



华中科技大学

# 第1章 引论

许向阳

**xuxy@hust.edu.cn**





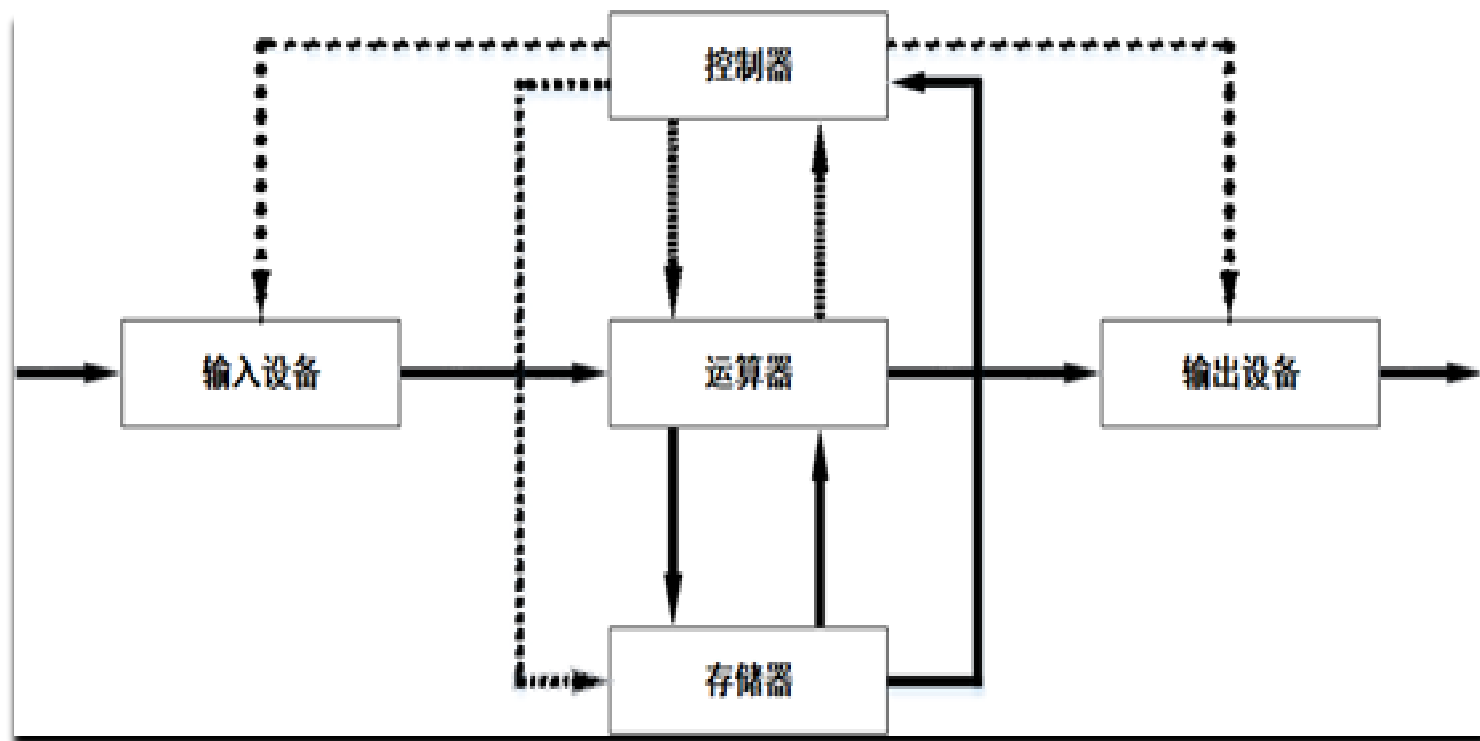
# 第一章 引论

- 1.1 计算机的体系结构
- 1.2 进制及转换和运算
- 1.3 80X86 系列汇编语言
- 1.4 C++的发展历史及特点
- 1.5 语法图与程序流程图
- 1.6 编译环境的安装与使用



# 1.1 计算机的体系结构

## 1.1.1 计算机系统的结构





# 1.1 计算机的体系结构

## 1.1.2 数据与程序的存储方式

- 内存按字节编址
- 一个字节由8位二进制构成，二进制位没有地址。
- 数据和指令均于二进制形式存于内存，**数据可以当作指令看待，指令也可以当作数据看待**
- 当指令计数器指向某个内存地址，自此地址开始的数据被当作指令执行。
- 每个字节能存一个字符，一个整数占用多个字节
- Windows分配内存的最小单位为节：1节=16字节。





# 1.1 计算机的体系结构

## 1.1.2 数据与程序的存储方式

➤ ASCII American Standard Code for Information Interchange

利用 字符编码的规律编写程序

统计 一个字符数组中各个小写字母出现的次数

```
char a[1024];
```

```
int count[26]={0}; // 所有元素的初值均为 0
```

```
// count[0]; 表示 'a' 出现的次数
```

```
// count[1]; 表示 'b' 出现的次数 .....
```

```
for (int i=0; i<1024; i++) count[ a[i] - 'a' ] ++;
```





# 1.1 计算机的体系结构

`int count[26]={0};` // 所有元素的初值均为 0

**Q：** 下面个表达式的含义是什么？

`int count[26]={1};`

`count[0] =1;` 其他元素的初值为0

`int count[26]={2, 3};`

`count[0] =2; count[1]=3;` 其他元素的初值为0

`int count[26];`

所有元素 未初始化





## 1.2 进制及其转换和运算

```
int x = 10;    // x的初值 为 (十进制的) 10
int y = 0x10;  // y的初值 为 (十进制的) 16
int p = 010;   // p 的初值 为 (十进制的) 8
int q = 0b10;  // q的初值 为 (十进制的) 2
```

不同进制的表达

0x 开头，十六进制

0 开头，八进制

0b 开头，二进制





# 1.2 进制及其转换和运算

## 二进制运算

$$\begin{array}{r} 10110010 \\ \text{按位与} \& 11010011 \\ \hline 10010010 \end{array}$$

$$\begin{array}{r} 10110010 \\ \text{按位或} | 11010011 \\ \hline 11110011 \end{array}$$

$$\begin{array}{r} 10110010 \leftarrow \\ \text{按位异或} ^ 11010011 \leftarrow \\ \hline 01100001 \leftarrow \end{array}$$

按位与      &

逻辑与      &&

按位或      |

逻辑或      ||

按位异或      ^







## 1.2 进制及其转换和运算

### 按位运算 VS 逻辑运算

```
int x = 0b101, y = 0b010;  
int z = x && y;           // z=1 两个都为真，结果为真  
  
00891860 cmp      dword ptr [x],0  
00891864 je       main+58h (0891878h)  
00891866 cmp      dword ptr [y],0  
0089186A je       main+58h (0891878h)  
0089186C mov      dword ptr [ebp-134h],1  
00891876 jmp      main+62h (0891882h)  
00891878 mov      dword ptr [ebp-134h],0  
00891882 mov      eax,dword ptr [ebp-134h]  
00891888 mov      dword ptr [z],eax
```





# 1.3 80X86系列汇编语言

## 32位的寄存器

EAX、EBX、ECX、EDX、ESI、EDI、EBP、ESP

## 6 种寻址方式

立即寻址

直接寻址

寄存器寻址

寄存器间接寻址

变址寻址

基址加变址寻址





## 1.4 C++的历史及特点

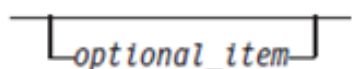
- (1) C的超集;
- (2) 强类型检查;
- (3) 混合型面向对象;
- (4) 支持多继承;
- (5) 运算符重载;
- (6) 支持移动语义;
- (7) 异常及断言处理;
- (8) 函数与类模板;
- (9) Lamda表达式;
- (10) 支持类型推导;
- (11) 支持名字空间。

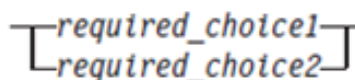


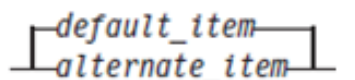


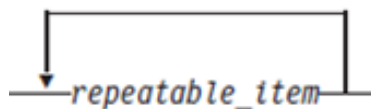
## 1.5 语法图及程序流程图

- ➡ 表示一个语法单位的开始
- 表示由一个语法单位流向另一个语法单位
- ➡ 表示续接上一次的流向
- ➡ 表示一个语法单位的结束

 表示一个可选项

 表示必选其中一项

 表示缺省项：如果替代项不出现，就表示使用缺省项

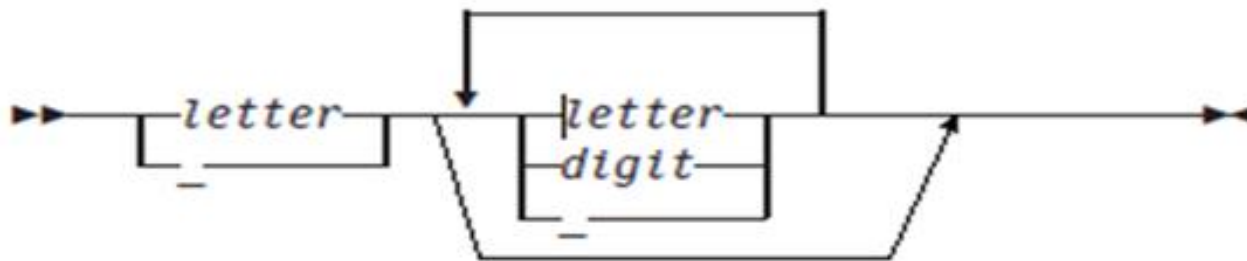
 表示重复出现项



## 1.5 语法图及程序流程图

**标识符**是由下划线或字母开始，后跟任意个由下划线、字母、数字组成的字符序列。

标识符的语法图

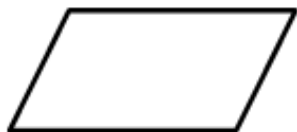


# 1.5 语法图及程序流程图

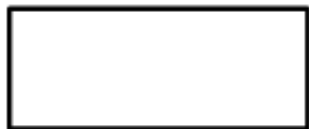
## 程序流程图的画法



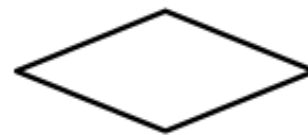
开始终结



输入输出



运算处理



判定转移



页面连接



程序流程



华中科技大学

# 1.6 编译环境的安装与使用

