`函数。函数的内部执行程序（称为函数体），用一对大括号括起来。

3. 变量定义

一个变量在内存里，占据存储单元，并在这存储单元中，存放变量的值。

变量命名必须符合的规则包括：标识符智能用（大、小写）字母、数字和下划线三种符号组成。

C 语言对大小写有严格的区分，一般变量以小写形式呈现。

4. 格式输入与输出函数

C 语言的输入/输出，是由函数实现。在程序直接调用输入/输出函数前，务必先加载 C 标准函数库。如 `printf()`函数和 `scanf()`函数的调用前，先用 `#include <stdio.h>` 加载标准库。

■ 输入函数的作用

将输入设备（如：键盘）按指定格式输入的数据，赋值给指定的变量存储单元，作为变量值。

■ 输出函数的作用

向系统指定的输出设备（如：显示器），输出若干任意类型的数据。

5. 注释部分

注释是程序员在编程时,对程序的某部分功能和作用,所做的标注说明,以便程序检查是,方便阅读程序结构。

注释是以“/*”开头,至“*/”结尾之间的内容,作为注释内容。

任务实现

【实训 1-3-1】C 语言结构分析一:

(一) 代码:

```
#include <stdio.h>           /*1、文件包含*/

int main()                   /*2、主函数*/
{                             /*3、程序开始*/
    int x,y,total;           /*4、变量定义*/
    scanf("%d",&x);          /*5、格式输入函数*/
    scanf("%d",&y);          /*6、格式输入函数*/
    total=x+y; /*对a、b求和*/ /*7、求和与注释*/
    printf("total=%d \n",total); /*8、格式输出函数*/
}                             /*9、程序结束*/
```

(二) 运行效果

total=32766

【实训 1-3-2】C 语言结构分析二：**(三) 代码：**

```
#include <stdio.h>           /*1、文件包含*/
#include <math.h>           /*1、文件包含*/

int main()                  /*2、主函数*/
{                            /*3、程序开始*/
    float x, y;             /*4、变量定义*/
    x=4;                   /*5、输入x值*/
    x=fabs(x);              /*保证被开方根为正数*/
    y=sqrt(x);              /*对x进行开平方根*/
    printf("%f", y);       /*8、格式输出函数*/
}                            /*9、程序结束*/
```

(四) 运行效果

2.000000

任务 1-4 C 程序的工作原理

任务要求

通过本任务中 C 语言程序的具体特点, 掌握源程序的编辑、编译、链接目标代码、运行可执行程序的全过程。

相关知识

(一) C 程序的特点

根据程序设计语言的发展历程, C 程序语言属于“第三代高级语言”; 因此要让计算机能识别与执行 C 语言程序, 务必要经过编译过程; 把 C 语言编写的源程序代码, 转换为计算机能直接执行(由 0、1 符号组成)的机器语言。

C 语言的工作原理与步骤, 主要经过源程序的创建、编译源程序、链接目标代码、运行可执行文件的过程。

(二) C 程序的工作原理

1、创建与编辑源程序

用文本编辑器或 C 语言的集成开发环境 (IDE), 把设计与编写好的 C 语言程序代码, 录入到计算机中, 保存为 C 语言的源程序文件(文件名自定义, 但扩展名必须为“.cpp”)。此源程序文件的格式是文本文件。

例如: C 源程序文件为“products.cpp”

2、源程序的编译

源程序的编译过程, 就是把已编辑好的高级语言源程序代码, 翻译成计算机执行的二进制目标代码。

计算机系统对源程序编译的同时, 还会对源程序的语法结构进行检查。当发现错误时, 会在屏幕上显示错误的位置与种类, 以便后期用编辑工具, 对高级语言的源代码进行排错与修正。

若无错误的高级语言源程序被编译后, 就会产生一个与源程序文件名同名, 但扩展名为“.obj”的目标程序文件。

例如: 编译源程序文件“products.cpp”后, 产生目标程序为“products.obj”

3、链接目标代码

编译后所产生的目标文件, 并不能在计算机中直接运行。因编译时, 每个模块都是单独编译的, 若要计算机能运行程序, 还需要把每个模块编译后的目标程序, 与计算机系统提供的标准库函数链接。使用系统提供的“链接程序”把目标程序链接后, 就产生扩展名为“.exe”的可执行目标程序。

例如: 编译的源程序文件“products.cpp”, 经过连接后, 生成的可执行目标程

序为“products.exe”

4、运行可执行文件

当可执行程序生成后，就可在计算机操作系统的支持下，直接运行。

当运行可执行目标程序文件后；若运行效果，与程序语言编写设计的初衷一致，该开发工作完成。但若运行效果未达到预期目的，需对源程序进行修改，重新经过“编辑-->编译-->链接”的开发过程，直至达到正确的运行效果。

任务 1-5 腾讯云 C 语言开发环境搭建

任务要求

通过本任务，掌握在腾讯云服务器中，搭建 C 语言开发环境的能力。

相关知识

(一) cmake

CMake 是一个跨平台的安装（编译）工具，可以用简单的语句来描述所有平台的安装（编译过程）。Cmake 内建 C 语言、C++、Fortran、Java 的自动相依性分析功能。

任务实现

【实训 1-5】：在 centos7 中，安装与调试 cmake

1、配置要求：

- 硬件要求：
腾讯云 CVM 服务器
- 操作系统要求：
CentOS 7
- 软件要求：
cmake-3.20.1.tar.gz
- cmake 下载地址：
<http://www.cmake.org/cmake/resources/software.html>

2、安装前提

系统中已经安装了 g++和 ncurses-devel

执行代码：

```
yum install gcc-c++  
yum install ncurses-devel
```

3、安装 cmake

将 cmake-3.20.1.tar.gz 文件上传到/root/opt，再 mv 到/usr/local 中执行下列命令：

- 步骤一：
上传文件位置：