



地球与空间科学系  
DEPARTMENT OF EARTH AND SPACE SCIENCES

# C程序设计基础

## Introduction to C programming Lecture 1: Introduction

张振国 [zhangzg@sustech.edu.cn](mailto:zhangzg@sustech.edu.cn)/19820223664

南方科技大学/理学院/地球与空间科学系

# 张振国

南方科技大学地球与空间科学系副教授

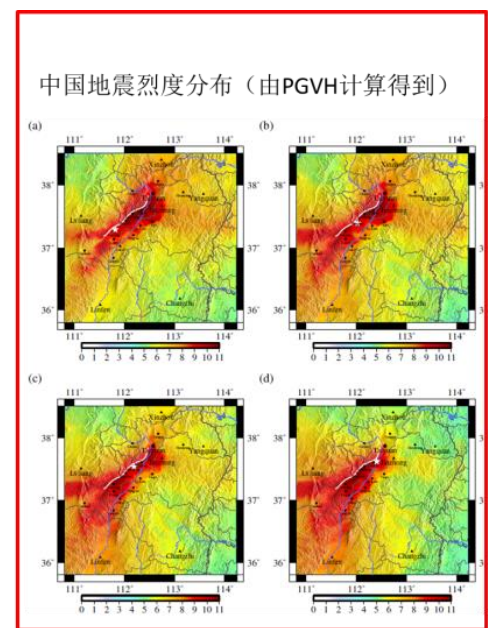
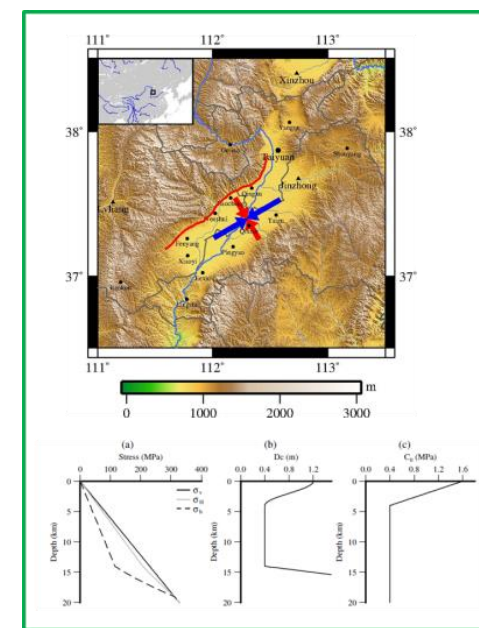
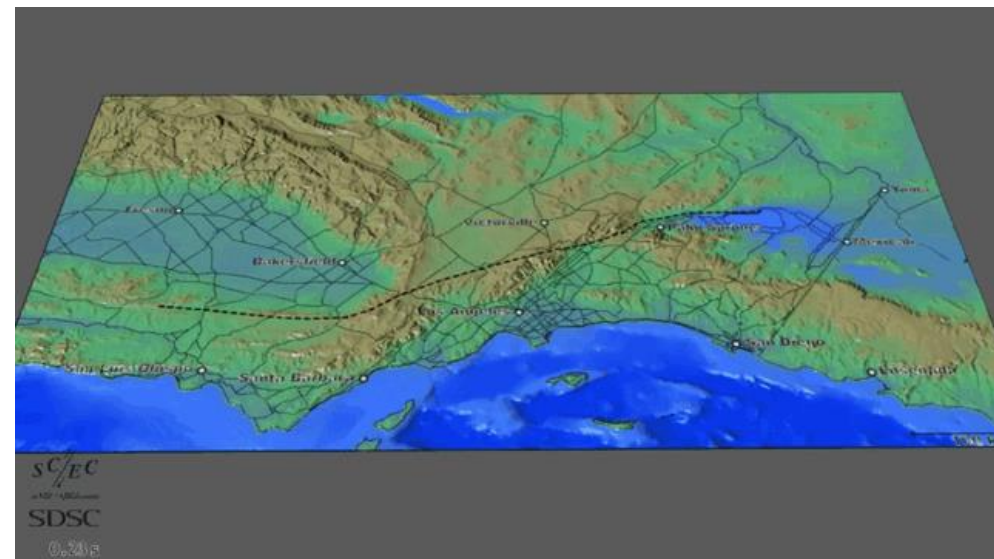
[zhangzg@sustech.edu.cn](mailto:zhangzg@sustech.edu.cn)

## 研究方向:

主要研究复杂介质中地震波传播、地震动力学、地震灾害、高性能计算等领域。

学术主页: <http://ess.sustech.edu.cn/Case-detail-id-7.html>

办公室: 理学院E3131



# Course objectives

---

- Learn the concepts of programming
- Learn how to program in C with a good habit
- Learn how to solve problems in science and engineering

# Course syllabus

---

- |         |        |               |        |
|---------|--------|---------------|--------|
| 1. 概述   | 2hours | 8. 函数         | 2hours |
| 2. 算法   | 2hours | 9. 宏          | 2hours |
| 3. 数据   | 4hours | 10. 指针        | 2hours |
| 4. 顺序程序 | 2hours | 11. 结构体       | 2hours |
| 5. 条件   | 2hours | 12. 文件I/O     | 4hours |
| 6. 循环控制 | 2hours | 13. 常见问题分析+复习 | 2hours |
| 7. 数组   | 4hours |               |        |

# Grading

---

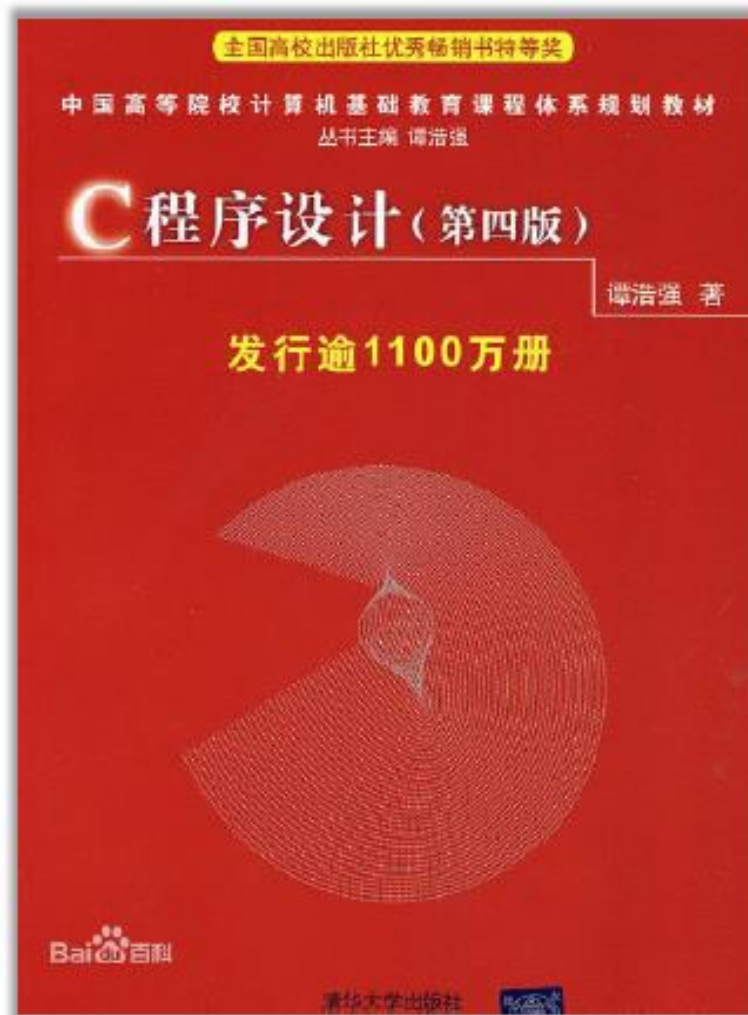
- Course/lab attendance: **10%**
- Assignments: **50%**
- Final exam: **40%**

# Polices

---

- Lecture will be released after class
- Assignment needs to be submitted on Friday. Finished individually
- Assignment will be reviewed in the lab session in the later week (code + simple report)

# Reference Books



- K. N. King, C语言程序设计现代方法（第2版），人民邮电出版社
- 谭浩强, C程序设计（第五版），清华大学出版社

---

## 无个人电脑同学：

- 一丹图书馆一楼
- 商学院210（自习室，20台）
- 软件：DEV-C++(体积小)

上课时间、地点：

理论课：1-16周，星期二第7-8节 一教107

实验课：1-16周，星期三第5-6节 三教504机房

理论课：1-16周，星期二第7-8节 一教107

实验课：1-16周，星期二第9-10节 三教507、508机房





众口难调  
相互理解  
尽量合作  
共同进步



姓名	学号	邮箱	年级	院系
徐天鸿	12231218	<a href="mailto:12231218@mail.sustech.edu.cn">12231218@mail.sustech.edu.cn</a>	博士生	地球与空间科学系
任碧荷	12132703	<a href="mailto:12132703@mail.sustech.edu.cn">12132703@mail.sustech.edu.cn</a>	硕士生	地球与空间科学系
张一舟	12112714	<a href="mailto:12112714@mail.sustech.edu.cn">12112714@mail.sustech.edu.cn</a>	21级本科	计算机科学与工程系
刘锦润	12011216	<a href="mailto:12011216@mail.sustech.edu.cn">12011216@mail.sustech.edu.cn</a>	20级本科	计算机科学与工程系
陈康睿	12110524	<a href="mailto:12110524@mail.sustech.edu.cn">12110524@mail.sustech.edu.cn</a>	21级本科	计算机系智能科学与技术图灵班
李晗晓	11812511	<a href="mailto:11812511@mail.sustech.edu.cn">11812511@mail.sustech.edu.cn</a>	18级本科	
刘凯	12032502	<a href="mailto:12032502@mail.sustech.edu.cn">12032502@mail.sustech.edu.cn</a>	20级硕士	计算机科学与工程系
鲍志远	11910501	<a href="mailto:11910501@mail.sustech.edu.cn">11910501@mail.sustech.edu.cn</a>	19级本科	计算机科学与工程系

# Contents

---

## Introduction to C (...) programming

- What is programming
- What is C language
- A first example in C
- How to run C program in Linux/Windows/MacOS

# Contents

---

- **What is programming**
- What is C language
- A first example in C
- How to run C program in Linux/Windows/MacOS

---

programming/编程

program/程序



---

## ■ 程序

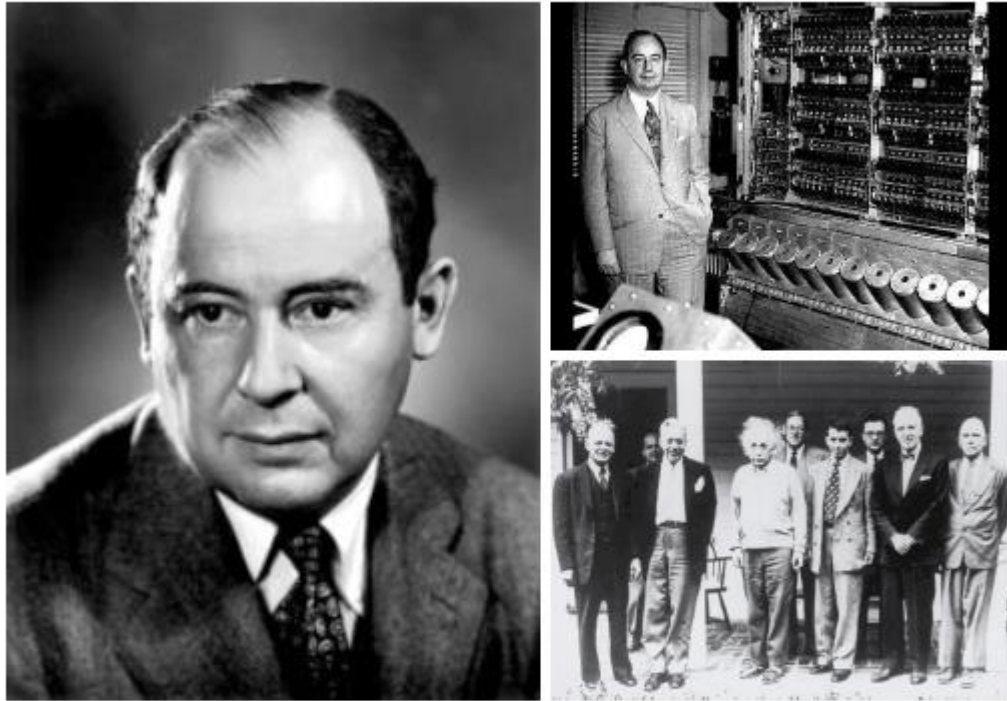
- 人们为解决某种问题用计算机可以识别的代码编排的一系列加工步骤。
- 程序的执行过程实际上是对程序所表达的数据进行处理的过程。

程序program与计算机密不可分



学好编程，做优秀的码农，  
可以在众多岗位上发挥作用

# Von Neumann architecture



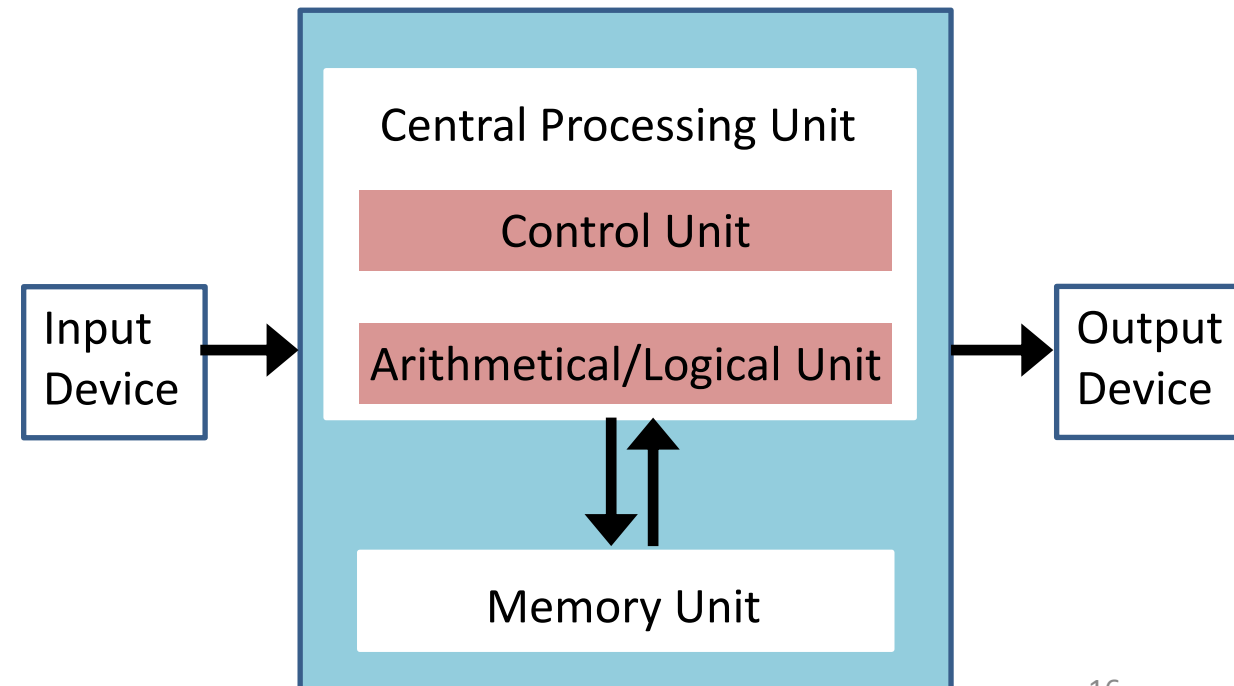
**John von Neumann (1903-1957)**

Hungarian-American mathematician physicist  
Founder of modern computer architecture

**Machine is programmable**

Von Neumann architecture (1946)

冯·诺依曼架构





# Von Neumann architecture

## Input unit

- Obtains data/information from physical world
- Interacts with human



**Machine is programmable**

Von Neumann architecture (1946)

冯·诺依曼架构

Input  
Device



Central Processing Unit

Control Unit

Arithmetical/Logical Unit



Memory Unit



Output  
Device

# Von Neumann architecture

## Output unit

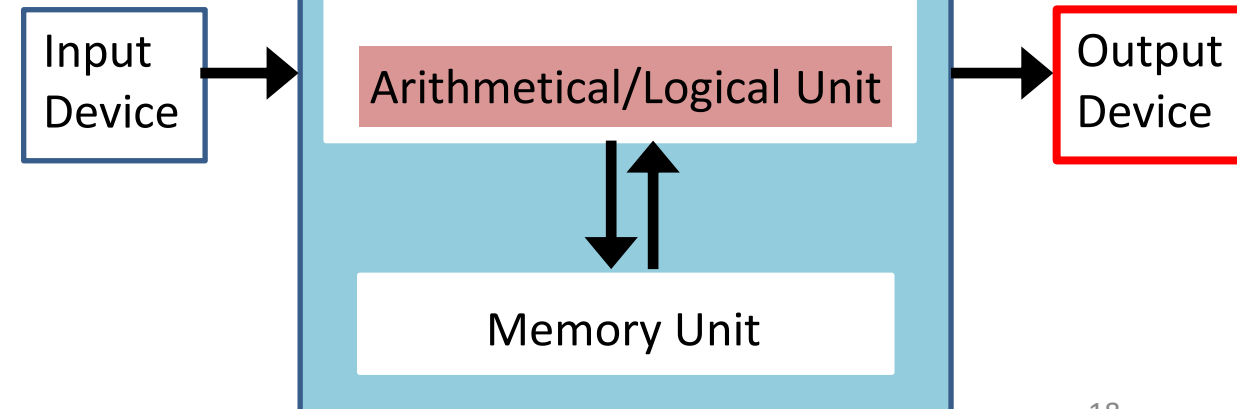
- Output data/information to physical world
- Interacts with human



**Machine is programmable**

Von Neumann architecture (1946)

冯·诺依曼架构



# Von Neumann architecture

## Memory unit

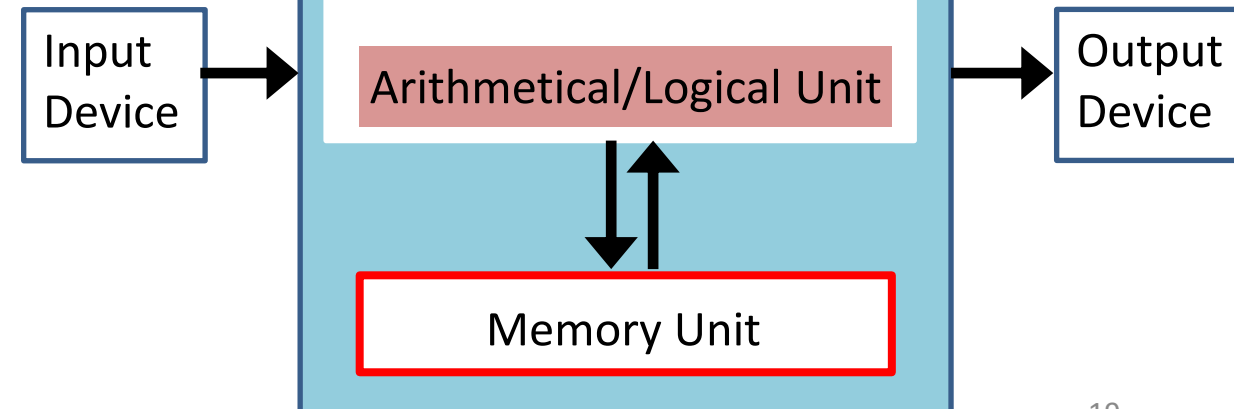
- Stores data/information in the machine, rapid-access
- Known as memory, primary memory, RAM



**Machine is programmable**

Von Neumann architecture (1946)

冯·诺依曼架构



# Von Neumann architecture

## Arithmetic logic unit (ALU)

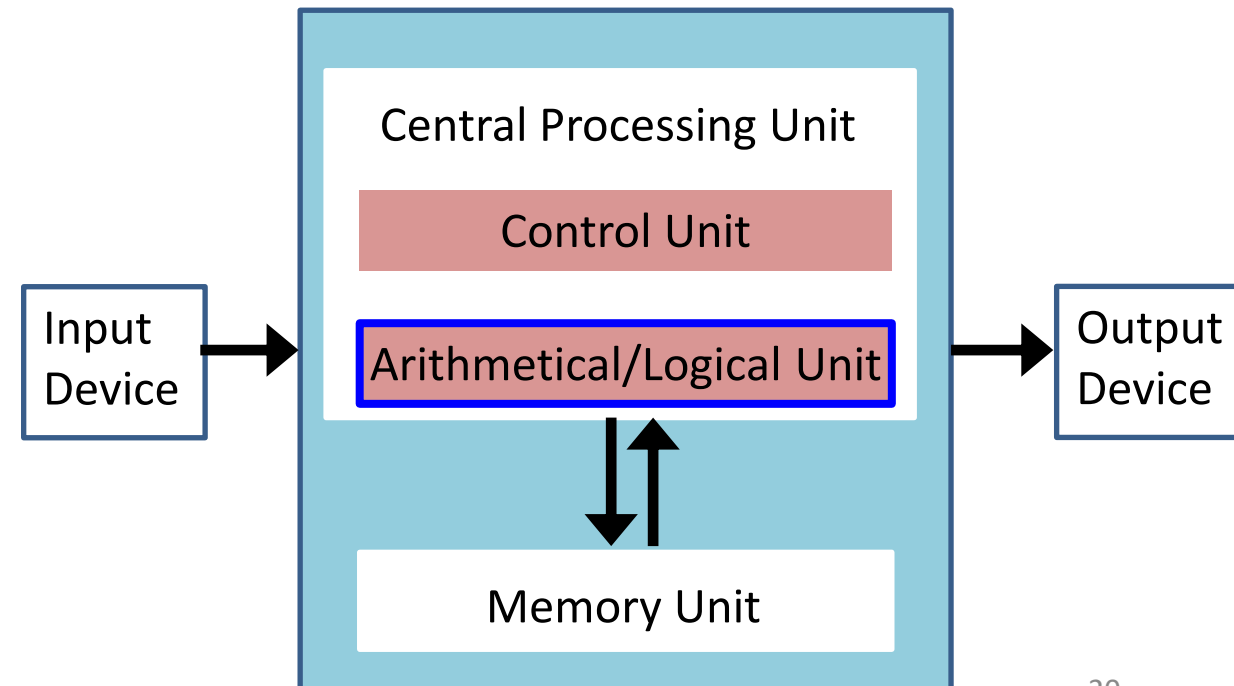
- Defines standard calculations and mechanisms for processing
- ALU is implemented on CPU



**Machine is programmable**

Von Neumann architecture (1946)

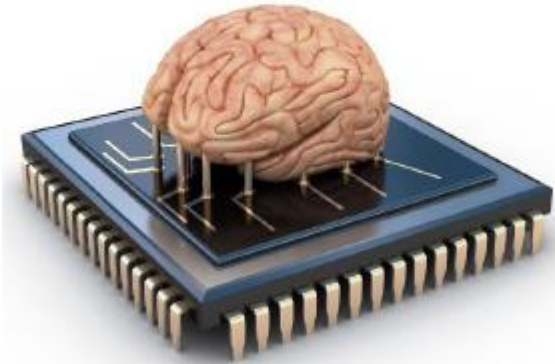
冯·诺依曼架构



# Von Neumann architecture

## Control unit

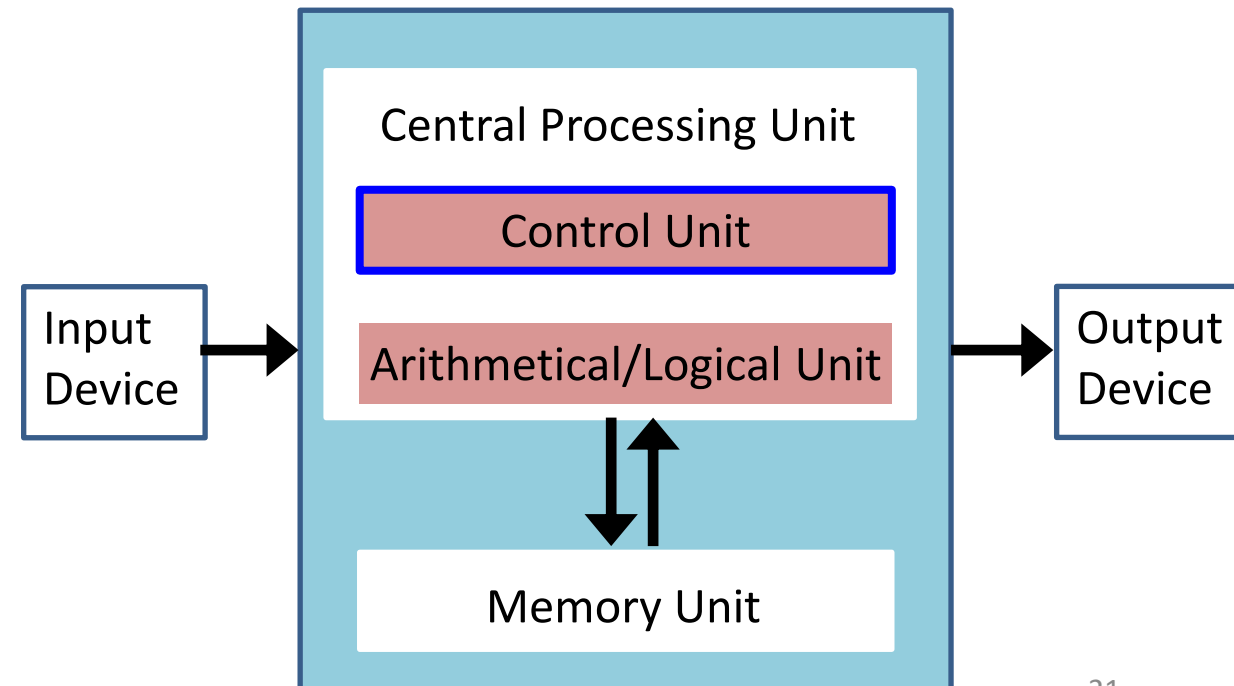
- Defines logics and workflows to process data/information
- **You can control this part!**



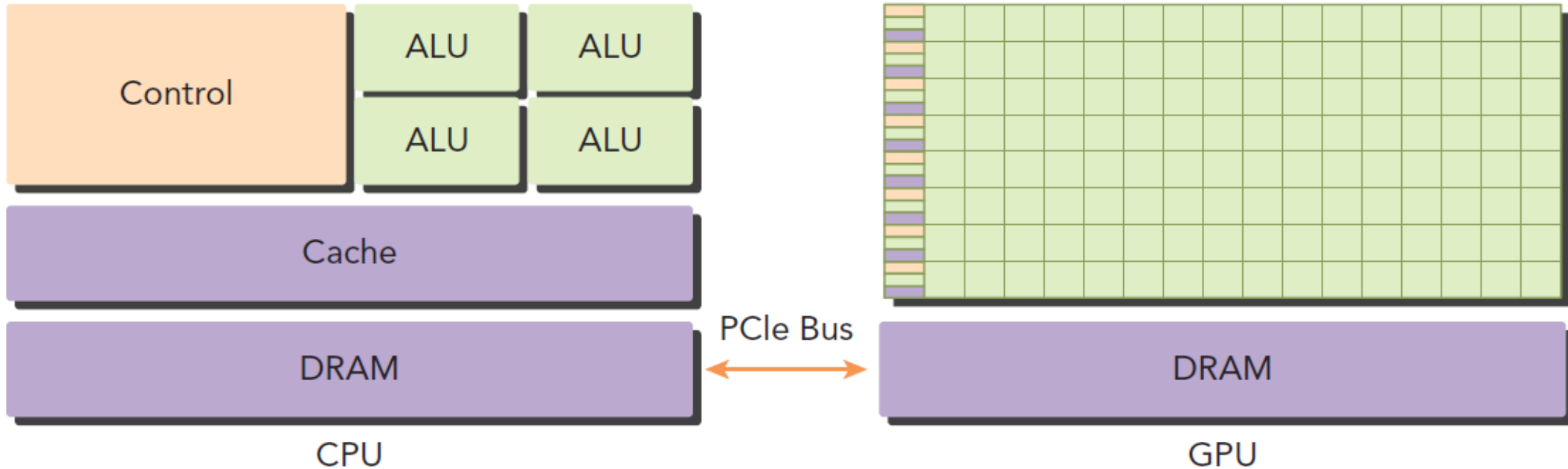
**Machine is programmable**

Von Neumann architecture (1946)

冯·诺依曼架构



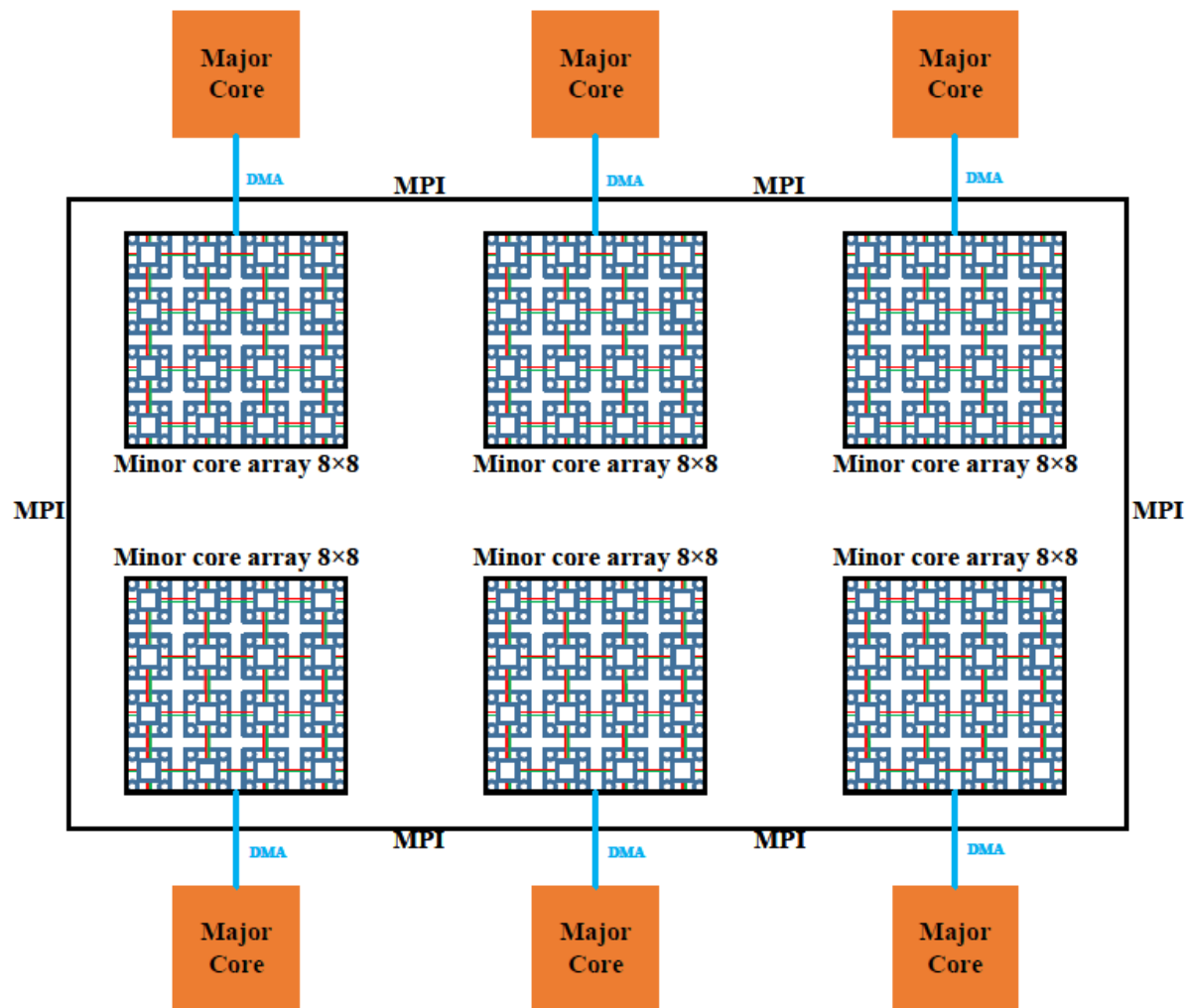
# More



- Goal: Minimum instruction delay
- Huge cache
- Complex control logic

- Goal: Maximum throughput
- More registers, high bandwidth
- Hide delay by thread rotation
- Multiple threads sharing control logic

# More

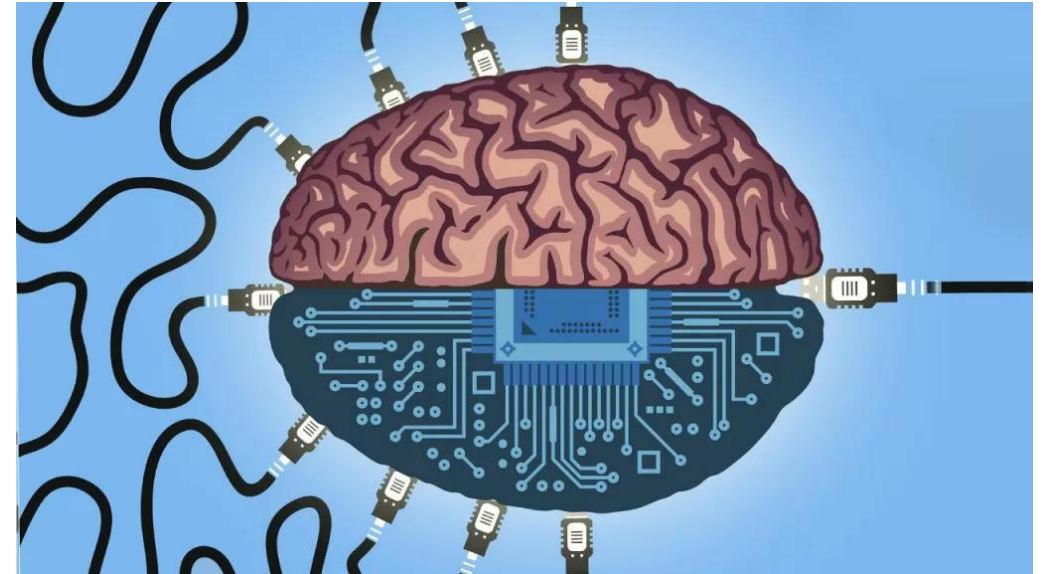
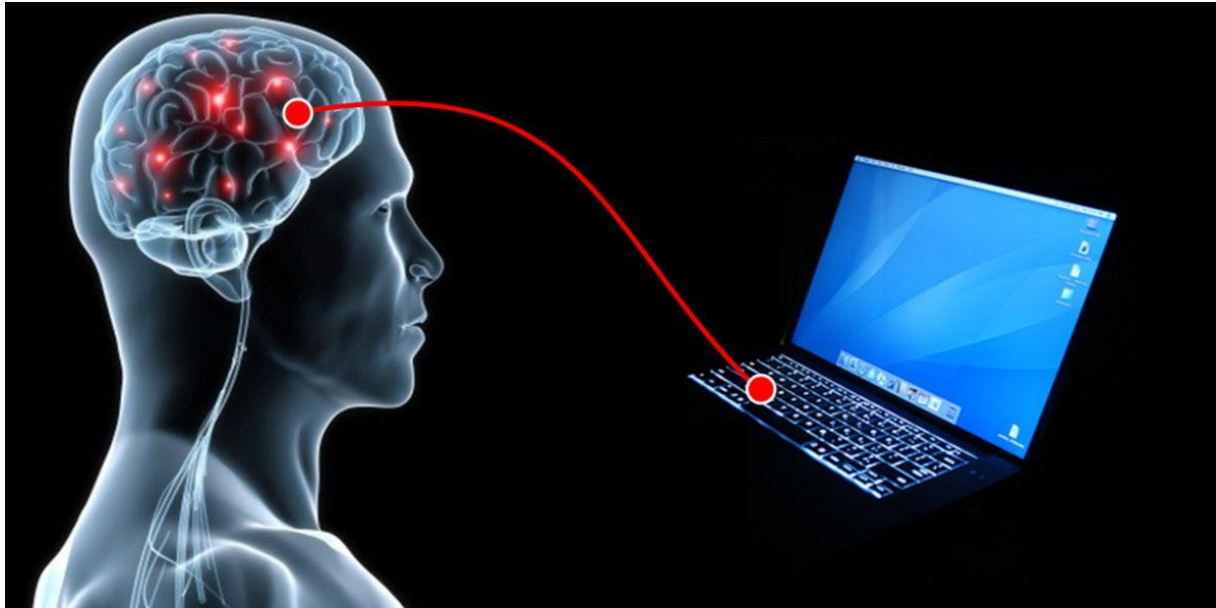


华为鲲鹏、NPU。。。

申威众核芯片架构

# How to program

---



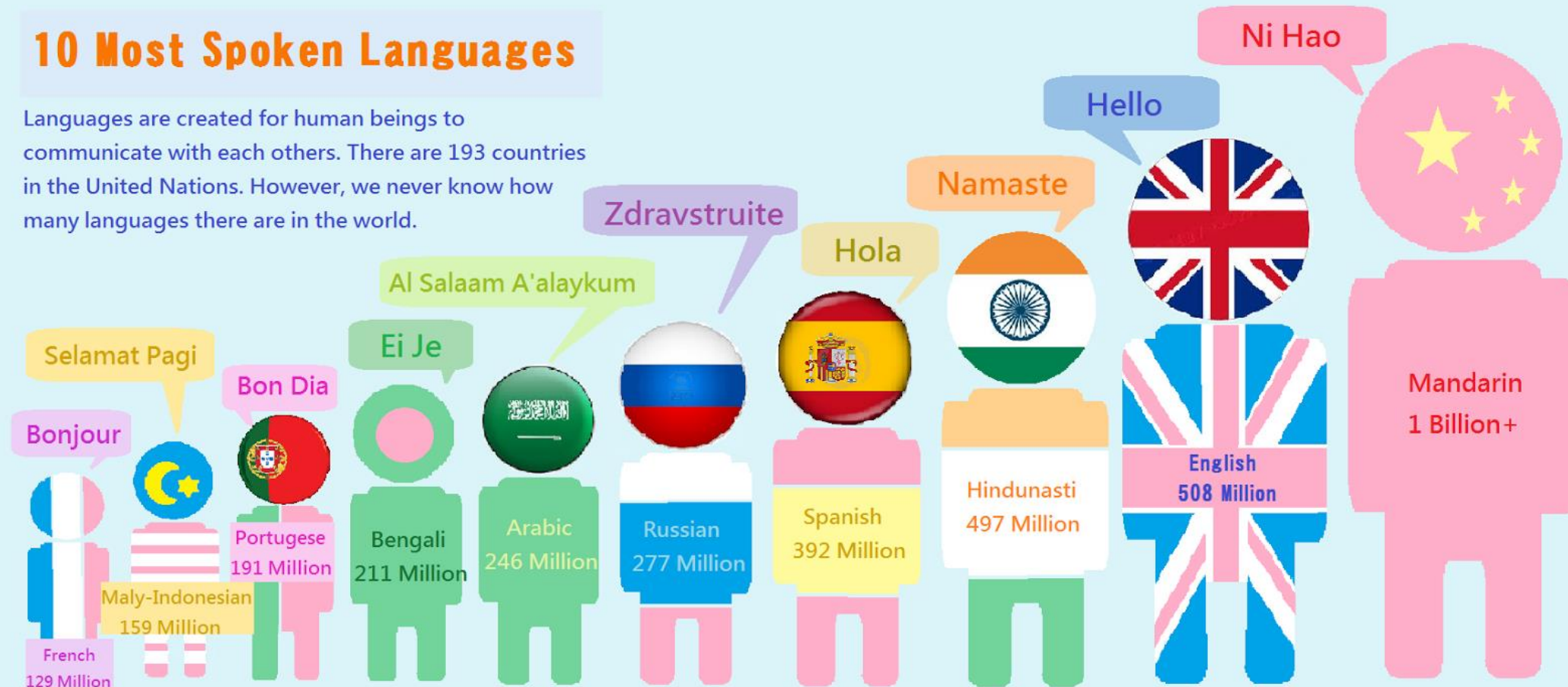


# How to program

We need a language

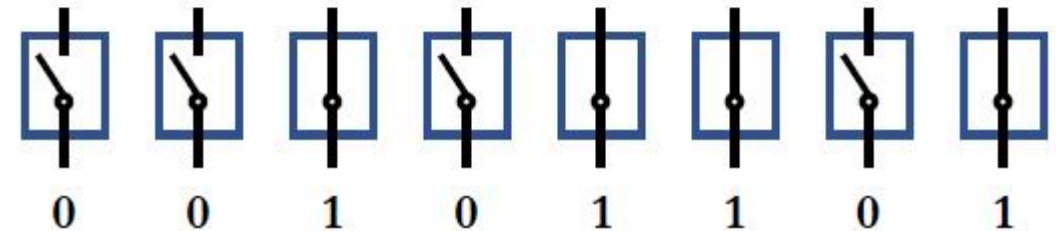
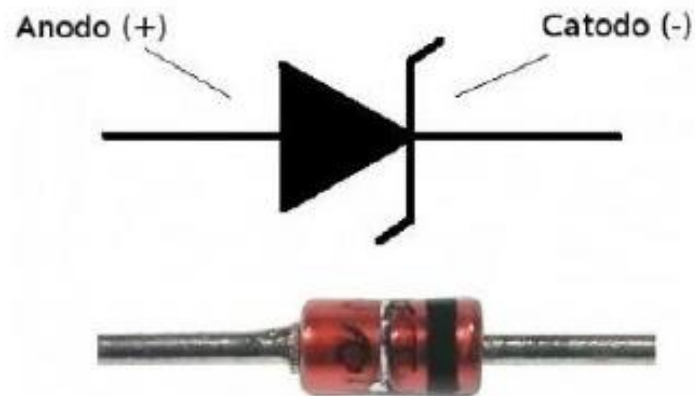
## 10 Most Spoken Languages

Languages are created for human beings to communicate with each others. There are 193 countries in the United Nations. However, we never know how many languages there are in the world.



Sources: 1. <http://geography.about.com/cs/countries/a/numbercountries.htm> 2. <http://listverse.com/2008/06/26/top-10-most-spoken-languages-in-the-world/>

- Human languages are complex;
- Machine speaks binary language: 0 and 1;
- A computer is nothing but a vast collection of electronic switches to store information.






## How computer interprets an image?


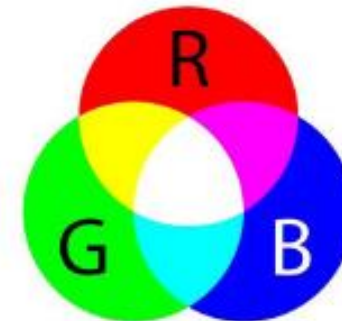


**Our colourful world is  
quantized in 256 values!**

**Machine uses 8 bit to quantize colours**  
 **$00000000 = 2^8 = 256$  scales**



**(130, 10, 50)**  
**11011011**  
**00000101**  
**00111010**



**(60, 60, 220)**  
**00010011**  
**00010011**  
**01111010**

We need a language not  
a tool.





用Python写个解挠曲方程 ( $d^4w/dx^4=f(x)$ ,  $x \in (0, 1)$ , 边界条件为  $w=d^2w/dx^2=0$ ) 的有限差分程序

以下是使用有限差分的 Python 代码来解决挠曲方程:

```
less Copy code

import numpy as np

def beam_bending(f, N):
    h = 1.0/(N+1) # grid size
    x = h*np.arange(1, N+1) # grid points

    # Stiffness matrix
    A = (2 + h**2)*np.eye(N) - np.eye(N, k=1) - np.eye(N, k=-1)

    # Right-hand side vector
    b = h**2*np.array([f(xi) for xi in x])

    # Impose boundary conditions
    b[0] -= 0
    b[-1] -= 0
    A[0,1] = 0
    A[-1,-2] = 0
    A[0,0] = 1
    A[-1,-1] = 1

    # Solve linear system
    w = np.linalg.solve(A, b)
    return w

使用方法:

python Copy code

def f(x):
    return np.sin(np.pi*x)

N = 10 # number of grid points
w = beam_bending(f, N) # solve the equation
```

Coded by ChatGPT

# How to program

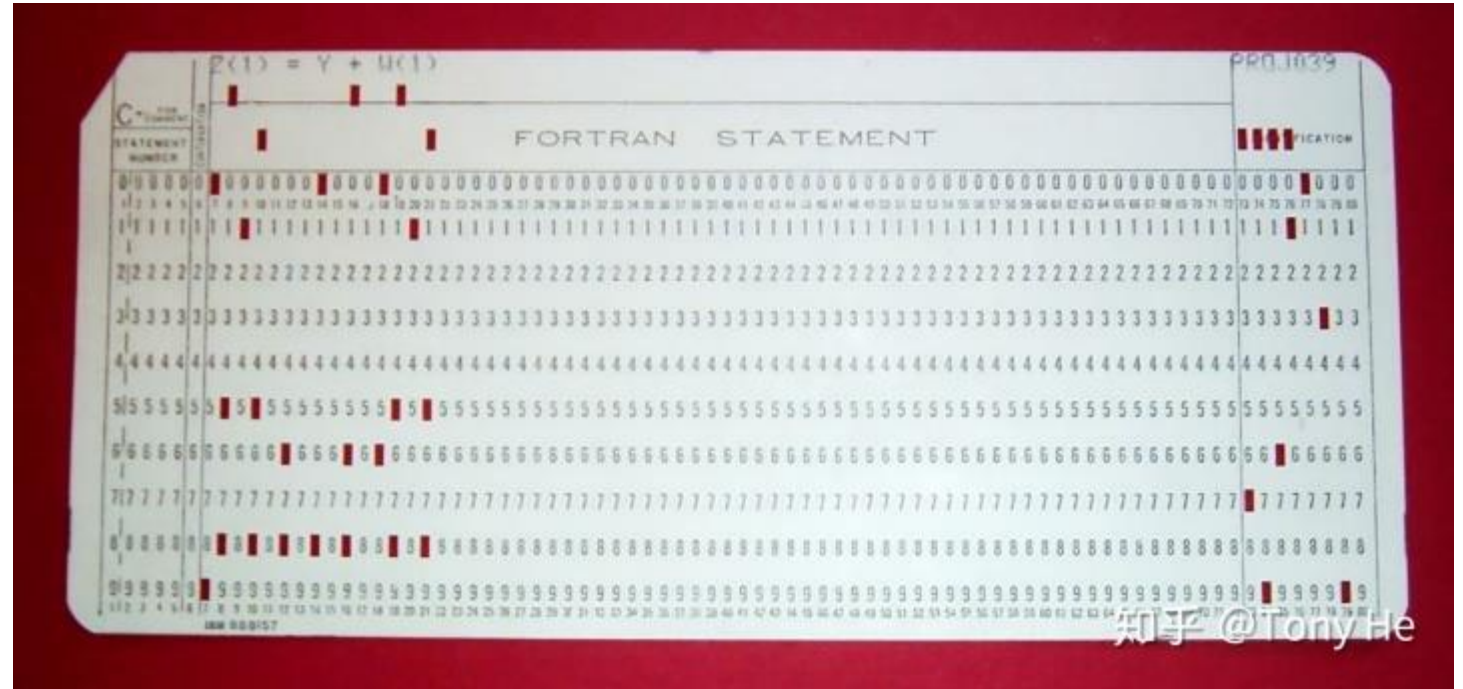
We need a language not a tool.

$$\frac{\partial^4 w}{\partial x^4} = f(x), \quad x \in (0, 1)$$

**警告:**  
不要尝试依赖类似  
工具完成作业!

# Programming languages

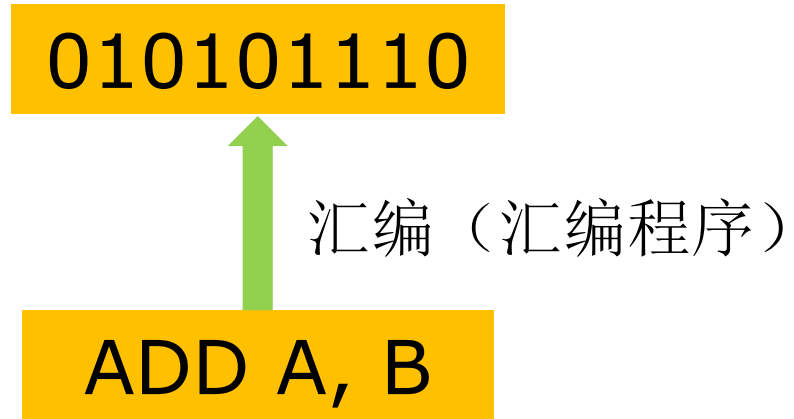
- Machine language



# Programming languages

---

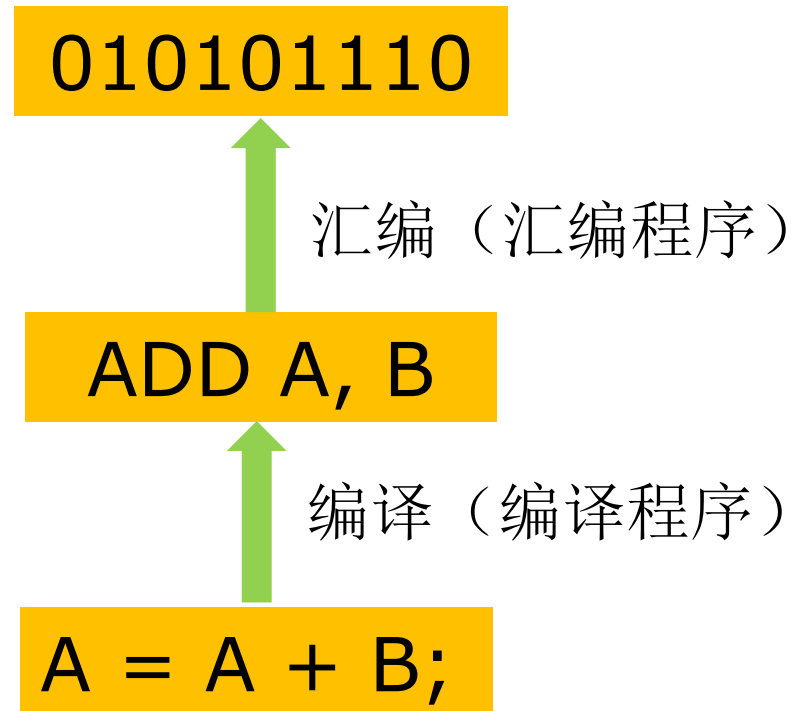
- Machine language
- Symbolic language



# Programming languages

---

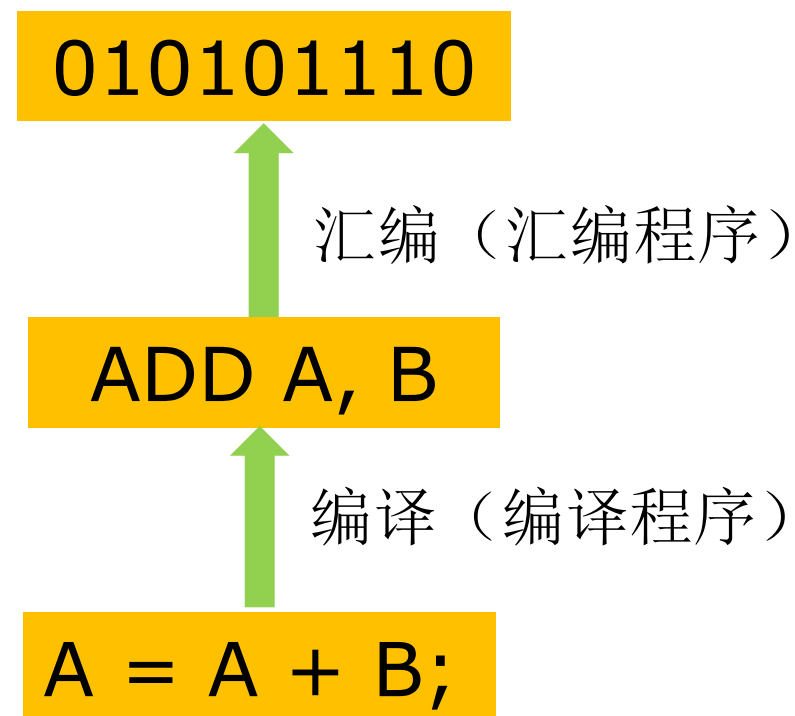
- Machine language
- Symbolic language
- High-level language





# Programming languages

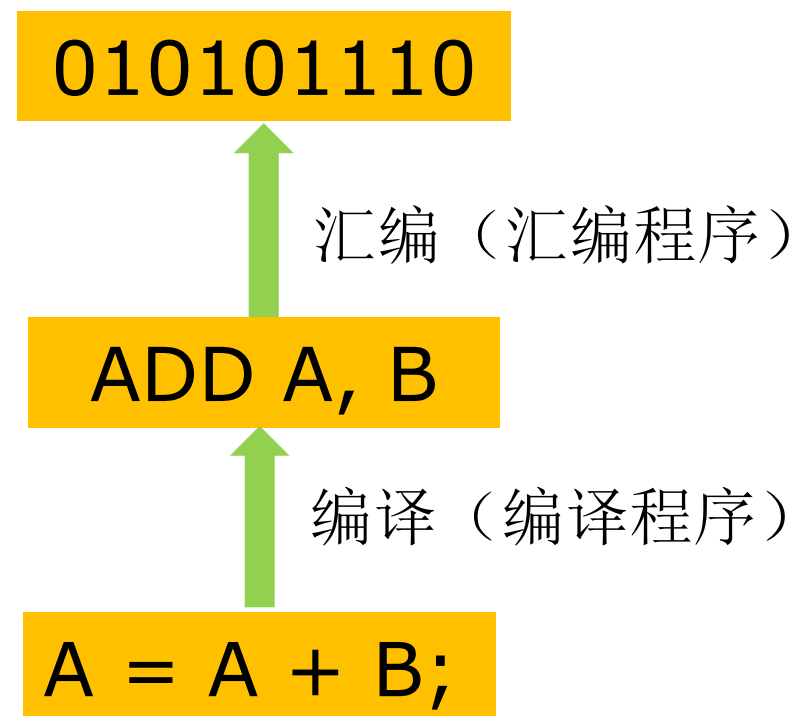
- High-level language
  - Unstructured language
    - 典例: 早期BASIC、COBOL、FORTRAN等语言
    - 特点: 不能反映结构程序设计的思想
    - 缺点: 程序的逻辑比较复杂



# Programming languages

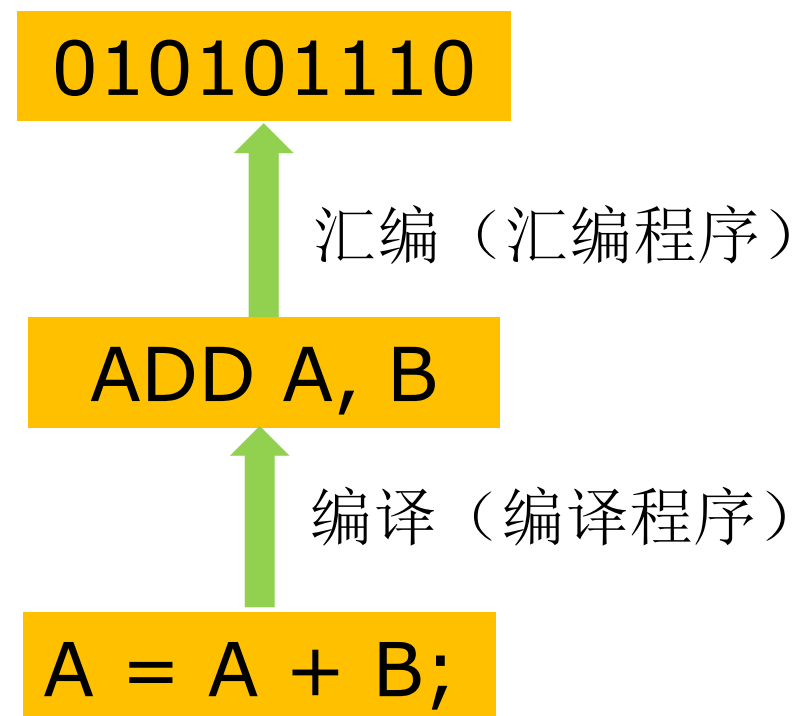
---

- High-level language
  - Unstructured language
  - Structured language
    - 规定程序必须由具有良好特性的基本结构（顺序、选择、循环结构）
    - 特点: 清晰易读和逻辑严密
    - QBASIC, FORTRAN, C



# Programming languages

- High-level language
    - Unstructured language
    - Structured language
    - Object-Oriented Language
      - C++, C#, Visual Basic, Java
      - 处理规模较大的问题
- 基于过程的语言
- 面向对象的语言



# Contents

---

- What is programming
- **What is C language**
- A first example in C
- How to run C program in Linux/Windows/MacOS

---

C是众多高级编程语言中的一种



## ■ 程序设计语言

- 提供了一种表达数据与处理数据的功能
- 要求程序员按照语言的规范编程

- 
- C语言是国际上广泛流行的高级语言。
  - C语言是在B语言的基础上发展起来的。
  - B（BCPL）语言是1970年由美国贝尔实验室设计的,并用于编写了第一个UNIX操作系统,在PDP 7上实现。优点：精练,接近硬件，缺点：过于简单,数据无类型。
  - 1973年贝尔实验室的D.M.Ritchie 在B语言的基础上设计出了C语言，对B取长补短，并用之改写了原来用汇编编写的UNIX，(即UNIX第5版），但仅在贝尔实验室使用。

- 
- 1975年UNIX第6版发布,C优点突出引起关注。
  - 1977年出现了《可移植C语言编译程序》，推动了UNIX在各种机器上实现，C语言也得到推广，其发展相辅相成。
  - 1978年影响深远的名著《The C Programming Language》由Brian W.Kernighan和Dennis M.Ritchie 合著,被称为标准C。
  - 之后，C语言先后移植到大、中、小、微型计算机上,已独立于UNIX和PDP，风靡世界,成为最广泛的几种计算机语言之一。

- 
- 1983年,美国国家标准化协会(ANSI)根据C语言各种版本对C的发展和扩充,制定了新的标准ANSI C，比标准C有了很大的发展。
  - 1988年K & R按照 ANSI C修改了他们的《The C Programming Language》。
  - 1987年，ANSI公布了新标准——87 ANSI C。
  - 1990年，国际标准化组织接受了87 ANSI C为ISO C 的标准(ISO9899—1990)。
  - 1994年，ISO又修订了C语言标准。
  - 目前流行的C语言编译系统大多是以ANSI C为基础进行开发的。



# C语言特征

---

- （1）语言简洁、紧凑，使用方便、灵活。 32个关键字、9种控制语句，程序形式自由。
- （2）运算符丰富。 34种运算符。
- （3）数据类型丰富，具有现代语言的各种数据结构。
- （4）具有结构化的控制语句，是完全模块化和结构化的语言。
- （5）语法限制不太严格，程序设计自由度大。

---

(6) 允许直接访问物理地址,能进行位操作,能实现汇编语言的大部分功能,可直接对硬件进行操作。兼有高级和低级语言的特点。

(7) 目标代码质量高,程序执行效率高。只比汇编程序生成的目标代码效率低10%-20%。

(8) 程序可移植性好(与汇编语言比)。基本上不做修改就能用于各种型号的计算机和各种操作系统。

**问题：**目前有各种编程语言，如Java, Python, Matlab, Fortran, ..., 为什么要学习C语言？

**解释1：**高级语言，接近自然表达，易于编写。

**解释2：**非数值计算需求

**问题：**既然有了面向对象的C++语言，为什么还要学习C语言？

**解释1：**C++是由于开发大型应用软件的需要而产生的，并不是所有的人都要去编写大型软件。

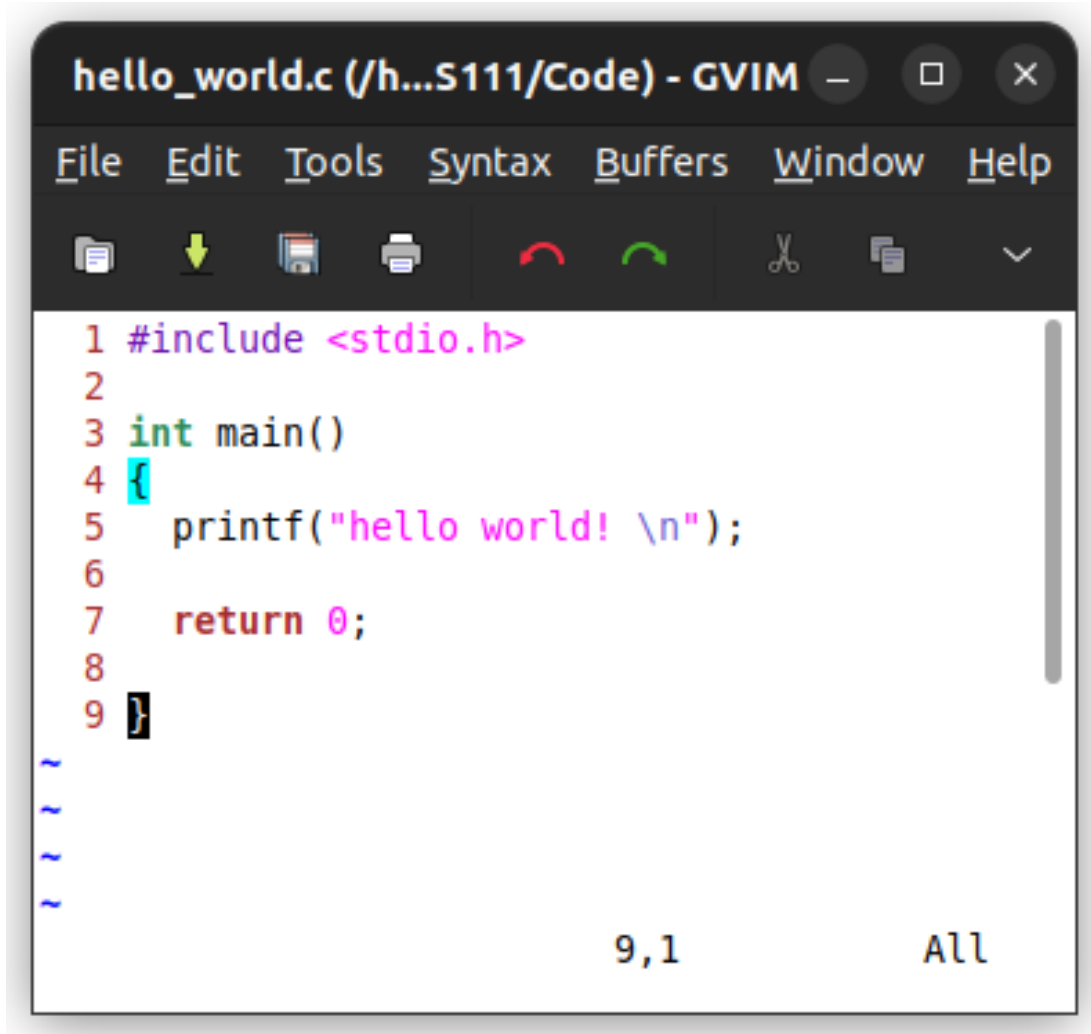
**解释2：**面向对象的基础是面向过程。C++是面向对象的语言，C是面向过程的，学起来比C语言困难得多，所以不太适合程序设计的初学者。

# Contents

---

- What is programming
- What is C language
- **A first example in C**
- How to run C program in Linux/Windows/MacOS

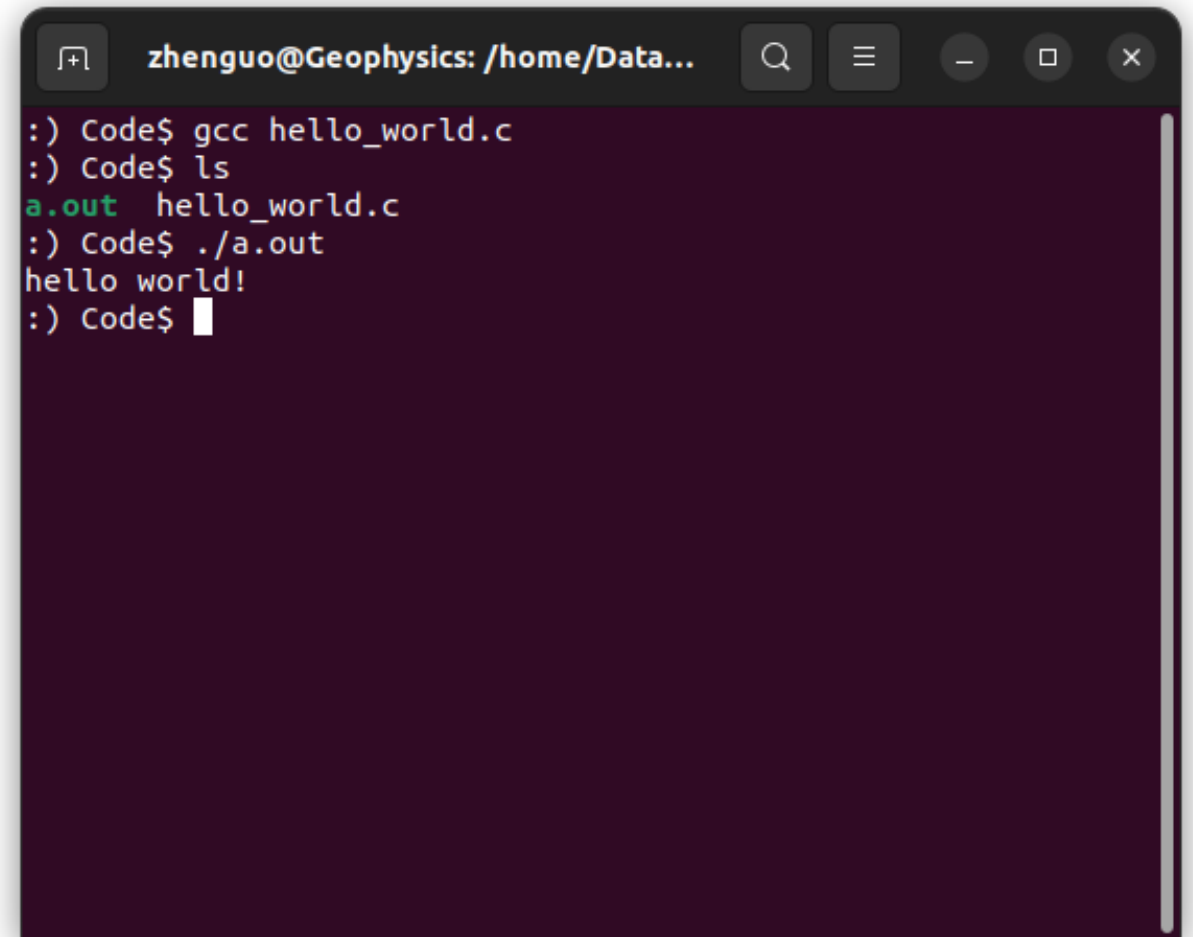
# A first example



The screenshot shows a GVIM editor window titled "hello\_world.c (/h...S111/Code) - GVIM". The menu bar includes File, Edit, Tools, Syntax, Buffers, Window, and Help. Below the menu is a toolbar with icons for file operations and editing. The main text area contains the following C code:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("hello world! \n");
6
7     return 0;
8
9 }
```

At the bottom right of the editor, the status bar displays "9,1" and "All".



The screenshot shows a terminal window with the prompt "zhenguo@Geophysics: /home/Data...". The following commands and output are shown:

```
: ) Code$ gcc hello_world.c
: ) Code$ ls
a.out hello_world.c
: ) Code$ ./a.out
hello world!
: ) Code$
```

## The simplest example

```
#include <stdio.h>

int main()
{
    printf("Hello World");

    return 0;
}
```

Main body of program:

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

Action:

```
printf("Hello World");
```

Pre-compiled head file:

```
#include <stdio.h>
```



# 简单的C语言程序介绍

---

```
#include <stdio.h>
void main()
{
    printf ("Hello world!\n");
}
```

/\*文件包含\*/  
/\*主函数 \*/  
/\*函数体开始\*/  
/\*输出语句\*/  
/\*函数体结束\*/

**说明：** main-主函数名， void-函数类型

- 每个C程序必须有一个主函数main
- { }是函数开始和结束的标志, 不可省
- 每个C语句以分号结束
- 使用标准库函数时应在程序开头一行写：

```
#include <stdio.h>
```



int main()  
int/float/char/\*\* function()



return val;

void main()  
void function()



int main()

有利于检查程序是否正常执行，推荐

---

```
int main()  
int/float/char/*** function()
```

```
void main()  
void function()
```



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



标准写法，可移植性高

**说明：** 输出一行信息: sum is 579

## 例 求两数之和

```
#include <stdio.h>
void main( )      /*求两数之和*/
{
    int a, b, sum;  /*声明，定义变量为整型*/
    /*以下3行为C语句 */
    a=123; b=456;
    sum=a+b;
    printf(" sum is %d\n" , sum);
}
```

**说明：** /\*……\*/表示注释。注释只是给人看的, 对编译和运行不起作用。所以可以用汉字或英文字符表示，可以出现在一行中的最右侧，也可以单独成为一行。//注释后续行内容

---

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("Line 1 ! \n");
```

```
    printf("Line 2 ! \n");
```

```
    /*
```

```
    printf("Line 3 ! \n");
```

```
    printf("Line 4 ! \n");
```

```
    printf("Line 5 ! \n");
```

```
    return 0;
```

```
}
```

**ERROR!**

---

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("Line 1 ! \n");
```

```
    printf("Line 2 ! \n");
```

```
    /*
```

```
    printf("Line 3 ! \n");
```

```
    printf("Line 4 ! \n");
```

```
    /*
```

```
    printf("Line 5 ! \n");
```

```
    */
```

```
    return 0;
```

```
}
```

X

Line 1 !

Line 2 !

Line 3 !

Line 4 !

✓

Line 1 !

Line 2 !

---

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("Line 1 ! \n");
```

```
    printf("Line 2 ! \n");
```

```
    /*
```

```
    printf("Line 3 ! \n");
```

```
    printf("Line 4 ! \n");
```

```
    /*
```

```
    printf("Line 5 ! \n");
```

```
    */
```

```
    return 0;
```

```
}
```

X

Line 1 !

Line 2 !

Line 3 !

Line 4 !

✓

Line 1 !

Line 2 !

注意代码颜色（编译器有一定的语法高亮能力）



## 例 求3个数中较大者。

```
#include <stdio.h>
void main( )    /* 主函数*/
{
    int max(int x, int y); / 对被调用函数max的声明 */
    int a, b, c;      /*定义变量a、b、c */
    scanf(" %d, %d" , &a, &b); /*输入变量a和b的值*/
    c=max(a, b); /*调用max函数, 将得到的值赋给c */
    printf(" max=%d\\n" , c); /*输出c的值*/
}
```

**说明：**本程序包括main和被调用函数max两个函数。max函数的作用是将x和y中较大者的值赋给变量z。return语句将z的值返回给主调函数main。

### • 程序运行情况如下：

- 8, 5 ↙ (输入8和5赋给a和b)
- max=8 (输出c的值)

```
int max(int x, int y)
{
    int z;
    if (x>y) z=x;
    else z=y;
    return (z);
}
```

# 简单的C语言程序介绍

---

C程序：

(1) C程序是由函数构成的。这使得程序容易实现模块化。

(2) 一个函数由两部分组成：

函数的首部：例1.3中的max函数首部

```
int max(int x,int y )
```

函数体：花括号内的部分。若一个函数有多个花括号{}，则最外层的一对花括号为函数体的范围。

函数体包括两部分：

声明部分：`int a,b,c;` 可缺省

执行部分：由若干个语句组成。可缺省

# 简单的C语言程序介绍

---

C程序:

所有变量必须先声明才能使用

```
int main(void)
```

```
{
```

```
    int a, b, c;
```

```
    a = 1;
```

```
    b = 2;
```

```
    c = a+b;
```

```
}
```

```
int a;
```

```
int b, c;
```

```
int main(void)
```

```
{
```

```
    int a, b;
```

```
    a = 1;
```

```
    b = 2;
```

```
    int c;
```

```
    c = a+b;
```

```
}
```

# 简单的C语言程序介绍

---

## 注意：

函数的声明部分和执行部分都可缺省，例如：

```
void dump ()  
{  
}
```

这是一个空函数, 什么也不做, 但是合法的函数。

??? : 什么时候用到???

# 汉字编程？

```
hello_world.c (/h...S111/Code) - GVIM
File Edit Tools Syntax Buffers Window Help

1 #include <stdio.h>
2
3 int main()
4 {
5     printf("hello world! \n");
6     _____
7     return 0;
8     _____
9 }

~
~
~
~

9,1 All
```

打印（“你好！”）；  
返回0；

## 仓颉语言-华为



人人都是程序员  
2022-03-26 18:54 · 优质科技领域创作者

+ 关注

就在刚刚，我已成功运行华为仓颉国产编程语言的“Hello World!” 🤖 成为了一名仓颉程序员 😊

6545

1387

收藏

分享

① 举报

评论 1387



说点什么吧...

评论



人人都是程序员 作者

530 个

大家好热情，我就几个大家比较关心的热门问题进行集中回答。1.不是「中文编程」🤖 看我输出的应该能猜出来用什么编程🤖 说是「中文编程」的都是营销号。2.我签了保密协议，不能发代码相关的东西，请大家理解。🤖 3.特点是简单、易上手🤖 另外，等到能公开的那一刻，我第一时间出教程，关注我别错过。 特别感谢大家对华为的支持，对国产编程语言的支持 🙏 华为加油 🙏 仓颉加油 🙏 国产编程语言加油 🙏

回复 · 11小时前

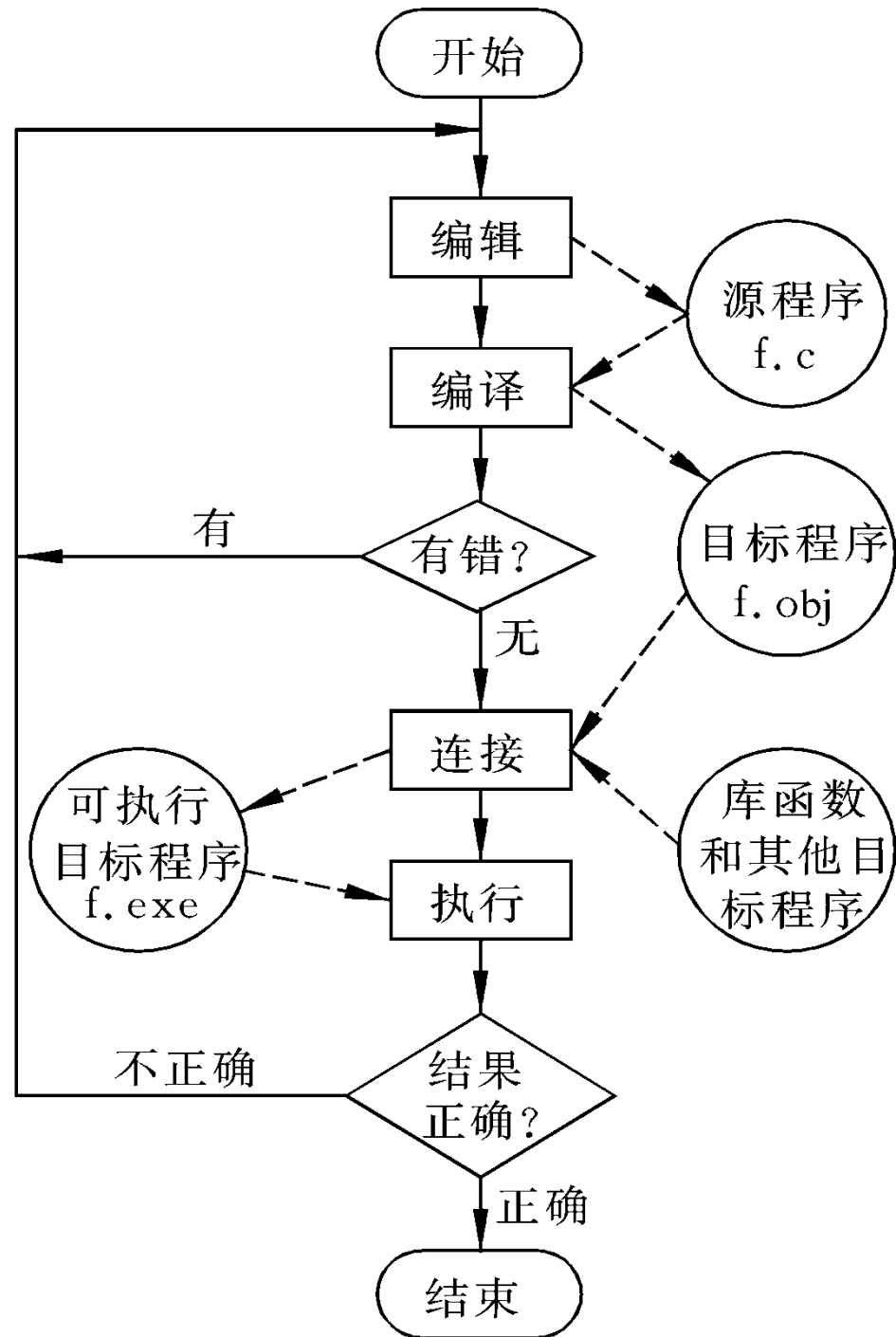
查看全部 282 条回复

# Contents

---

- What is programming
- What is C language
- A first example in C
- **How to run C program in Linux/Windows/MacOS**

# How to run C program



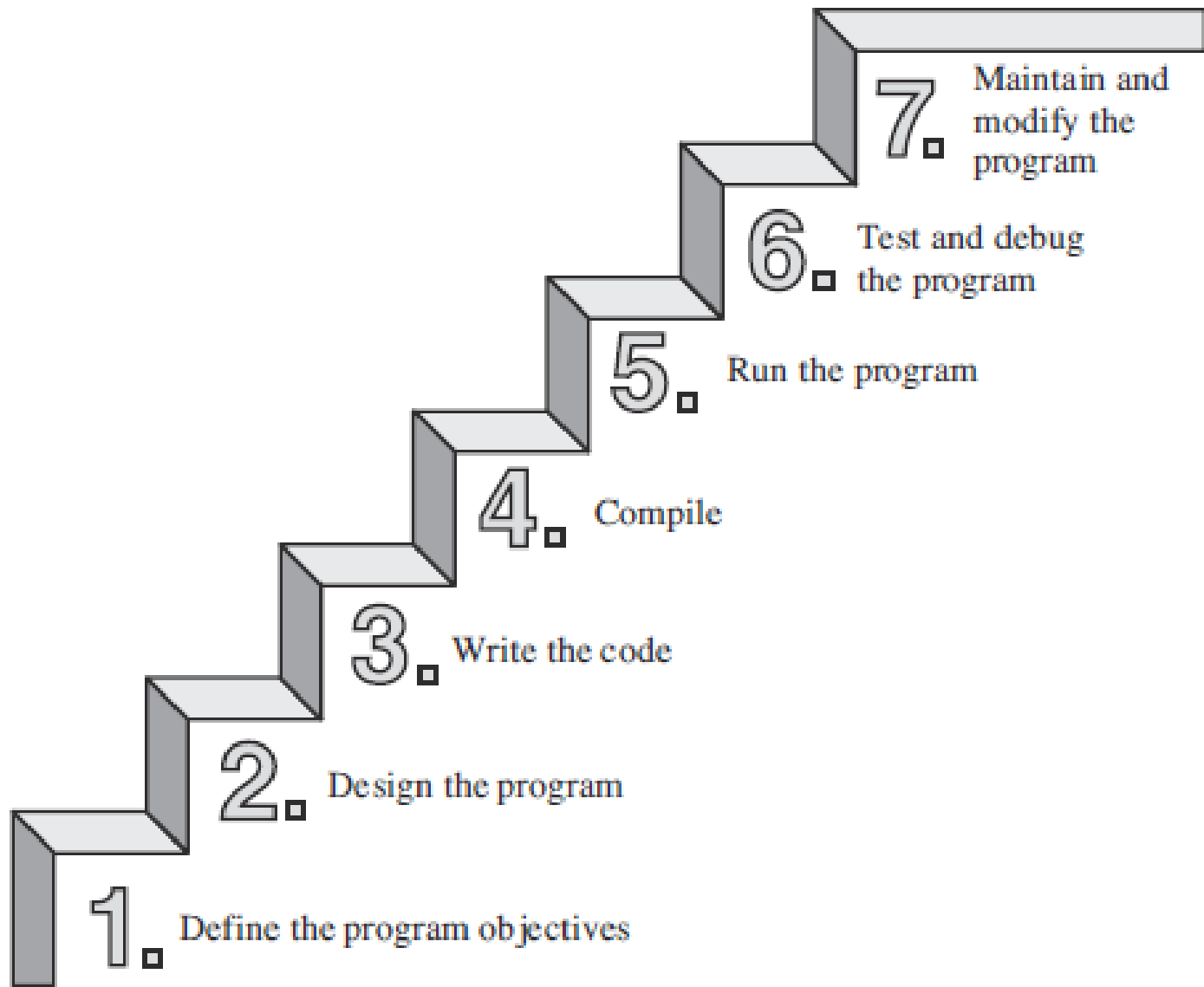


---

注意：

上述编程方式介绍只是完整编程的一部分，面对实际问题的编程更为复杂，耗时更多。

编程7步骤/7步走

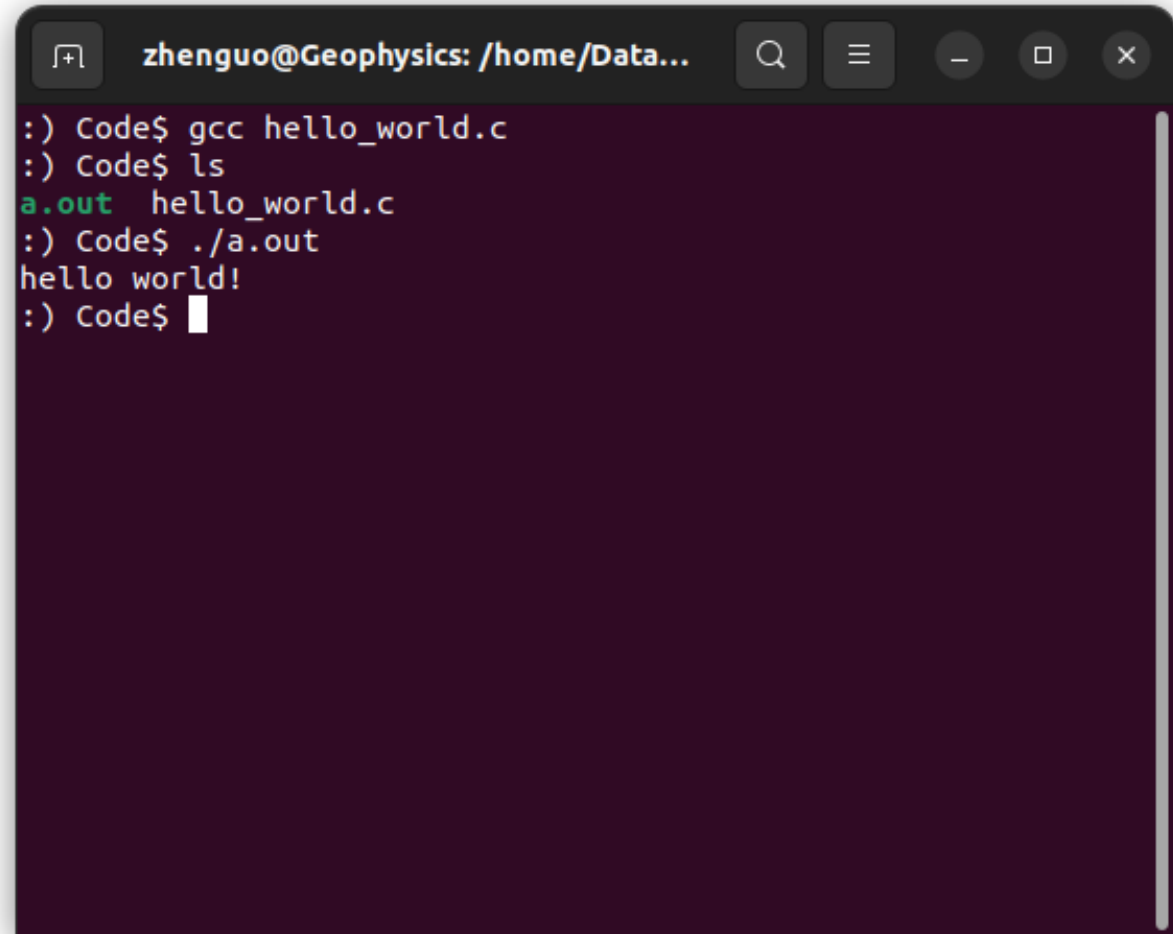


# on Linux

---

I. `gcc hello_world.c`

II. `./a.out`



```
zhenguo@Geophysics: /home/Data...  
:) Code$ gcc hello_world.c  
:) Code$ ls  
a.out hello_world.c  
:) Code$ ./a.out  
hello world!  
:) Code$
```

# on Linux

---

I. gcc hello\_world.c

II. ./a.out

I. gcc hello\_world.c -o hello

II. ./hello

I. gcc -E hello\_world.c -o hello.i #预处理阶段

II. gcc -S hello\_world.i -o hello.s #编译阶段

III. gcc -c hello\_world.s -o hello.o #汇编阶段

IV. gcc hello.o -o hello #链接阶段

V. ./hello

# on Linux

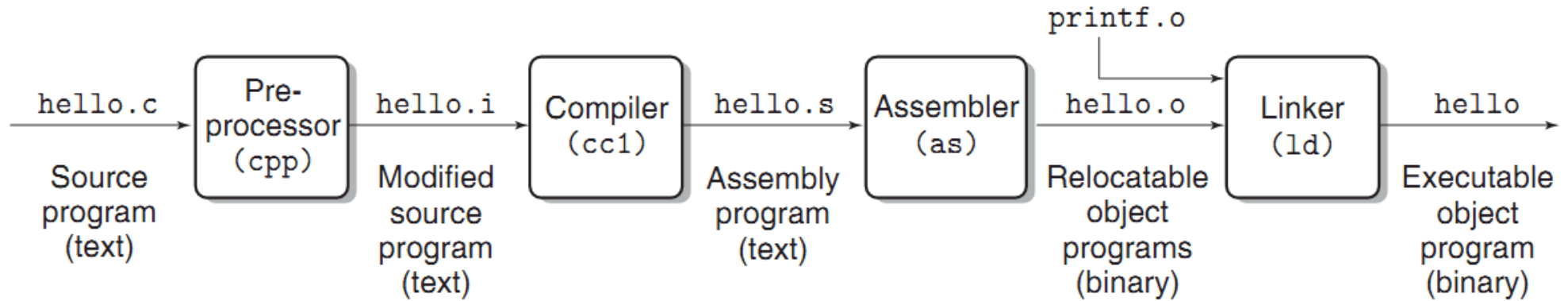


Figure 1.3 The compilation system.

```
unix> ./hello
hello, world
unix>
```

- ❑ 预处理Preprocessing phase
- ❑ 编译Compilation phase
- ❑ 汇编Assembly phase
- ❑ 链接Linking phase

# on Linux

---

- I. `gcc -E hello_world.c -o hello.i` #预处理阶段
- II. `gcc -S hello_world.i -o hello.s` #编译阶段
- III. `gcc -c hello_world.s -o hello.o` #汇编阶段
- IV. `gcc hello.o -o hello` #链接阶段
- V. `./hello`

大型程序的建议：

用Make/Cmake等脚本语言实现预处理、编译、调用、链接等一系列步骤

