

计算机引论

山东建筑大学
计算机学院
秦松

9

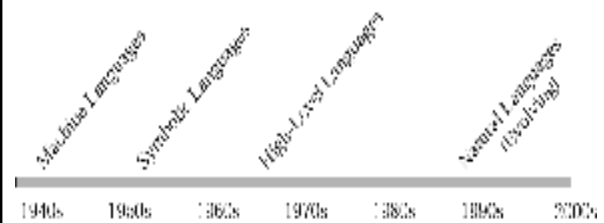
Chapter 9

程序设计语言

演化

计算机语言及evolution: 演化

• **Computer language** : 计算机语言。根据预先定义的规则（语法syntax）而写出的预定语句的结合，这些语句组成程序。



计算机语言

早期没有计算机语言，必须写二进制指令，随着程序越来越大，计算机科学家们提出利用符号来代表二进制模式。



Ada Augusta Byron 世界上第一位软件工程师（以她命名了ADA语言）

Ada

机器语言

- **Machine language** : 机器语言，由“0”和“1”的字符串组成。早期唯一的程序设计语言。
- 指令必须由0、1组成，内部电路是由开关、晶体管及其他电子器件组成，而这些器件只有两种状态：关或开。



注:

计算机唯一理解的语言是机器语言。

Symbolic language符号语言

用符号或助记符代替机器语言，表示不同的机器语言指令。称为**Symbolic language符号语言**

assembler(汇编程序):将符号代码翻译为机器语言的特定程序。

由于符号语言必须被汇编成机器语言成为Assembly language(汇编语言)。

这些语言近似的表示机器语言



符号语言程序（乘法）

```
1  Entry  main, ^m<r2>
2  subl2 #12,sp
3  jsb   C$MAIN_ARGS
4  movab $CHAR_STRING_CON
5
6  pushal -8(fp)
7  pushal (r2)
8  calls #2,read
9  pushal -12(fp)
10 pushal 3(r2)
11 calls #2,read
12 mull3  -8(fp),-12(fp),-
13 pushal 6(r2)
14 calls #2,print
15 clrl  r0
16 ret
```

High-level language高级语言

适应许多不同的计算机，设计目标就是使程序员摆脱汇编语言繁琐的细节。

高级语言与符号语言的共性：compilation(编译):高级语言转化为机器语言的过程。

C 语言程序

```
/* This program reads two integer numbers from the
   keyboard and prints their product. */
#include <stdio.h>
int main (void)
{ /*Local Declarations */
    int number1, number2;
    int result;
    // Statements */
    scanf("%d,%d",&number1,&number2);
    result = number1 * number2;
    printf("The result is:%d", result);
    return 0;
}
```

natural language自然语言

natural language自然语言：理想状况下，类似英语、汉语，目前还在实验室中。

构建程序

Building a program构建程序

程序员的工作是写程序，然后转换为机器语言（可执行文件）。

•过程如下：

(1)writing and editing the program：编写和编辑程序

编写程序的软件是文本编辑器，但程序是一行行代码。程序编好并存盘，文件输入到编译器，称为**源文件Source file**

(2)compiling the program：编译程序

源文件必须翻译成机器语言，计算机才能理解。绝大多数高级语言使用编译器——预处理程序、翻译程序。预处理读源代码，为翻译程序作准备。当准备代码时，预处理程序扫描特殊命令（预处理程序指示），预处理程序查询特殊代码库，在代码中作替换并在其他方面为把代码翻译成机器语言做准备，与处理结果称为翻译单元。

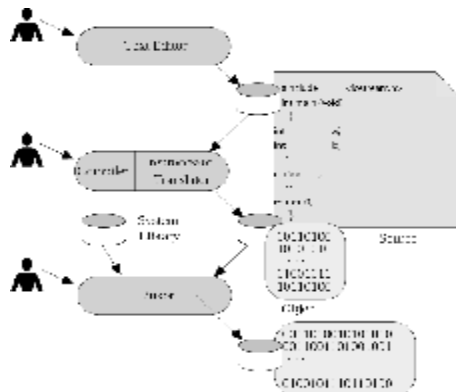
Building a program构建程序

翻译程序读取翻译单元转换为目标模块（机器语言代码），但还不能运行，不具备程序运行的所有部分

(3)Linking the program with the required library modules：用所需的库模块链接程序

子程序和某些输入输出处理必须附加到程序中，链接器，将所有这些函数编写的和系统的汇编到可执行程序中去。

Building a program构建程序



链接程序

一旦程序被链接好或。它就可以执行了

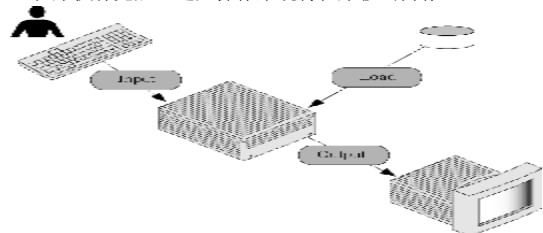


程序的执行

程序的执行

使用操作系统命令，如run，将程序载入内存并执行。江程序载入内存是由操作系统程序——载入程序来完成的。它定位可执行程序，将其读入内存，一切就绪后，控制交给程序，开始执行。

典型：程序载入，读取输入数据，处理数据，输出结果，程序执行完后，通知操作系统将程序移出内存



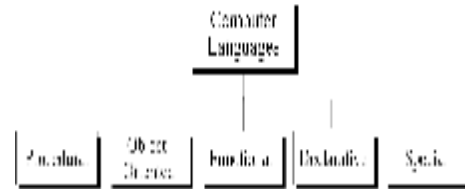
9.4

语言的分类

语言的分类

计算机语言根据其解决问题的方法及所解决问题的种类分为五类：

- procedural：过程化语言
- Object-Oriented：面向对象的语言
- Functional：函数型语言
- Declarative：说明型语言
- Special：专用语言



过程化（强制性）语言

•Procedural (imperative) languages 过程化（强制性）语言，使用传统方法编程，它采用与计算机硬件执行程序相同的方法（取指令、译指令、执行）来执行程序，过程化语言是一套指令，顺序执行除非有强制控制，逐条执行，可能有重复执行或略过。

今年发展的一些高级语言：FORTRAN、COBOL、Pascal、C、Ada等。

面向对象的语言

Object-Oriented（面向对象）与过程化编程不同，语言中的数据项可以当作对象，程序当作在对象上执行的一系列操作。

面向对象编程，对象和操作是绑定在一起的，程序员首先定义对象和可以应用于该对象的数据类型，调用绑在一起的操作解决问题。程序员可以利用这个结合体，调用一些或全部已定义的操作来解决问题。对象是主动的。

面向对象的语言，例如C++、Java

函数型语言

函数型语言：在函数程序设计中，程序当作函数来考虑，实现功能：可定义一系列基本（原子）函数

允许程序员通过组合基本函数创建新的函数

Lisp、Scheme是函数性语言。

