

由机器语言到高级语言

战德臣

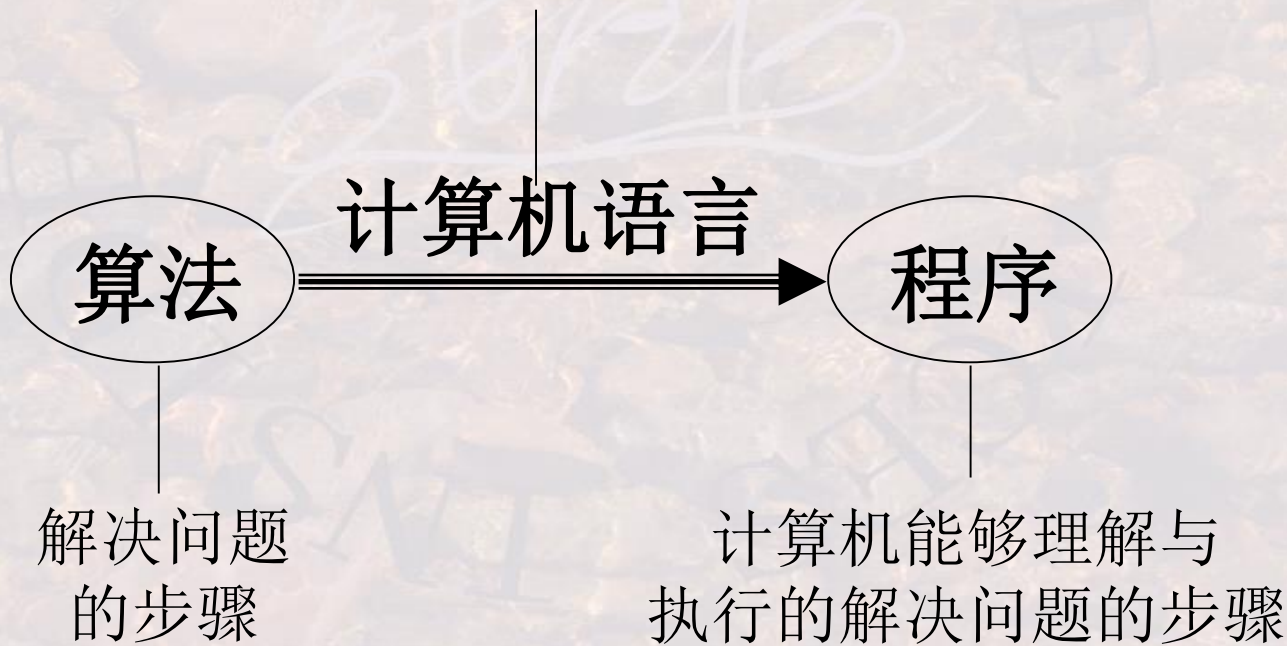
哈尔滨工业大学 教授.博士生导师
教育部大学计算机课程教学指导委员会委员



**Research Center on Intelligent
Computing for Enterprises & Services,
Harbin Institute of Technology**

算法、计算机语言与计算机程序

步骤书写的规范、语法规则、标准的集合
是人和计算机都能理解的语言



计算机语言---机器语言

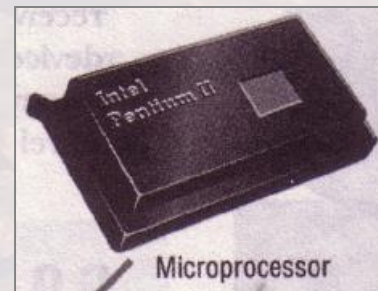
指令系统

指令系统： CPU用二进制和编码提供的可以解释并执行的命令的集合。

操作码

地址码

100001	10 00000111
100010	11 00001010



计算7+10并存储的程序

100001	10
00000111	
100010	10
00001010	
100101	11
00000110	
111101	00

机器语言

机器语言： 用二进制和编码方式提供的指令系统所编写程序的语言被称为机器语言

所有程序都需转换成机器语言程序，计算机才能执行

问：用机器语言编写程序存在什么问题呢？

计算机语言---汇编语言

◆用符号编写程序 ==> 翻译 ==> 机器语言程序

◆人们提供了用助记符编写程序的规范/标准。同时开发了一个翻译程序，实现了将符号程序自动转换成机器语言程序的功能。



操作码	地址码
100001	1000000111
↓	
MOV A, 7	

计算7+10并存储的程序

```
MOV A, 7
ADD A, 10
MOV (6), A
HLT
```

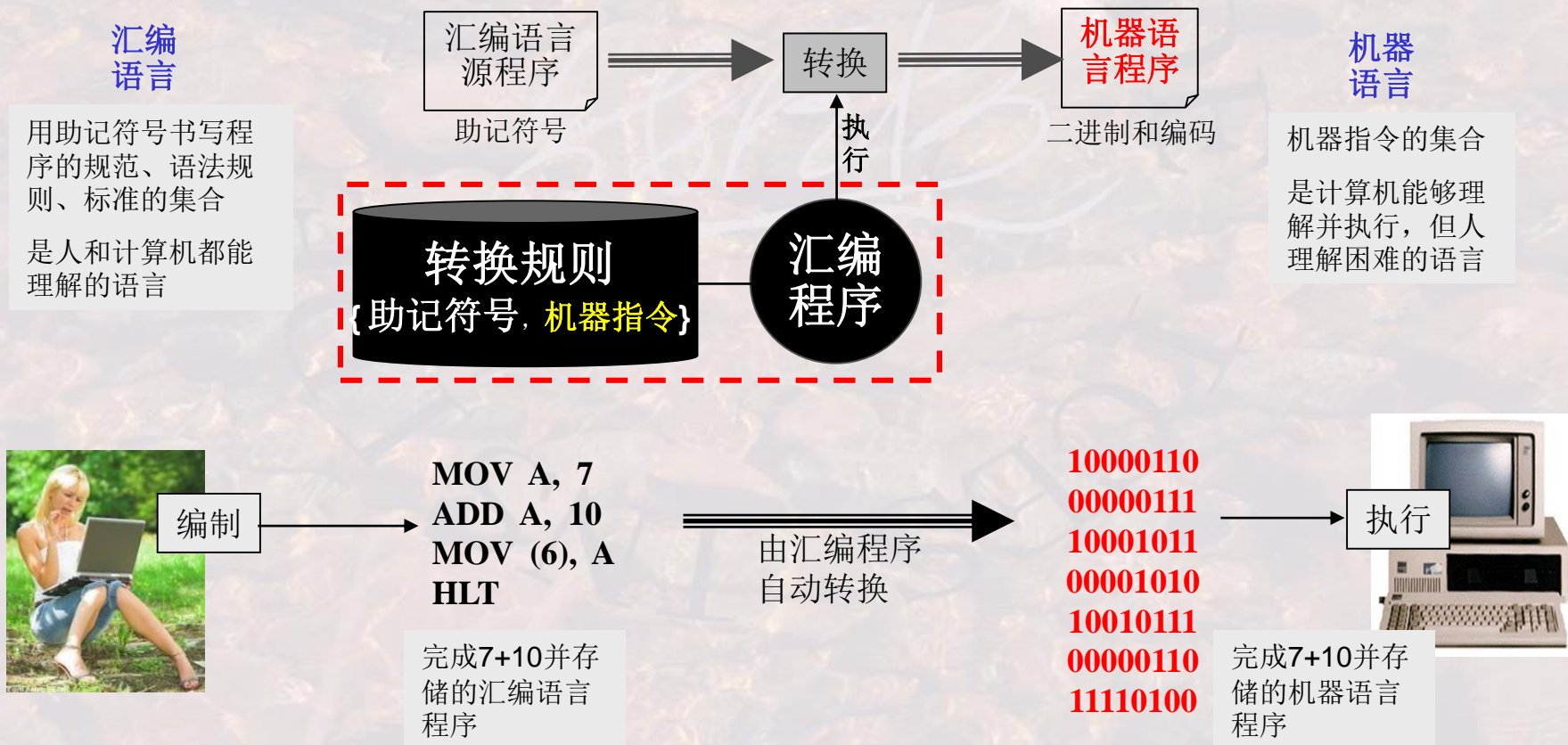
◆汇编语言：是用助记符编写程序的语言。

◆汇编语言源程序：是用汇编语言编出的程序。

◆汇编程序：是将汇编语言源程序翻译成机器语言程序的程序。

计算机语言---汇编语言---汇编程序(编译器)

◆汇编语言程序处理过程



由机器语言到高级语言

(5)为什么还要提出高级语言?

计算机语言---高级语言

◆人们提供了类似于**自然语言方式**、以**语句为单位**书写程序的**规范/标准**。并开发了一个**翻译程序**，实现了将语句程序**自动**翻译成机器语言程序的功能。



计算7+10并存储的程序

```
Result = 7+10;  
Return
```

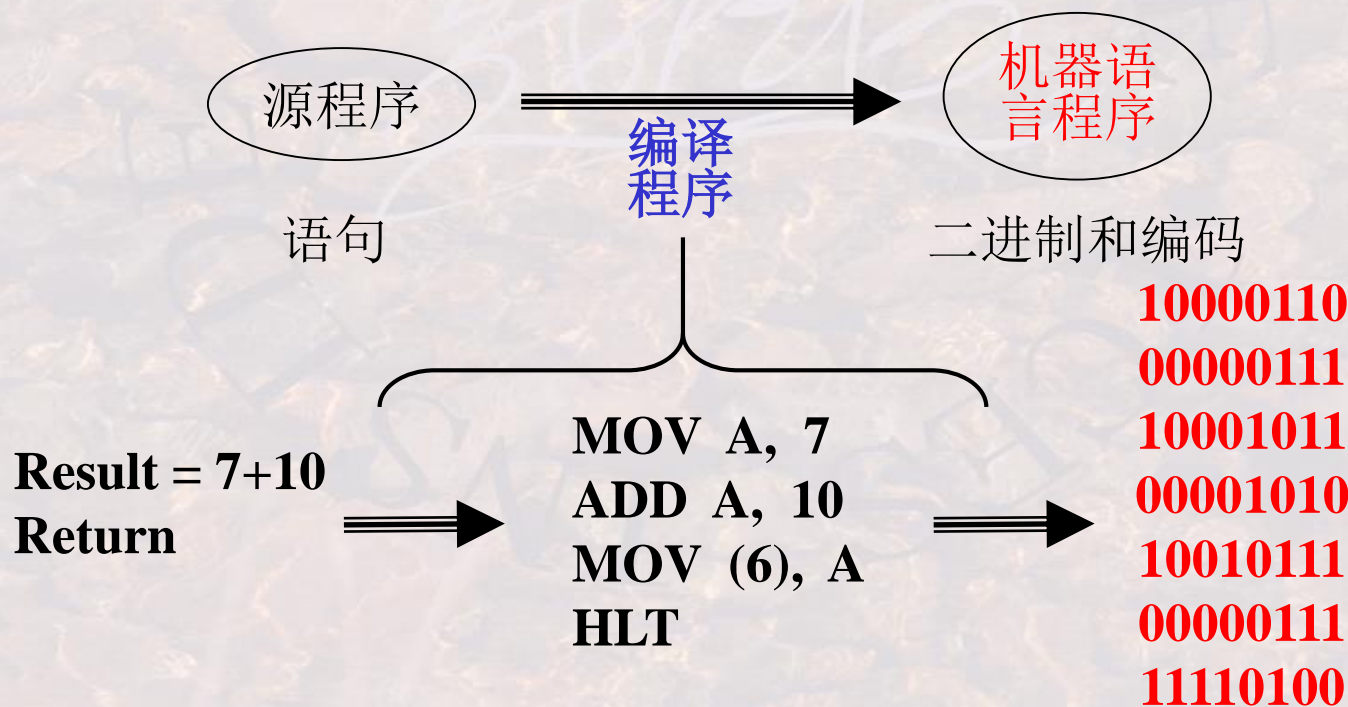
- ◆**高级语言**：是用类似自然语言的语句编写程序的语言。
- ◆**高级语言源程序**：是用高级语言编出的程序。
- ◆**编译程序**：是将高级语言源程序翻译成机器语言程序的程序。

(6)高级语言和汇编语言的差别在哪里？

◆**高级语言**：机器无关性；一条高级语言语句往往可由若干条机器语言语句实现且不具有对应性

◆**汇编语言**：机器相关性；汇编语言语句和机器语言语句有对应性

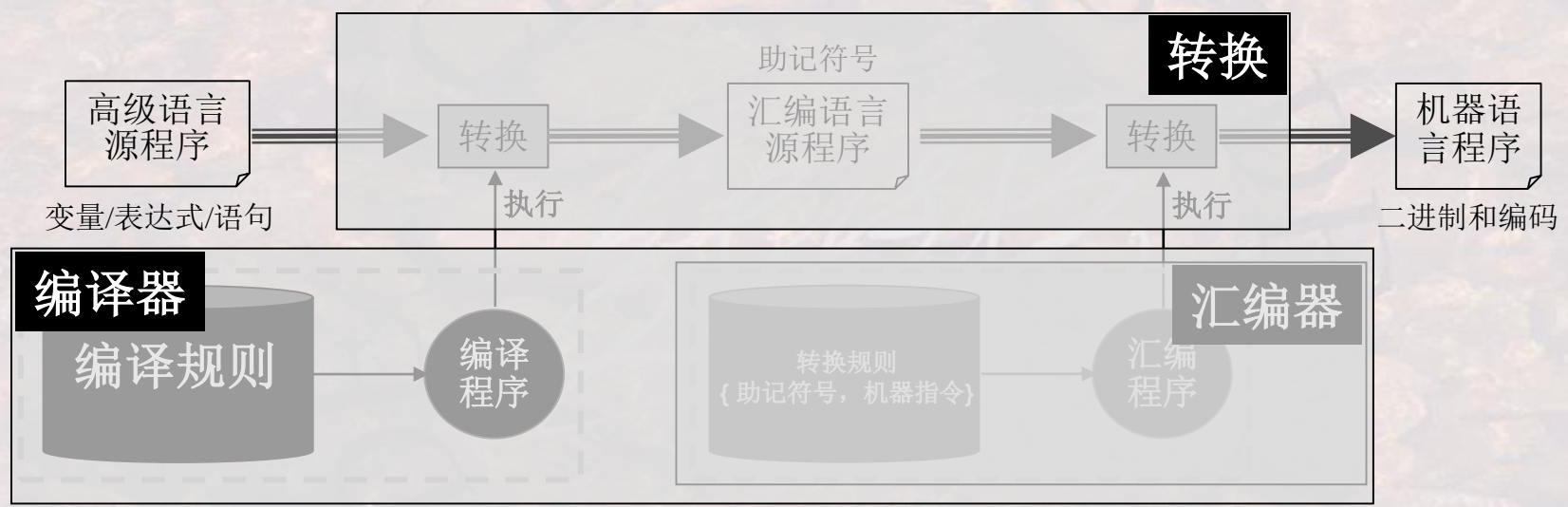
高级语言程序处理过程示意



由机器语言到高级语言

(6)编译器如何实现呢？

高级语言编译器



高级语言

编制

Result = 7+10
Return

自动转换

转换

MOV A, 7
ADD A, 10
MOV (6), A
HLT

自动转换

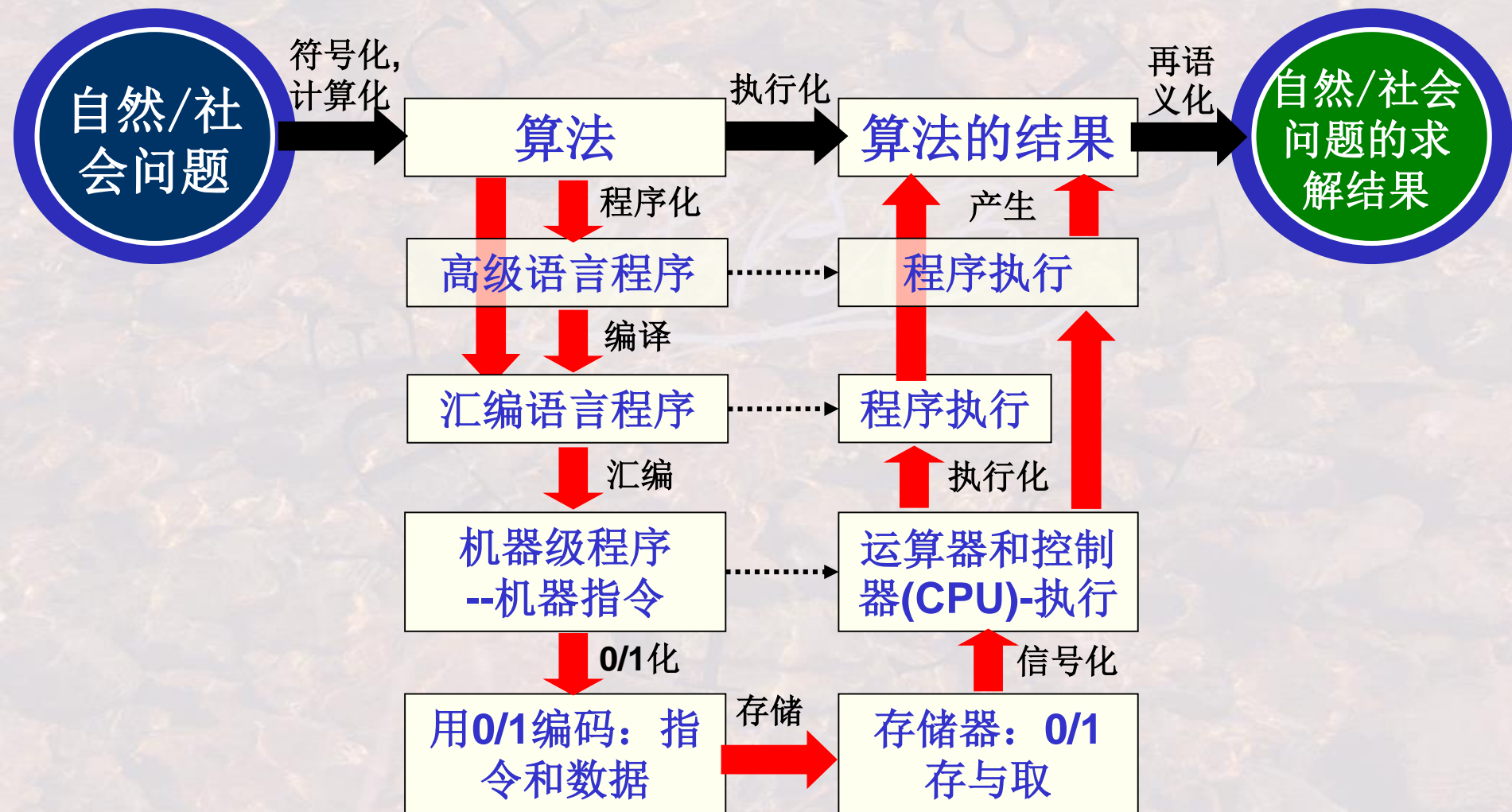
10000110
00000111
10001011
00001010
10010111
00000110
11110100

执行



机器语言

用高级语言进行问题求解



高级语言(程序)的 基本构成要素

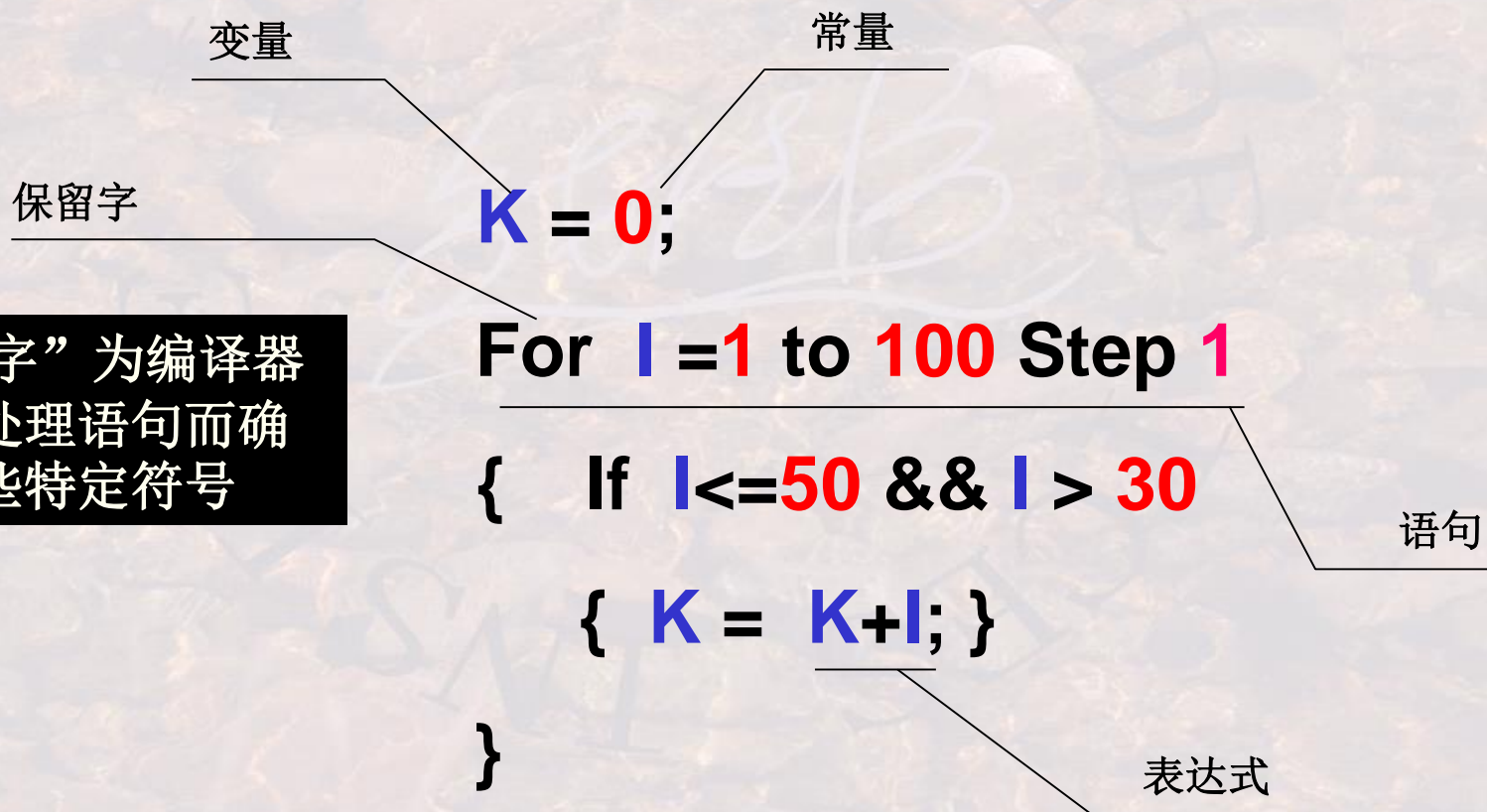
战德臣

哈尔滨工业大学 教授.博士生导师
教育部大学计算机课程教学指导委员会委员



**Research Center on Intelligent
Computing for Enterprises & Services,
Harbin Institute of Technology**

认识计算机语言程序



“保留字”为编译器识别和处理语句而确定的一些特定符号

常量、变量与表达式

- ◆算术表达式示例。算术表达式的结果是一数值；

$A1 + (B2 - x1 + 76) * 3$

$(B2 + yy4) / L3 - xx3$

- ◆比较表达式示例。比较表达式的计算结果是逻辑“真”或“假”；

$Grade < 90$

$Grade \geq 70$

$N4 < A1 + B2 + 20$ //注： $A1+B2+20$ 为算术表达式，计算完后再与 $N4$ 的值进行比较

- ◆逻辑表达式示例。逻辑表达式的计算结果是逻辑“真”或“假”；

$(x1 \geq A1) \&\& (B2 \lt y2)$

- ◆将表达式的计算结果赋值给一变量：赋值语句

$M = X > Y + 50;$

$M = (X > Y) \text{ AND } (X < Y);$

$K = K + (5 * K);$

$A1 + (B2 - x1 + 76) * 3$



$(+ A1 (* (+ (- B2 x1) 76) 3)$

语句与程序控制

顺序 结构

◆顺序结构

```
G5 = 1;  
G6 = 2;  
G7 = 3;  
G8 = 4;  
G9 = 5;  
G9 = G9 + G8;  
G9 = G9 + G7;  
G9 = G9 + G6;  
G9 = G9 + G5;
```

程序执行示例

```
G5 = 1;  
G6 = 2;  
G7 = 3;  
G8 = 4;  
G9 = 5;  
G9 = G9 + G8;  
G9 = G9 + G7;  
G9 = G9 + G6;  
G9 = G9 + G5;
```

G5	1
G6	2
G7	3
G8	4
G9	15

语句与程序控制

分支 结构

◆分支结构

IF 条件表达式 {
 (条件为真时运行的)程序语句序列1 }
ELSE {
 (条件为假时运行的)程序语句序列2 }

```
If D1>D2
{   D1=D1-5; }
Else
{   D1=D1+10; }
```

```
Y = 50;
Z = 80;
X = 30;
X = Z + Y;
If Y > Z {
    X = X - Y; }
Else {
    X = X - Z; }
X = X + Y;
If X > Z { X = Y; }
X = X - Z;
If X > Y
{ X = X - Y; }
```


语句与程序控制

分支 结构

X	530
Y	50
Z	80

Y = 50;

Z = 80;

X = 30;

X = Z + Y;

If Y > Z {
 X = X - Y; }

Else {
 X = X - Z; }

X = X + Y;

If X > Z { X = Y; }

X = X - Z;

If X > Y
{ X = X - Y; }

语句与程序控制

循环结构

◆循环结构(有界循环结构)

For (计数器变量 = 起始值 **To** 结束值 [增量表达式])

{ 循环体的程序语句序列 }

Next [计数器变量]

Sum=0;

For I = 1 to 5 Step 1

{ Sum = Sum + I; }

Next I

//继续其他语句

Sum=0;

For I =1 to 10000 Step 2

{ Sum = Sum + I; }

Next I

Sum

00

I

1

语句与程序控制

循环结构

◆循环结构(条件循环结构)

Do

{ 循环体的程序语句序列 }

While (条件表达式);

X	1
Y	1
Sum	1

X=1;

Y=2;

Sum=0;

Do {

Sum = X+Y;

X=X+1;

Y=Y+1;

} While (Sum<=10)

//其他语句

语句与程序控制

循环结构

◆循环结构(条件循环结构)

Do

{ 循环体的程序语句序列 }

While (条件表达式);

X	2
Y	2
Sum	0

X=1;

Y=2;

Sum=0;

Do {

Sum = X+Y;

X=X+1;

Y=Y+1;

} While (Sum<0)

//其他语句

语句与程序控制

循环结构

◆循环结构(条件循环结构)

While (条件表达式)

Do { 循环体的程序语句序列 }

X	1
Y	2
Sum	0

X=1;

Y=2;

Sum=0;

While (Sum<0)

Do {

Sum = X+Y;

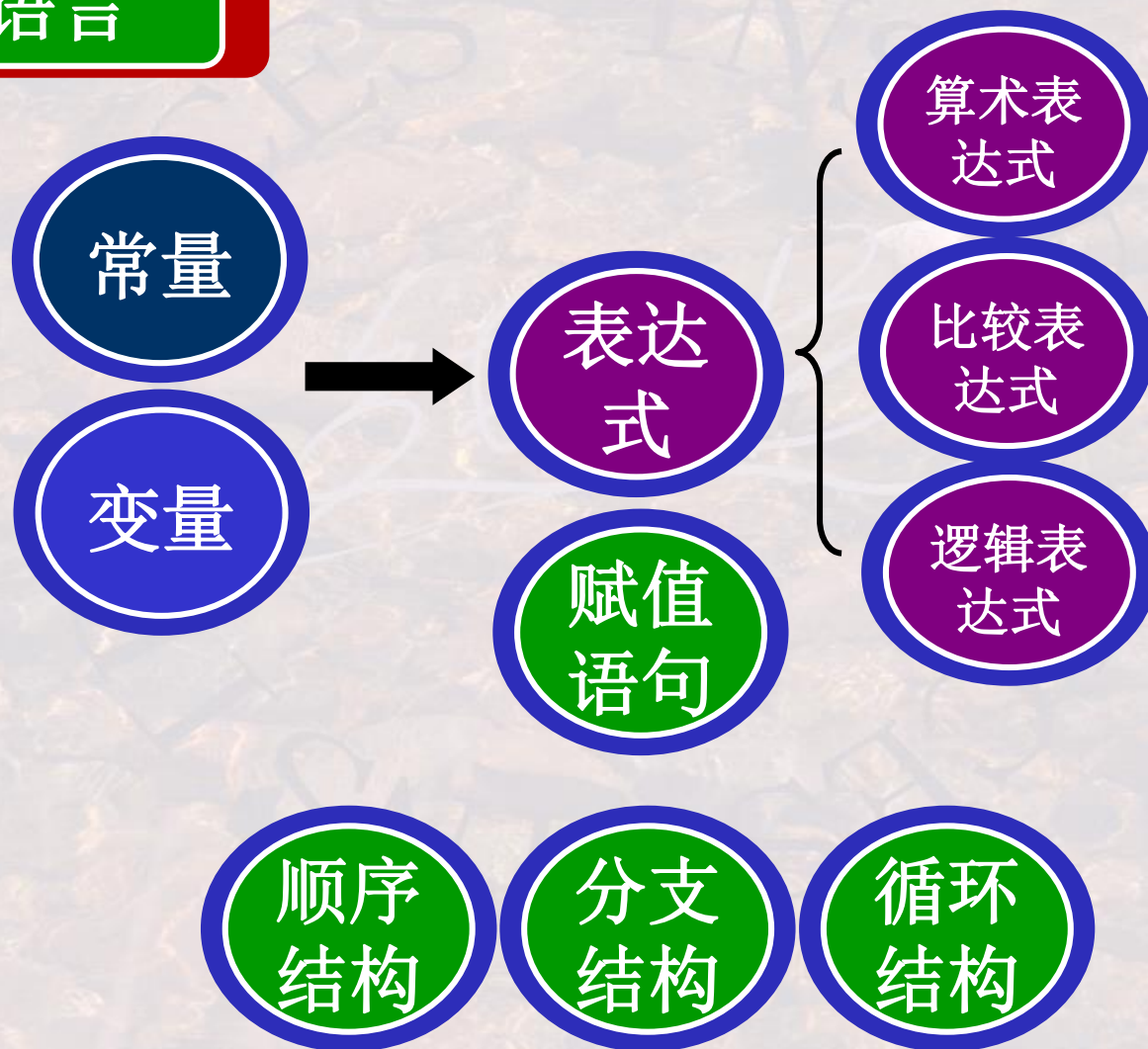
X=X+1;

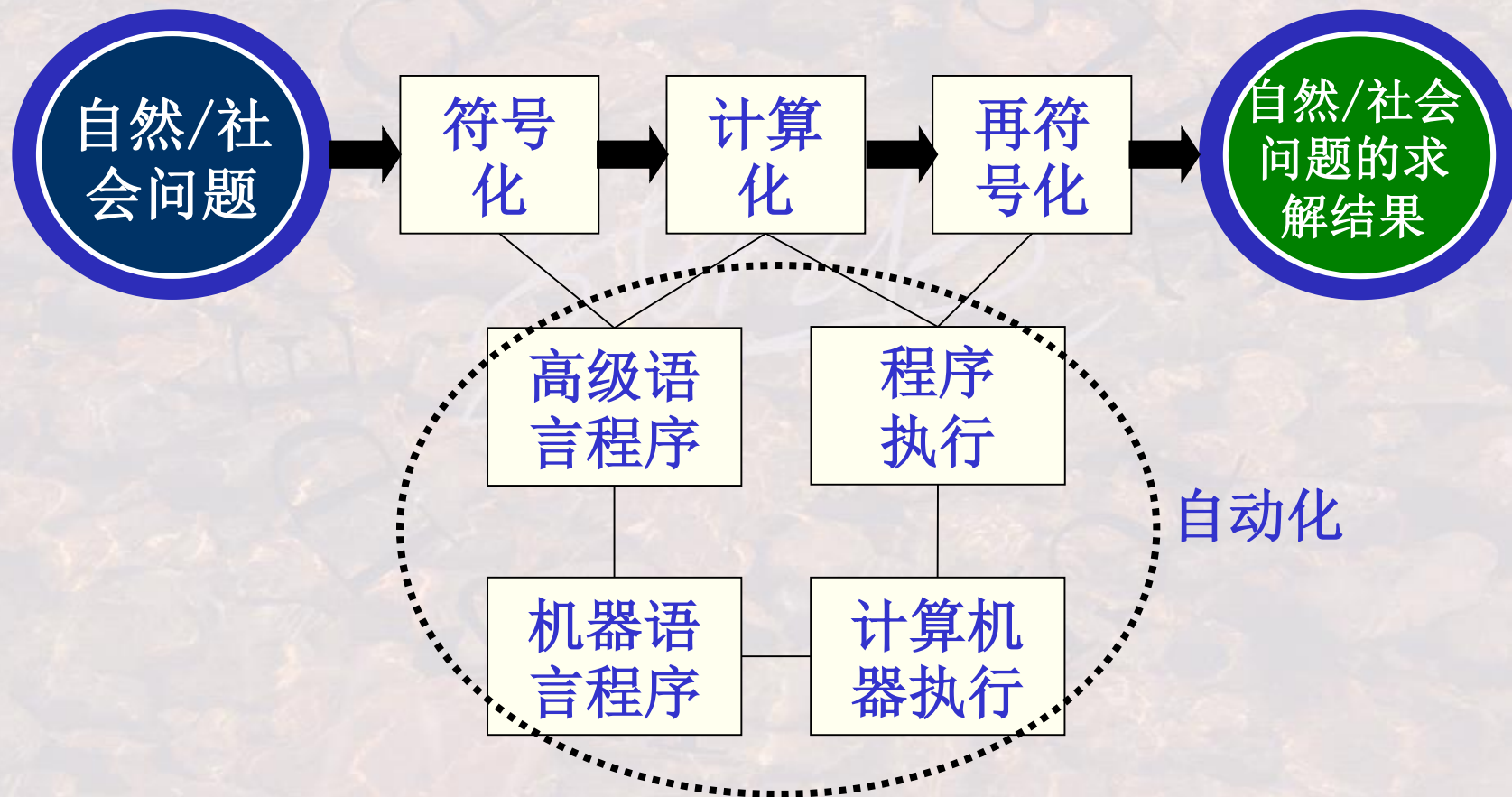
Y=Y+1;

}

<其他语句>

高级语言





用高级语言构造程序

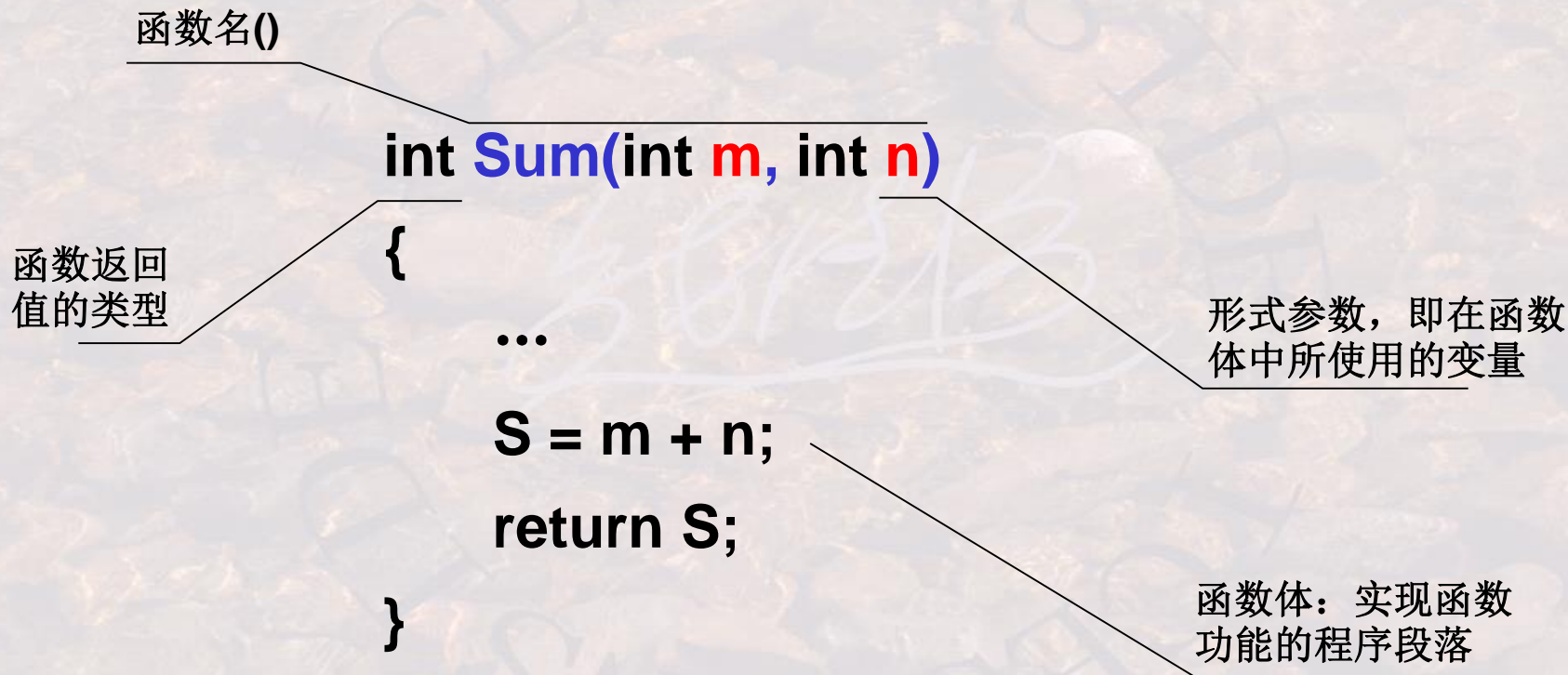
战德臣

哈尔滨工业大学 教授.博士生导师
教育部大学计算机课程教学指导委员会委员



**Research Center on Intelligent
Computing for Enterprises & Services,
Harbin Institute of Technology**

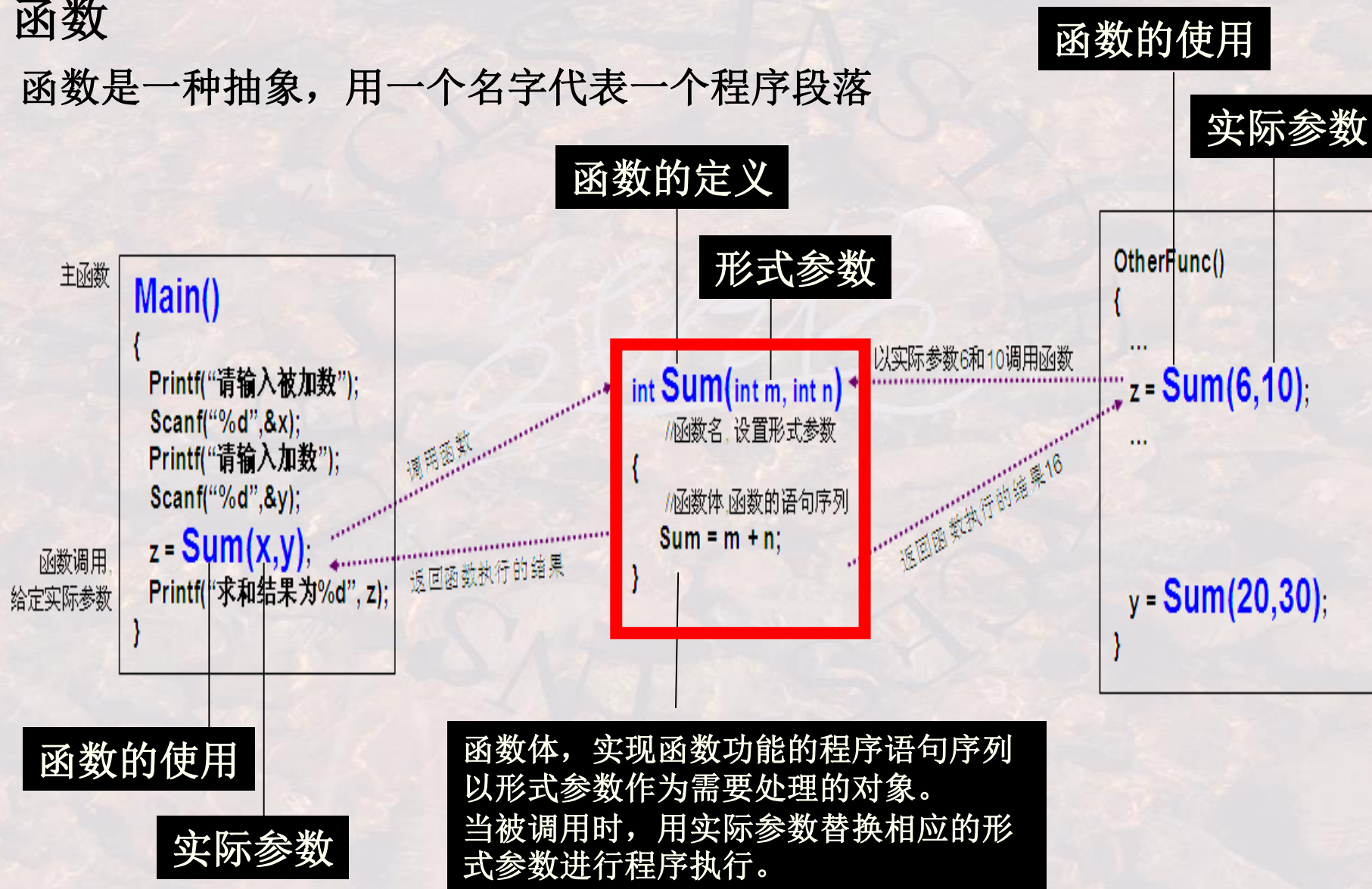
函数



数学上的函数只是一个符号表达，而计算机程序中的函数则是一段可以执行的程序

函数

函数是一种抽象，用一个名字代表一个程序段落



(3)你知道计算机语言或操作系统提供哪些函数吗？

系统提供的可以使用的函数类别

- **数学运算函数**，如三角函数、指数与对数函数、开方函数等；例如 $\sin(\alpha)$ ， $\text{Log}(x)$ 等；
- **数据转换函数**，如字母大小写变换、数值型数字和字符型数字相互转换等；
- **字符串操作函数**，如取子串、计算字符串长度等；例如， $\text{Len}(\text{"abcd"})$ ；
- **输入输出函数**，如输入输出数值、字符、字符串等；例如， $\text{Printf}(\cdots)$ ， $\text{Scanf}(\cdots)$ 等；
- **文件操作函数**，如文件的打开、读取、写入、关闭等；
- **其它函数**，如取系统日期、绘制图形等。

程序示例：阶乘的递归程序如下示意

```
long int Fact(int n)
```

```
{ long int x;
```

```
  if (n > 1)
```

```
  { x = Fact(n-1);
```

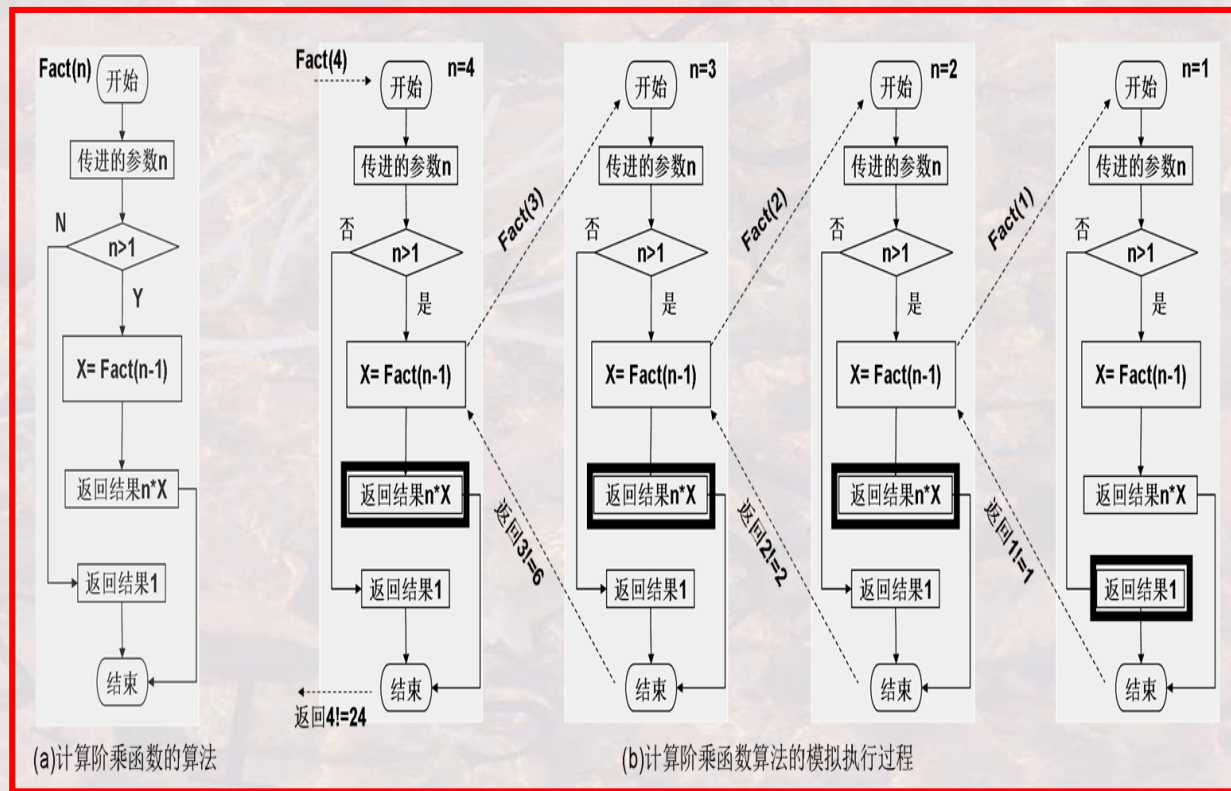
```
    /*递归调用*/
```

```
    return n*x; }
```

```
  else return 1;
```

```
  /*递归基础*/
```

```
}
```



程序示例：阶乘的迭代程序如下示意

```
long int Fact(int n)
{   int counter;
    long product = 1;
    for counter = 1 to n step 1
    { product = product * counter; }
    /*迭代*/
    return product;
}
```

$$n! = \begin{cases} 1 & \text{当 } n \leq 1 \text{ 时} \\ n \times (n-1) \times \dots \times 1 & \text{当 } n > 1 \text{ 时} \end{cases}$$

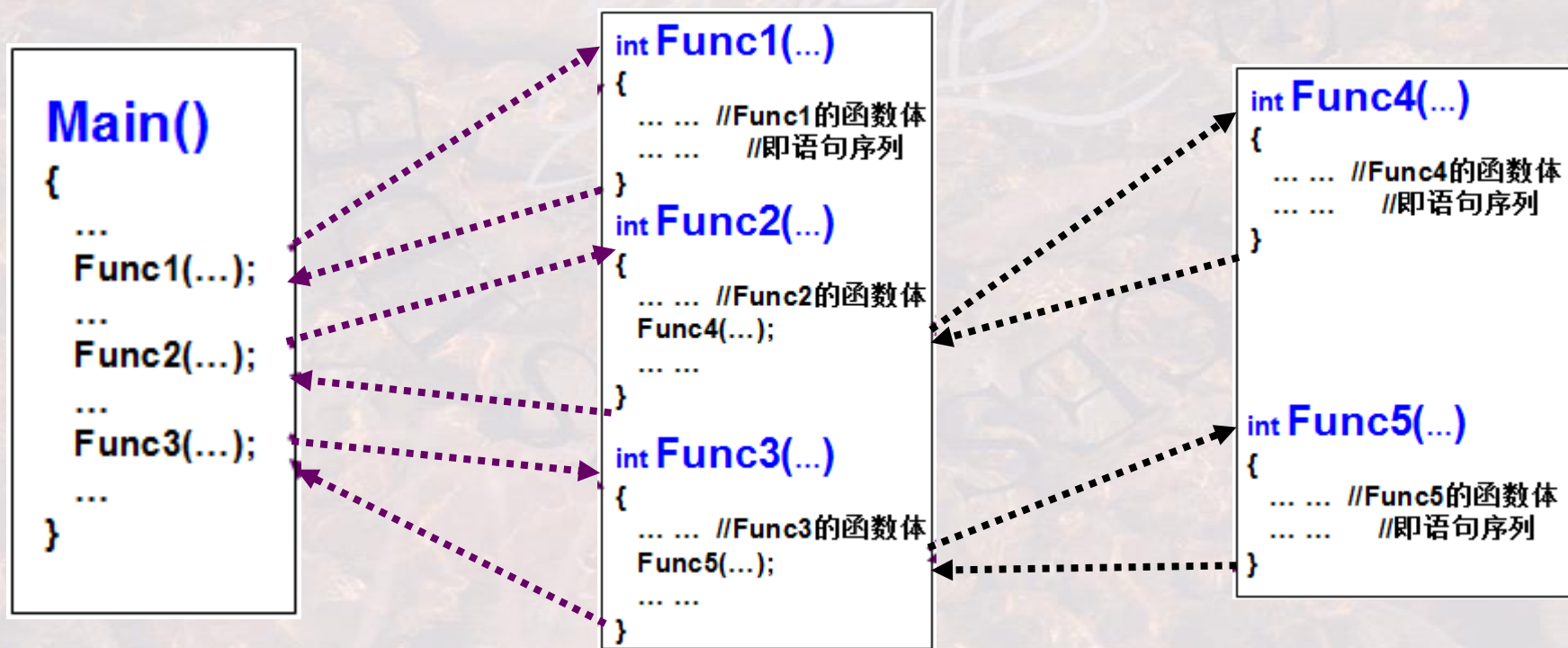
	Product	Counter
初始值	1	
循环第1次	1	1
循环第2次	1	2
循环第3次	2	3
循环第4次	6	4
循环第5次	24	5
循环第6次	120	6

传统程序构造及其表达方法----由粗到细

为控制复杂性，先以函数来代替琐碎的细节，着重考虑函数之间的关系，以及如何解决问题



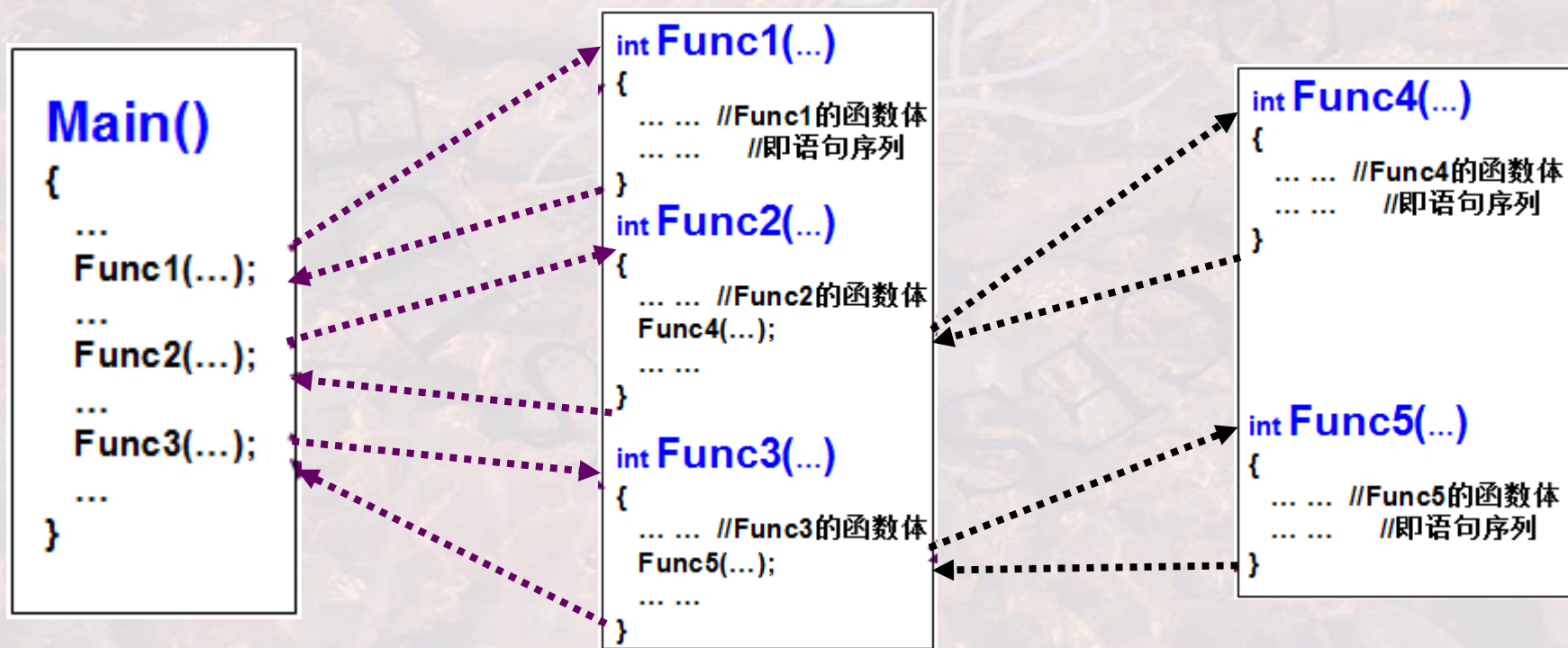
在前一阶段考虑清楚后或编制完成后，再编写其中的每一个函数。而函数的处理同样采取这种思路



传统程序构造及其表达方法----也可以由细到粗

上一层次的函数依据下层函数来编写，确认正确后再转至更上层问题处理

首先编写一些基础性的函数，并确定其正确后，再处理上一层次的问题。

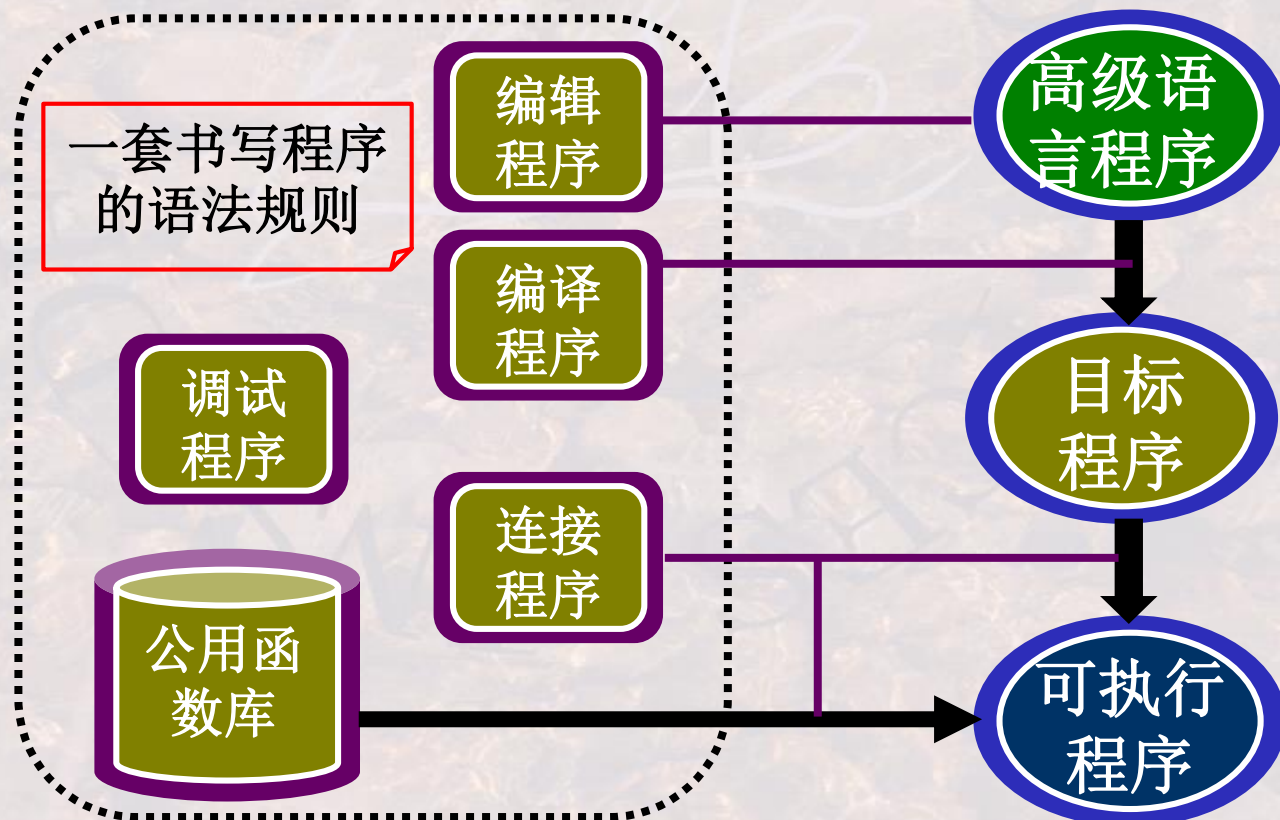


程序开发环境

◆ **程序**是算法的一种机器相容(Compatible)的表示，是利用计算机程序设计语言对算法描述的结果，是可以在计算机上执行的算法。

◆ **程序设计过程**：编辑源程序→编译→链接→执行。

计算机
语言程序
设计环境：
编辑、编
译、连接、
调试、运
行一体化
平台





计算机语言与编译器

--一种抽象-自动化机制示例

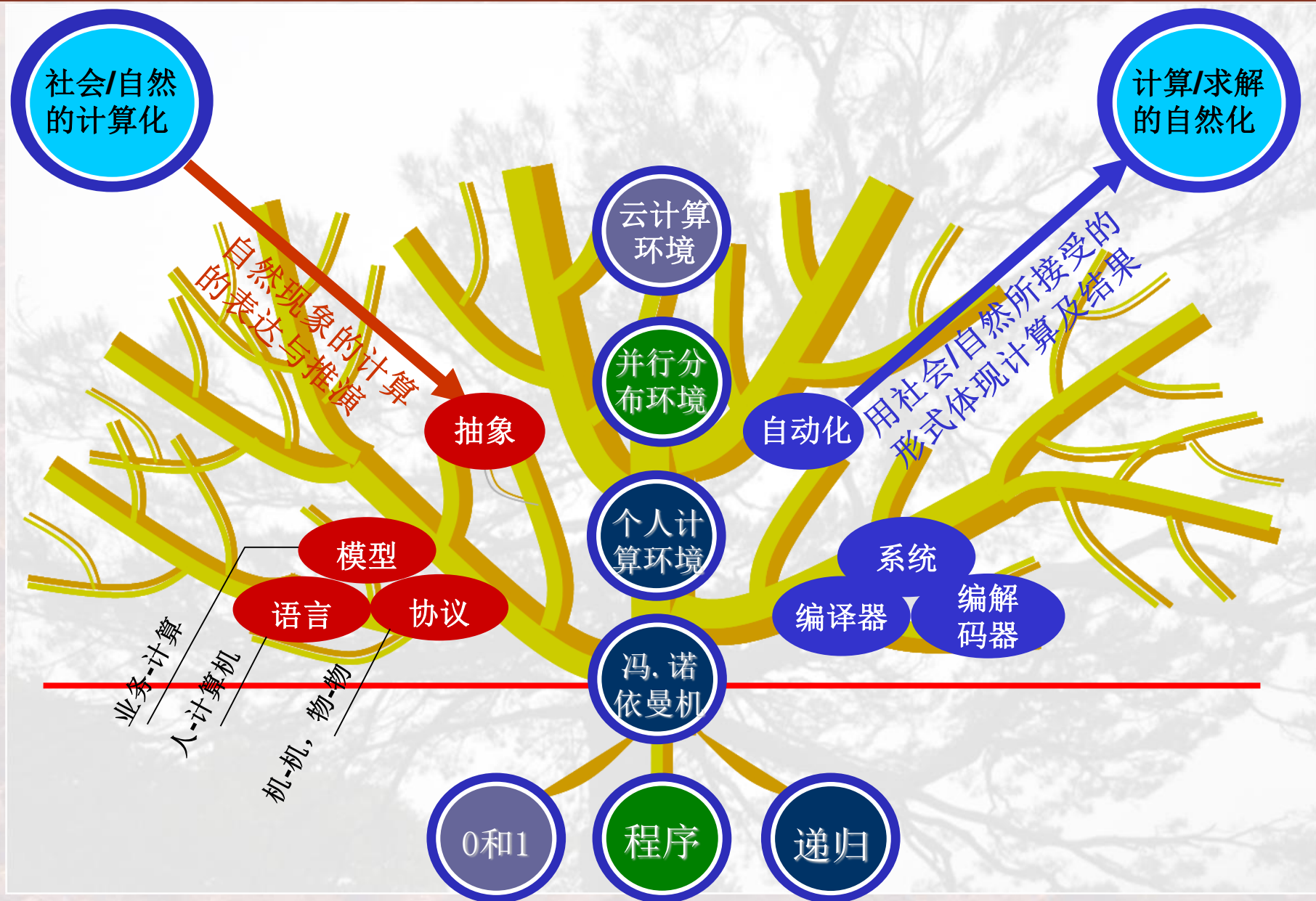
战德臣

哈尔滨工业大学 教授.博士生导师
教育部大学计算机课程教学指导委员会委员

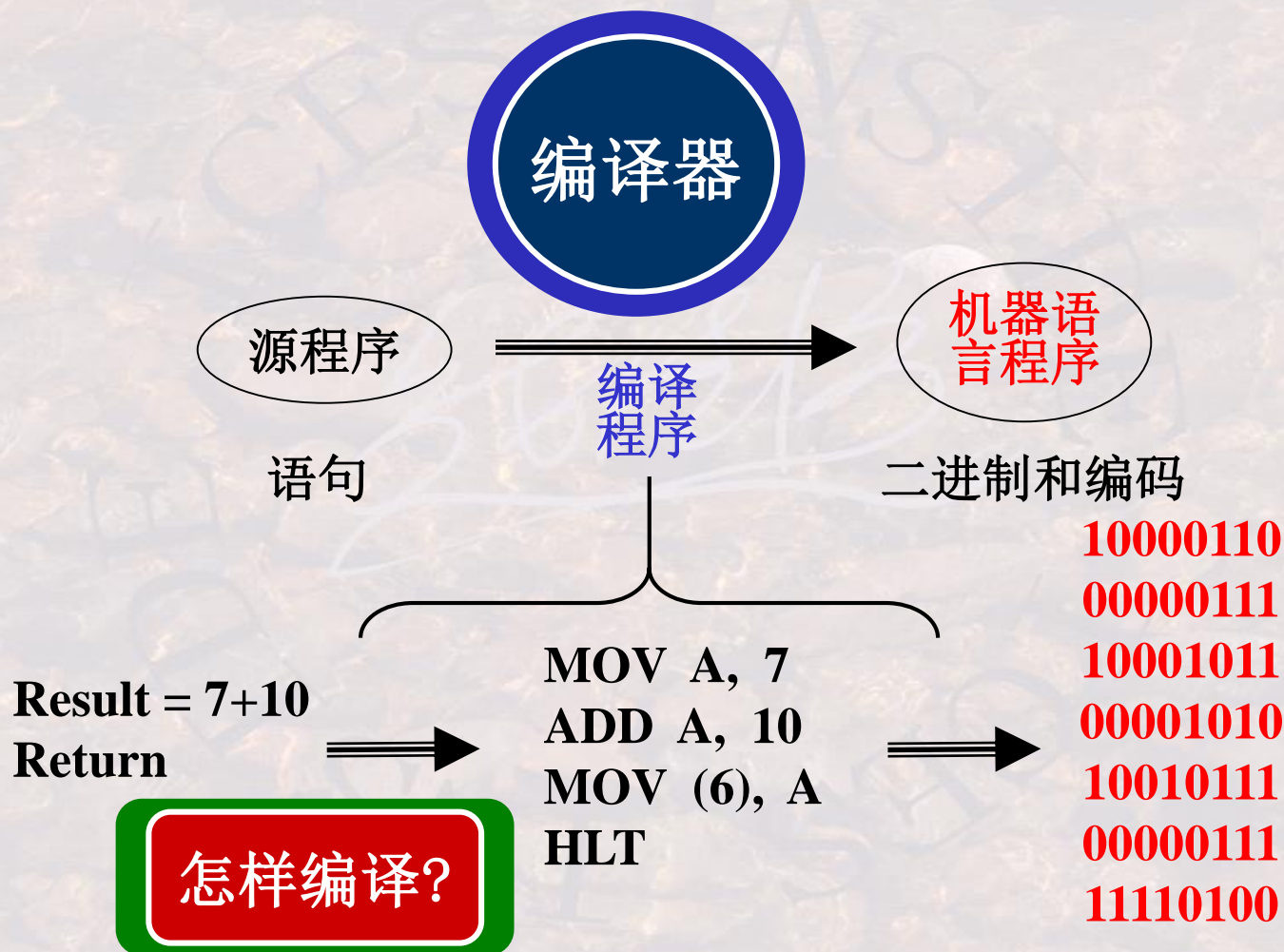


Research Center on Intelligent
Computing for Enterprises & Services,
Harbin Institute of Technology

(1)你记得计算之树中的不同抽象层次吗？



(2)为什么高级语言程序需要编译?



(3)高级语言中的模式化的语句?

由“**具体的**”运算式到“**模式**”运算式

Result = 7 + 10;

Sum = 8 + 15;

K = 100 + 105;

... ..

V = C + C;

注:

Result: 具体的变量
7, 10: 具体的

变化的部分

注:

V: 变量
C: 常量

= 赋值符号
+ 加法运算符号
; 语句结束符

不变的部分
(保留字)

= 赋值符号
+ 加法运算符号
; 语句结束符

(4)语句模式的识别

“模式” 运算式的识别及常量、变量的标识

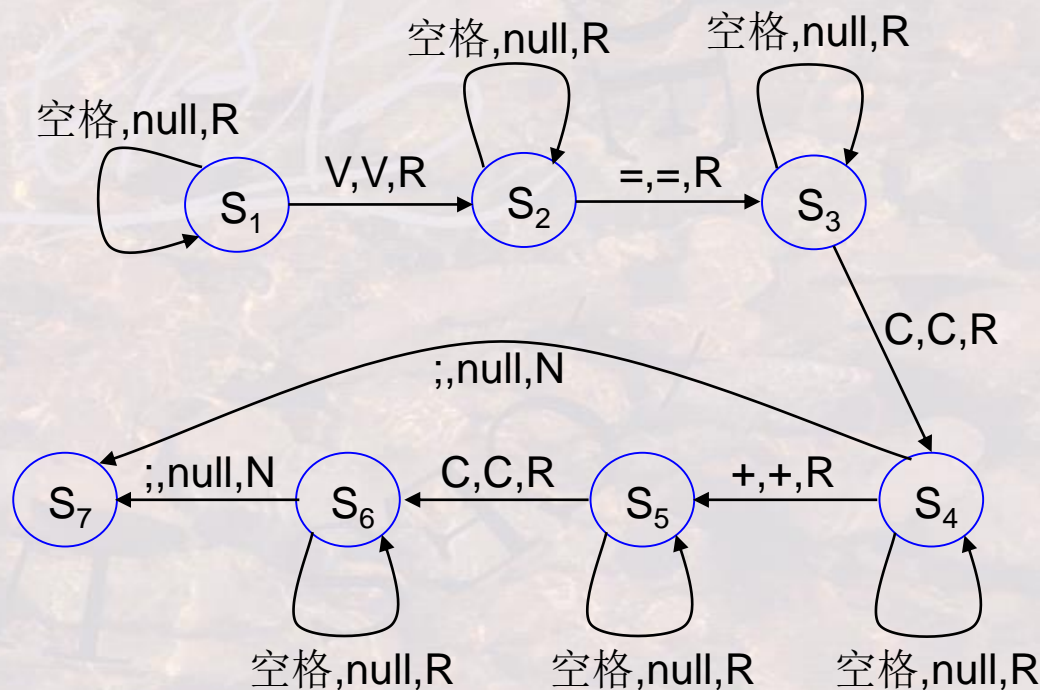
V = C + C;

注：字母表{V, C, =, +, 空格, ;}; S₁起始状态; S₇终止状态; null表示什么也不写回。

Result = 7 + 10;



(V, 1) = (C, 1) + (C, 2);

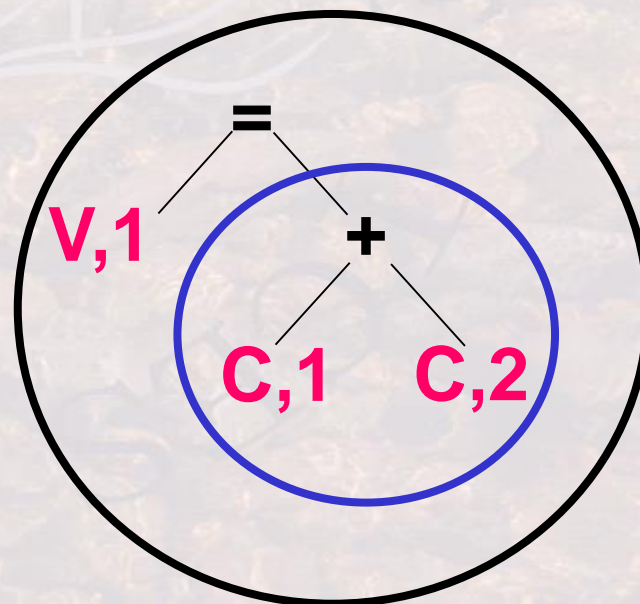
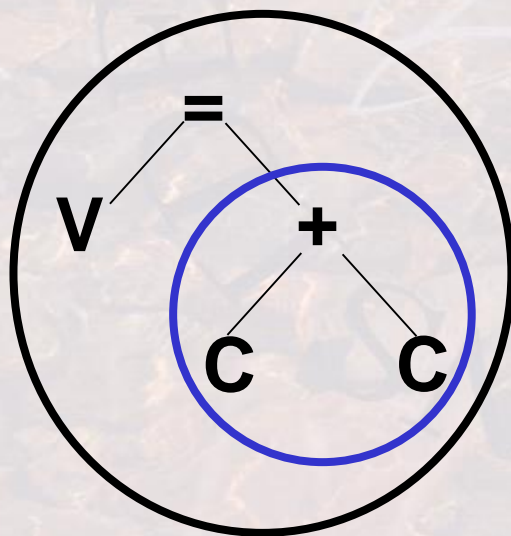


(c)能识别两种模式 “V=C;”和
“V=C+C;”并能去除空格的图灵机示
意图

(5)复杂模式的预先构造

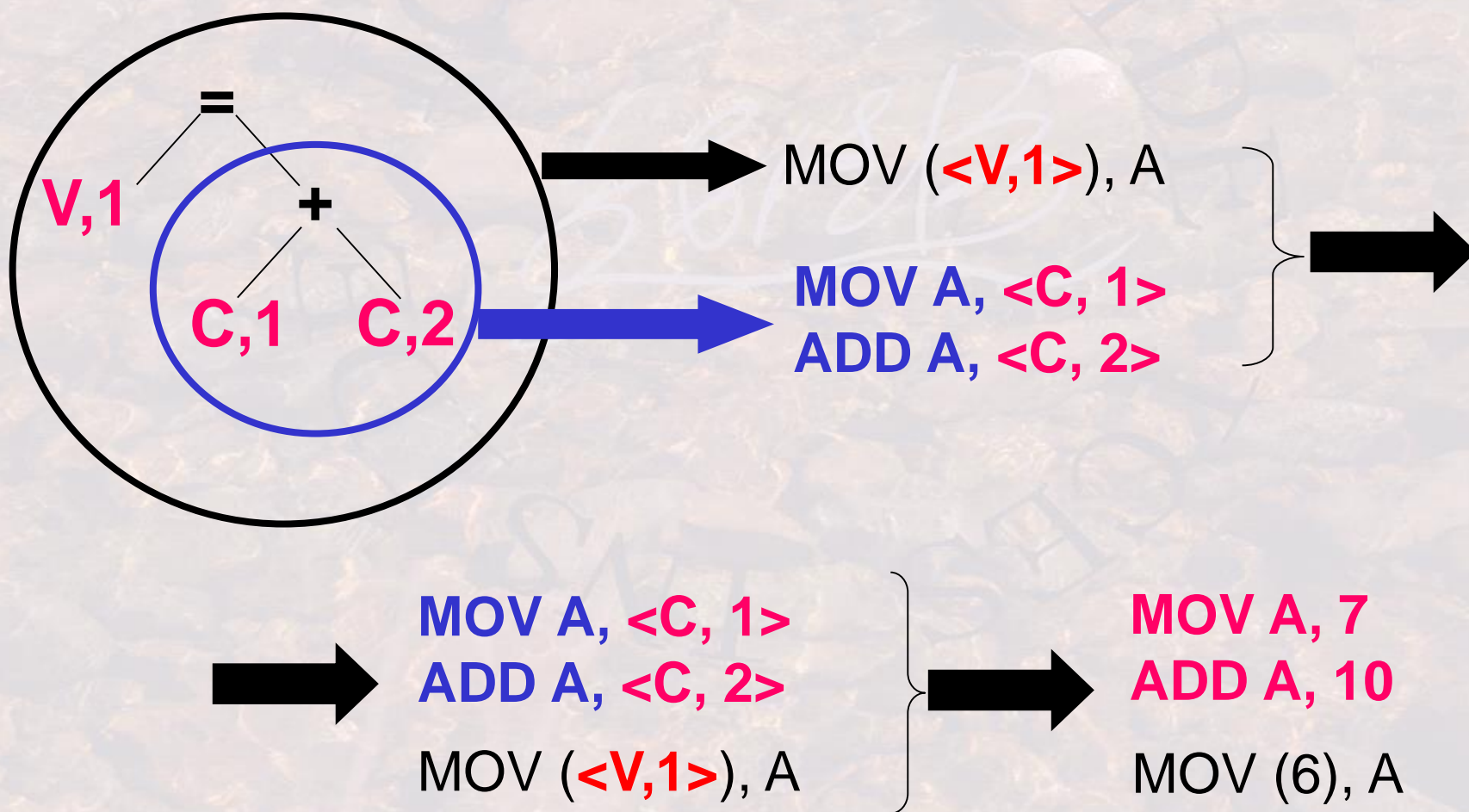
复杂模式转换为简单模式及其组合

$$V = C + C;$$



(6)简单模式与汇编语句的映射

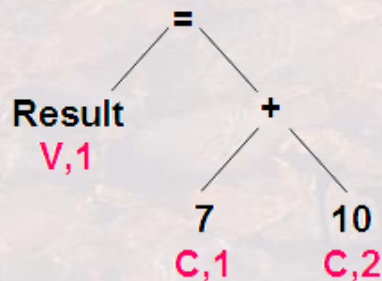
将简单模式转换成汇编语言语句序列，用常量值和变量地址进行替换，组合次序调整，得到最后的汇编语言程序



(7)小结

Result = 7 + 10;

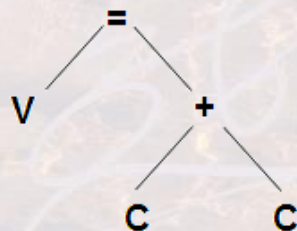
注：
Result: 具体的变量
7, 10: 具体的常量
=: 赋值符号
+: 加法运算符号



(a)一种具体的语句及其解析结构

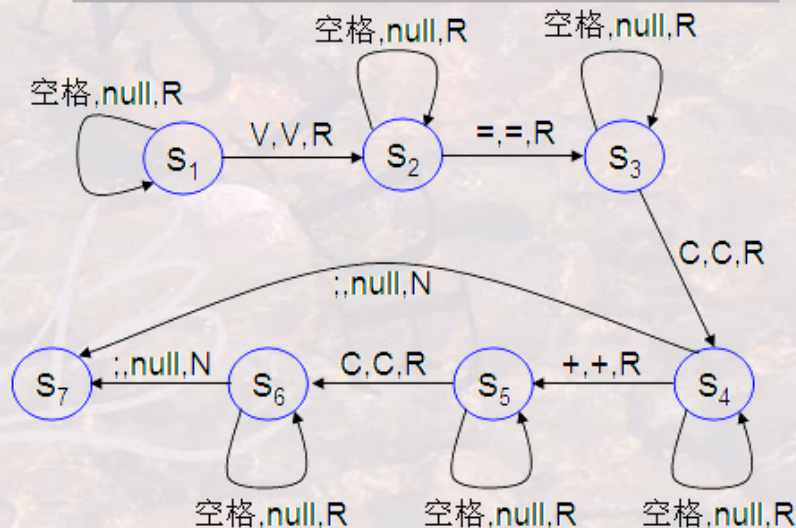
V = C + C;

注：
V: 变量
C: 常量
=: 赋值符号
+: 加法运算符号

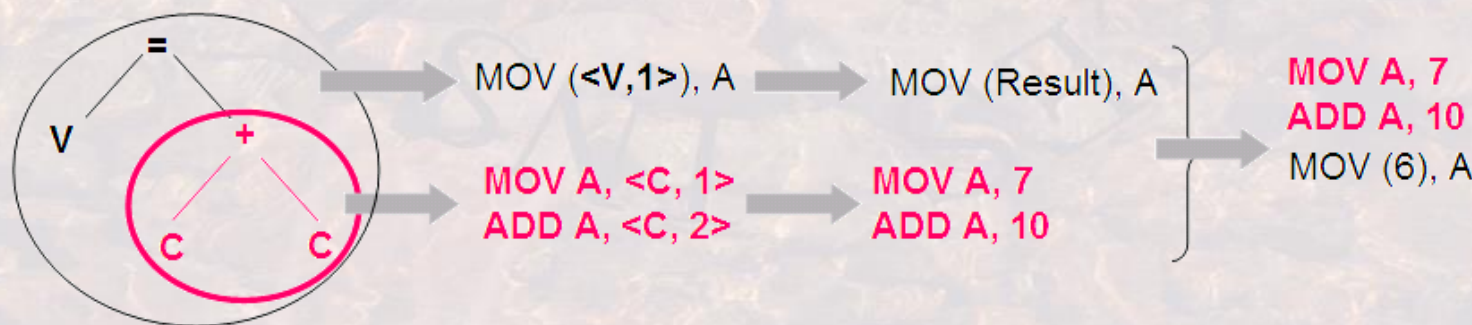


(b)图(a)所示语句的一种模式及其解析结构

注: 字母表{V, C, =, +, 空格, ;}; S₁起始状态; S₇终止状态; null表示什么也不写回。



(c)能识别两种模式“V=C;”和“V=C+C;”并能去除空格的图灵机示意图



(d)语法分析树转换成汇编语言语句的过程示意

(7)小结

Result = 7 + 10;

语句词汇的识别
(词法分析)

V = C + C;

注：
V: 变量
C: 常量
=: 赋值符号
+: 加法运算符

语句模式的识别
(语法分析)

形式语言
与自动机

编译系统/
编译原理



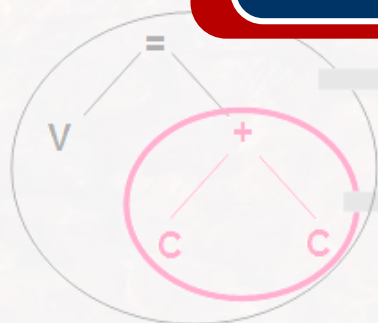
(a)一种具体的语句及其解析结构

复杂语句模式的构造

基本语句模式与
汇编语句的映射

汇编语句的组
装与次序调整

常量与变
量的替换



(d)语法分析树转换成汇编语言语句

注: 字母表{V, C, =, +, 空格, null};
终止状态; null表示什么也不



"V=C;"和"V=C+C;"的图灵机示意图

计算机语言的发展

战德臣

哈尔滨工业大学 教授.博士生导师
教育部大学计算机课程教学指导委员会委员



**Research Center on Intelligent
Computing for Enterprises & Services,
Harbin Institute of Technology**

用高级语言编写程序

符号化

语句化

结构化

```
K = 0;
```

```
For I = 1 to 100 Step 1
```

```
{ If I <= 50 && I > 30
```

```
{ K = K + I; }
```

```
}
```

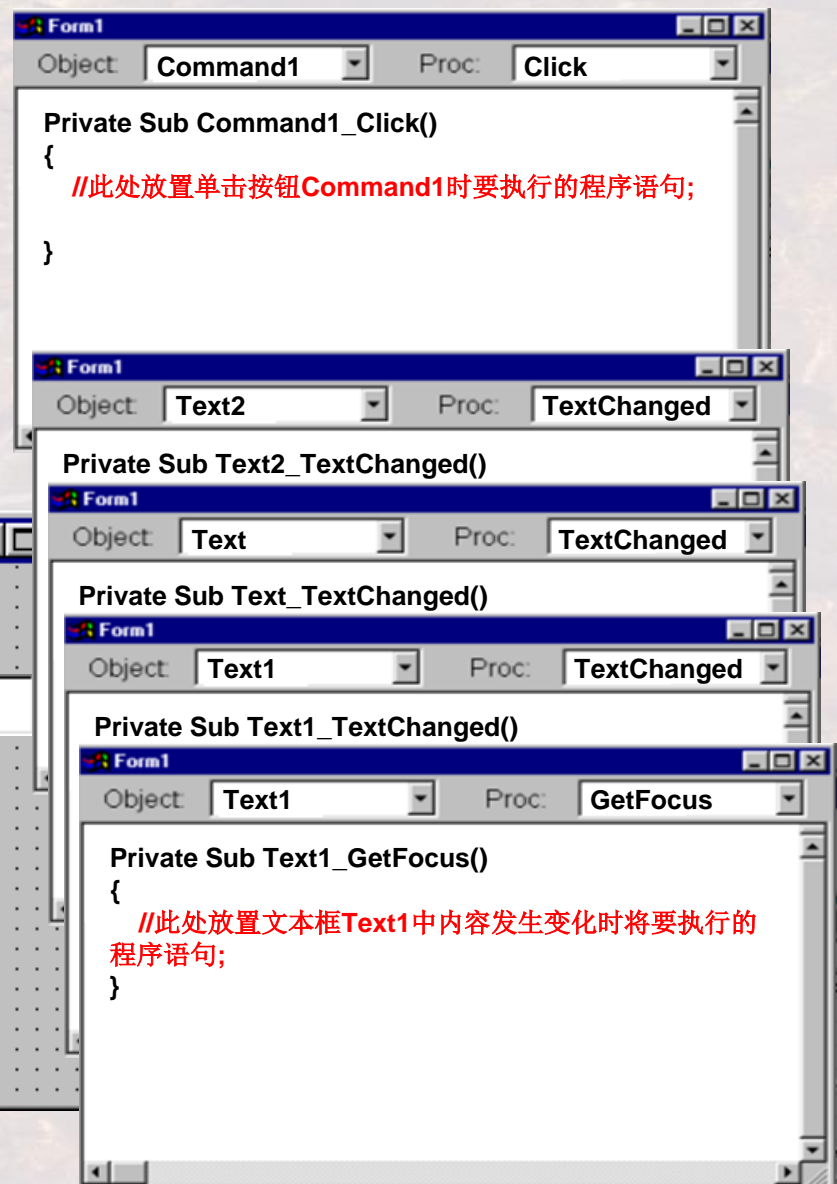
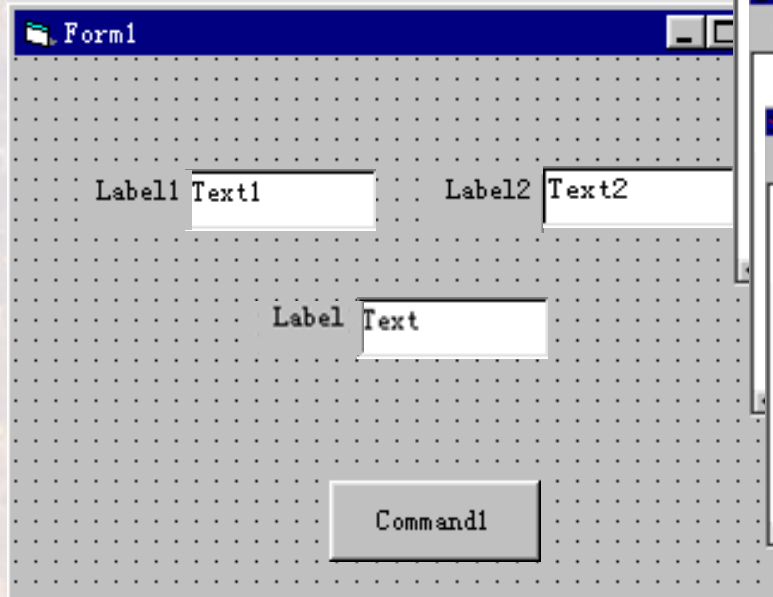
高级语言源程序

编译化

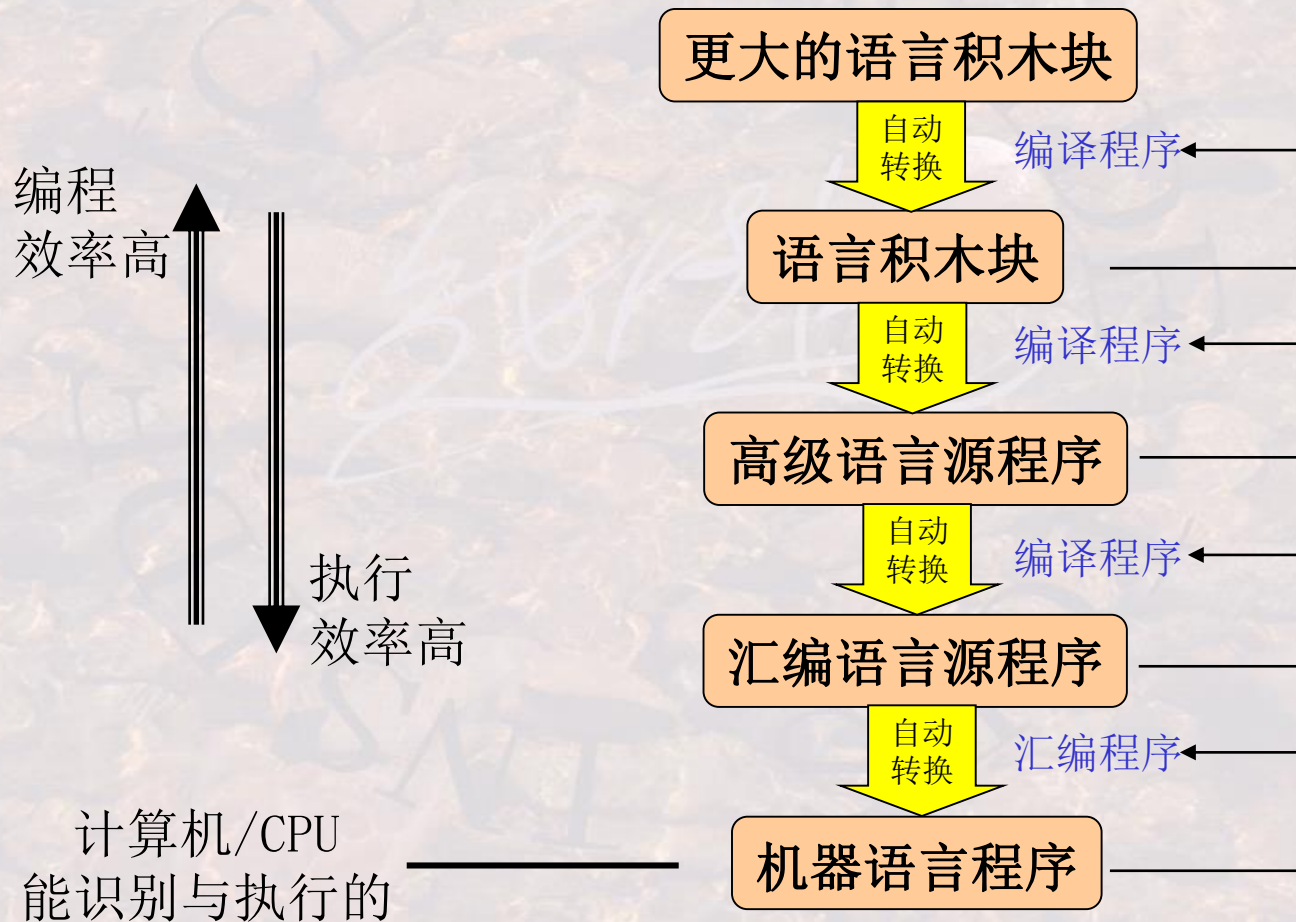
机器语言程序

面向对象的程序设计语言与 可视化构造语言

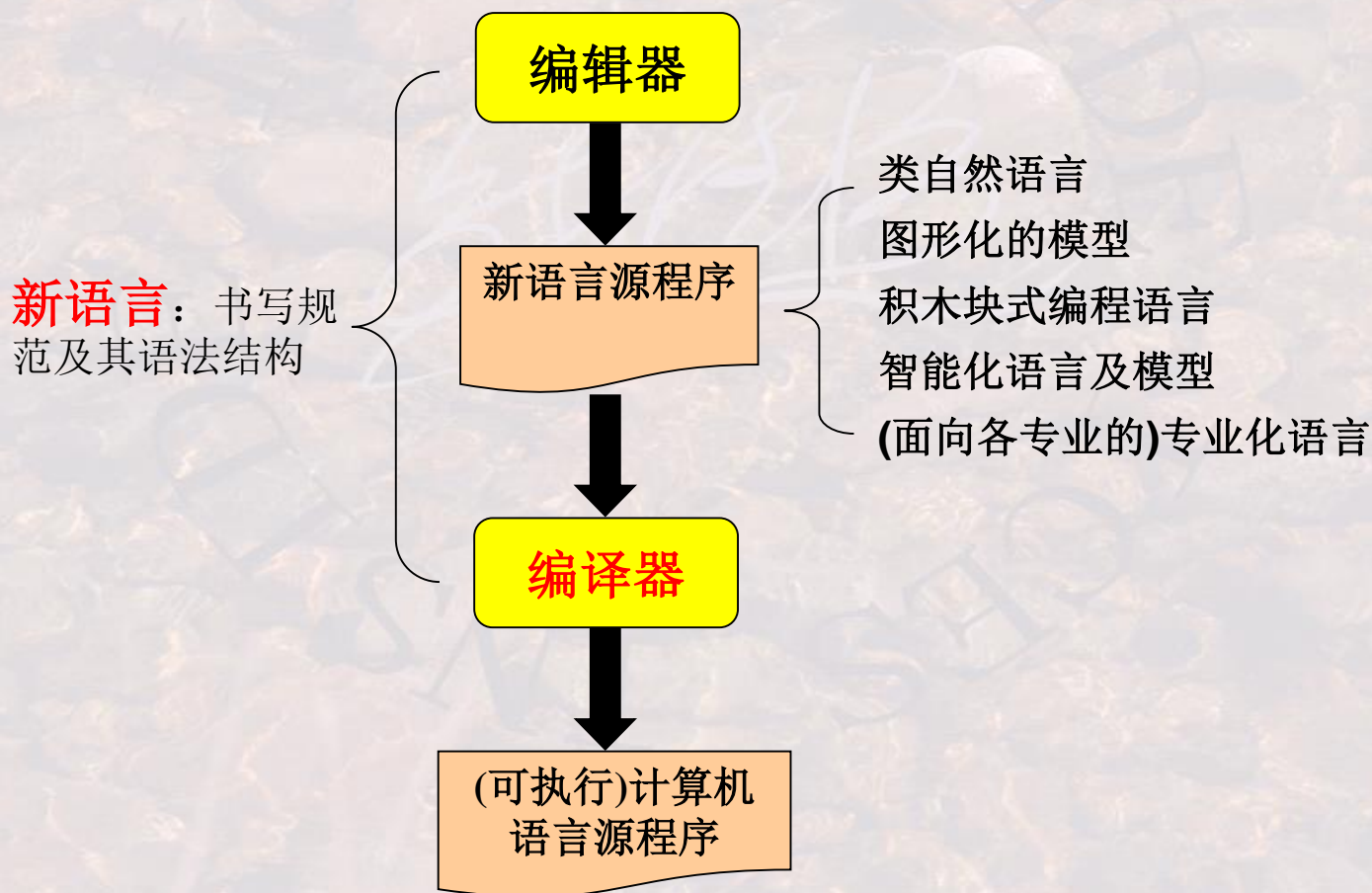
----像堆积木一样构造程序



计算机语言发展的基本思维



不仅要用语言，还要发明新语言



计算机技术是伴随着计算机语言的不断发展而发展起来的

◆因计算机语言获得图灵奖的

- 1966 A.J. Perlis: 编程技术和编译架构

- 1972 E.W. Dijkstra: **ALGOL**语言

- 1974 Donald E. Knuth: 程序语言

- 1977 John Backus : 高级语言, **Fortran**

- 1979 Kenneth E. Iverson: 编程语言, **APL**

- 1980 C. Antony R. Hoare: 编程语言

- 1981 Edgar F. Codd: 关系数据库语言

- 1984 Niklaus Wirth: 开发了**EULER**、**ALGOL-W**、**MODULA**和**PASCAL**一系列崭新的计算语言。

- 1987 John Cocke: 编译器

- 2001 Ole-Johan Dahl、Kristen Nygaard: 面向对象编程,**SIMULA I** 和 **SIMULA 67**中。

- 2003 Alan Kay :面向对象语言, **Smalltalk**

- 2005 Peter Naur:**Algol60**程序语言。

- 2006 Fran Allen: 编译器

计算机语言的发展

(3)能否提出新语言?

Mother Tongues

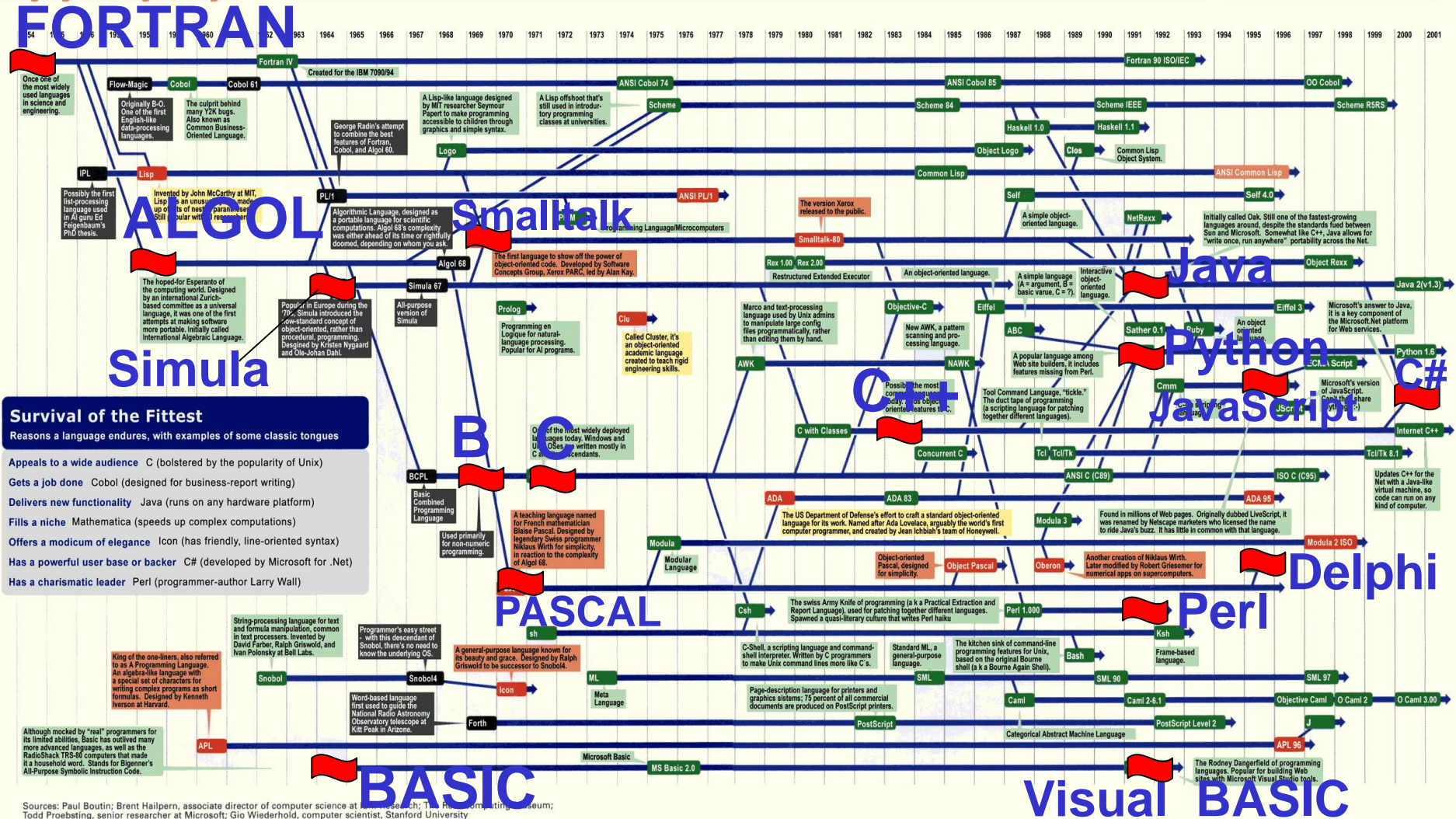
Tracing the roots of computer languages through the ages

Just like half of the world's spoken tongues, most of the 2,300-plus computer programming languages are either endangered or extinct. As powerhouses C/C++, Visual Basic, Cobol, Java and other modern source codes dominate our systems, hundreds of older languages are running out of life.

An ad hoc collection of engineers-electronic lexicographers, if you will-aim to save, or at least document the lingo of classic software. They're combing the globe's 9 million developers in search of coders still fluent in these nearly forgotten lingua frangas. Among the most endangered are Ada, APL, B (the predecessor of C), Lsp, Oberon, Smalltalk, and Simula.

Code-raker Grady Booch, Rational Software's chief scientist, is working with the Computer History Museum in Silicon Valley to record and, in some cases, maintain languages by writing new compilers so our ever-changing hardware can grok the code. Why bother? "They tell us about the state of software practice, the minds of their inventors, and the technical, social, and economic forces that shaped history at the time," Booch explains. "They'll provide the raw material for software archaeologists, historians, and developers to learn what worked, what was brilliant, and what was an utter failure." Here's a peek at the strongest branches of programming's family tree. For a nearly exhaustive rundown, check out the Language List at [HTTP://www.informatik.uni-freiburg.de/Java/misc/lang_list.html](http://www.informatik.uni-freiburg.de/Java/misc/lang_list.html). - Michael Mendeno

Key	
1954	Year introduced
Active: thousands of users	
Protected: taught at universities; compilers available	
Endangered: usage dropping off	
Extinct: no known active users or up-to-date compilers	
Lineage continues	



计算机语言的发展

(3)能否提出新语言?

Mother Tongues

Tracing the roots of computer languages through the ages

FORTRAN

Once one of the most widely used languages in science and engineering.

Flow-Magic

Originally B.O. One of the first English-like data-processing languages.

The culprit behind many Y2K bugs. Also known as Common Business-Oriented Language.

Cobol

Cobol 61

Created for the IBM 7090/94

Fortran IV

George Radin's attempt to combine the best features of Fortran, Cobol, and Algol 68.

Logo

A Lisp-like language designed by MIT researcher Seymour Papert to make programming accessible to children through graphics and simple syntax.

A Lisp offshoot that's still used in introductory programming classes at universities.

Scheme

ANSI Cobol 74

ANSI Cobol 85

Scheme 84

Scheme IEEE

Haskell 1.0

Haskell 1.1

Object Logo

Clos

Common Lisp Object System

ANSI Common Lisp

Self

Self 4.0

NetRexx

Object Rexx

Java 2(v1.3)

Java

Python 1.6

Python

C#

JavaScript

Internet C++

Delphi

Perl

Visual BASIC

BASIC

PASCAL

C++

C

B

Simula

Algol

Smalltalk

PL/I

PL/1

Algol 68

Prolog

Cle

AWK

NAWK

Objective-C

Eiffel

ABC

Tool Command Language "tcl"

Perl 1.000

Perl

Perl 5.000

Perl 5.001

Perl 5.002

Perl 5.003

Perl 5.004

Perl 5.005

Perl 5.006

Perl 5.007

Perl 5.008

Perl 5.009

Perl 5.010

Perl 5.011

Perl 5.012

Perl 5.013

Perl 5.014

Perl 5.015

Perl 5.016

Perl 5.017

Perl 5.018

Perl 5.019

Perl 5.020

Perl 5.021

Perl 5.022

Perl 5.023

Perl 5.024

Perl 5.025

Perl 5.026

Perl 5.027

Perl 5.028

Perl 5.029

Perl 5.030

Perl 5.031

Perl 5.032

Perl 5.033

Perl 5.034

Perl 5.035

Perl 5.036

Perl 5.037

Perl 5.038

Perl 5.039

Perl 5.040

Perl 5.041

Perl 5.042

Perl 5.043

Perl 5.044

Perl 5.045

Perl 5.046

Perl 5.047

Perl 5.048

Perl 5.049

Perl 5.050

Perl 5.051

Perl 5.052

Perl 5.053

Perl 5.054

Perl 5.055

Perl 5.056

Perl 5.057

Perl 5.058

Perl 5.059

Perl 5.060

Perl 5.061

Perl 5.062

Perl 5.063

Perl 5.064

Perl 5.065

Perl 5.066

Perl 5.067

Perl 5.068

Perl 5.069

Perl 5.070

Perl 5.071

Perl 5.072

Perl 5.073

Perl 5.074

Perl 5.075

Perl 5.076

Perl 5.077

Perl 5.078

Perl 5.079

Perl 5.080

Perl 5.081

Perl 5.082

Perl 5.083

Perl 5.084

Perl 5.085

Perl 5.086

Perl 5.087

Perl 5.088

Perl 5.089

Perl 5.090

Perl 5.091

Perl 5.092

Perl 5.093

Perl 5.094

Perl 5.095

Perl 5.096

Perl 5.097

Perl 5.098

Perl 5.099

Perl 5.100

Perl 5.101

Perl 5.102

Perl 5.103

Perl 5.104

Perl 5.105

Perl 5.106

Perl 5.107

Perl 5.108

Perl 5.109

Perl 5.110

Perl 5.111

Perl 5.112

Perl 5.113

Perl 5.114

Perl 5.115

Perl 5.116

Perl 5.117

Perl 5.118

Perl 5.119

Perl 5.120

Perl 5.121

Perl 5.122

Perl 5.123

Perl 5.124

Perl 5.125

Perl 5.126

Perl 5.127

Perl 5.128

Perl 5.129

Perl 5.130

Perl 5.131

Perl 5.132

Perl 5.133

Perl 5.134

Perl 5.135

Perl 5.136

Perl 5.137

Perl 5.138

Perl 5.139

Perl 5.140

Perl 5.141

Perl 5.142

Perl 5.143

Perl 5.144

Perl 5.145

Perl 5.146

Perl 5.147

Perl 5.148

Perl 5.149

Perl 5.150

Perl 5.151

Perl 5.152

Perl 5.153

Perl 5.154

Perl 5.155

Perl 5.156

Perl 5.157

Perl 5.158

Perl 5.159

Perl 5.160

Perl 5.161

Perl 5.162

Perl 5.163

Perl 5.164

Perl 5.165

Perl 5.166

Perl 5.167

Perl 5.168

Perl 5.169

Perl 5.170

Perl 5.171

Perl 5.172

Perl 5.173

Perl 5.174

Perl 5.175

Perl 5.176

Perl 5.177

Perl 5.178

Perl 5.179

Perl 5.180

Perl 5.181

Perl 5.182

Perl 5.183

Perl 5.184

Perl 5.185

Perl 5.186

Perl 5.187

Perl 5.188

Perl 5.189

Perl 5.190

Perl 5.191

Perl 5.192

Perl 5.193

Perl 5.194

Perl 5.195

Perl 5.196

Perl 5.197

Perl 5.198

Perl 5.199

Perl 5.200

Perl 5.201

Perl 5.202

Perl 5.203

Perl 5.204

Perl 5.205

Perl 5.206

Perl 5.207

Perl 5.208

Perl 5.209

Perl 5.210

Perl 5.211

Perl 5.212

Perl 5.213

Perl 5.214

Perl 5.215

Perl 5.216

Perl 5.217

Perl 5.218

Perl 5.219

Perl 5.220

Perl 5.221

Perl 5.222

Perl 5.223

Perl 5.224

Perl 5.225

Perl 5.226

Perl 5.227

Perl 5.228

Perl 5.229

Perl 5.230

Perl 5.231

Perl 5.232

Perl 5.233

Perl 5.234

Perl 5.235

Perl 5.236

Perl 5.237

Perl 5.238

Perl 5.239

Perl 5.240

Perl 5.241

Perl 5.242

Perl 5.243

Perl 5.244

Perl 5.245

Perl 5.246

Perl 5.247

Perl 5.248

Perl 5.249

Perl 5.250

Perl 5.251

Perl 5.252

Perl 5.253

Perl 5.254

Perl 5.255

Perl 5.256

Perl 5.257

Perl 5.258

Perl 5.259

Perl 5.260

Perl 5.261

Perl 5.262

Perl 5.263

Perl 5.264

Perl 5.265

Perl 5.266

Perl 5.267

Perl 5.268

Perl 5.269

Perl 5.270

Perl 5.271

Perl 5.272

Perl 5.273

Perl 5.274

Perl 5.275

Perl 5.276

Perl 5.277

Perl 5.278

Perl 5.279

Perl 5.280

Perl 5.281

Perl 5.282

Perl 5.283

Perl 5.284

Perl 5.285

Perl 5.286

Perl 5.287

Perl 5.288

Perl 5.289

Perl 5.290

Perl 5.291

Perl 5.292

Perl 5.293

Perl 5.294

Perl 5.295

Perl 5.296

Perl 5.297

Perl 5.298

Perl 5.299

Perl 5.300

Perl 5.301

Perl 5.302

Perl 5.303

Perl 5.304

Perl 5.305

Perl 5.306

Perl 5.307

Perl 5.308

Perl 5.309

Perl 5.310

Perl 5.311

Perl 5.312

Perl 5.313

Perl 5.314

Perl 5.315

Perl 5.316

Perl 5.317

Perl 5.318

Perl 5.319

Perl 5.320

Perl 5.321

Perl 5.322

Perl 5.323

Perl 5.324

Perl 5.325

Perl 5.326

Perl 5.327

Perl 5.328

Perl 5.329

Perl 5.330

Perl 5.331

Perl 5.332

Perl 5.333

Perl 5.334

Perl 5.335

Perl 5.336

Perl 5.337

Perl 5.338

Perl 5.339

Perl 5.340

Perl 5.341

Perl 5.342

Perl 5.343

Perl 5.344

Perl 5.345

Perl 5.346

Perl 5.347

Perl 5.348

Perl 5.349

Perl 5.350

Perl 5.351

Perl 5.352

Perl 5.353

Perl 5.354

Perl 5.355

Perl 5.356

Perl 5.357

Perl 5.358

Perl 5.359

Perl 5.360

Perl 5.361

Perl 5.362

Perl 5.363

Perl 5.364

Perl 5.365

Perl 5.366

Perl 5.367

Perl 5.368

Perl 5.369

Perl 5.370

Perl 5.371

Perl 5.372

Perl 5.373

Perl 5.374

Perl 5.375

Perl 5.376

Perl 5.377

Perl 5.378

Perl 5.379

Perl 5.380

Perl 5.381

Perl 5.382

Perl 5.383

Perl 5.384

Perl 5.385

Perl 5.386

Perl 5.387

Perl 5.388

Perl 5.389

Perl 5.390

Perl 5.391

Perl 5.392

Perl 5.393

Perl 5.394

Perl 5.395

Perl 5.396

Perl 5.397

Perl 5.398

Perl 5.399

Perl 5.400

Perl 5.401

Perl 5.402

Perl 5.403

Perl 5.404

Perl 5.405

Perl 5.406

Perl 5.407

Perl 5.408

Perl 5.409

Perl 5.410

Perl 5.411

Perl 5.412

Perl 5.413

Perl 5.414

Perl 5.415

Perl 5.416

Perl 5.417

Perl 5.418

Perl 5.419

Perl 5.420

Perl 5.421

Perl 5.422

Perl 5.423

Perl 5.424

Perl 5.425

Perl 5.426

Perl 5.427

Perl 5.428

Perl 5.429

Perl 5.430

Perl 5.431

Perl 5.432

Perl 5.433

Perl 5.434

Perl 5.435

Perl 5.436

Perl 5.437

Perl 5.438

Perl 5.439

Perl 5.440

Perl 5.441

Perl 5.442

Perl 5.443

Perl 5.444

Perl 5.445

Perl 5.446

Perl 5.447

Perl 5.448

Perl 5.449

Perl 5.450

Perl 5.451

Perl 5.452

Perl 5.453

Perl 5.454

Perl 5.455

Perl 5.456

Perl 5.457

Perl 5.458

Perl 5.459

Perl 5.460

Perl 5.461

Perl 5.462

Perl 5.463

Perl 5.464

Perl 5.465

Perl 5.466

Perl 5.467

Perl 5.468

Perl 5.469

Perl 5.470

Perl 5.471

Perl 5.472

Perl 5.473

Perl 5.474

Perl 5.475

Perl 5.476

Perl 5.477

Perl 5.478

Perl 5.479

Perl 5.480

Perl 5.481

Perl 5.482

Perl 5.483

Perl 5.484

Perl 5.485

Perl 5.486

Perl 5.487

Perl 5.488

Perl 5.489

Perl 5.490

Perl 5.491

Perl 5.492

Perl 5.493

Perl 5.494

Perl 5.495

Perl 5.496

Perl 5.497

Perl 5.498

Perl 5.499

Perl 5.500

Perl 5.501

Perl 5.502

Perl 5.503

Perl 5.504

Perl 5.505

Perl 5.506

Perl 5.507

Perl 5.508

Perl 5.509

Perl 5.510

Perl 5.511

Perl 5.512

Perl 5.513

Perl 5.514

Perl 5.515

Perl 5.516

Perl 5.517

Perl 5.518

Perl 5.519

Perl 5.520

Perl 5.521

Perl 5.522

Perl 5.523

Perl 5.524

Perl 5.525

Perl 5.526

Perl 5.527

Perl 5.528

Perl 5.529

Perl 5.530

Perl 5.531

Perl 5.532

Perl 5.533

Perl 5.534

Perl 5.535

Perl 5.536

Perl 5.537

Perl 5.538

Perl 5.539

Perl 5.540

Perl 5.541

Perl 5.542

Perl 5.543

Perl 5.544

Perl 5.545

Perl 5.546

Perl 5.547

Perl 5.548

Perl 5.549

Perl 5.550

Perl 5.551

Perl 5.552

Perl 5.553

Perl 5.554

Perl 5.555

Perl 5.556

Perl 5.557

Perl 5.558

Perl 5.559

Perl 5.560

Perl 5.561

Perl 5.562

Perl 5.563

Perl 5.564

Perl 5.565

Perl 5.566

Perl 5.567

Perl 5.568

Perl 5.569

Perl 5.570

Perl 5.571

Perl 5.572

Perl 5.573

Perl 5.574

Perl 5.575

Perl 5.576

Perl 5.577

Perl 5.578

Perl 5.579

Perl 5.580

Perl 5.581

Perl 5.582

Perl 5.583

Perl 5.584

Perl 5.585

Perl 5.586

Perl 5.587

Perl 5.588

Perl 5.589

Perl 5.590

Perl 5.591

Perl 5.592

Perl 5.593

Perl 5.594

Perl 5.595

Perl 5.596

Perl 5.597

Perl 5.598

Perl 5.599

Perl 5.600

Perl 5.601

Perl 5.602

Perl 5.603

Perl 5.604

Perl 5.605

Perl 5.606

Perl 5.607

Perl 5.608

Perl 5.609

Perl 5.610

Perl 5.611

Perl 5.612

Perl 5.613

Perl 5.614

Perl 5.615

Perl 5.616

Perl 5.617

Perl 5.618

Perl 5.619

Perl 5.620

Perl 5.621

Perl 5.622

Perl 5.623

Perl 5.624

Perl 5.625

Perl 5.626

Perl 5.627

Perl 5.628

Perl 5.629

Perl 5.630

Perl 5.631

Perl 5.632

Perl 5.633

Perl 5.634

Perl 5.635

Perl 5.636

Perl 5.637

Perl 5.638

Perl 5.639

Perl 5.640

Perl 5.641

Perl 5.642

Perl 5.643

Perl 5.644

Perl 5.645

Perl 5.646

Perl 5.647

Perl 5.648

Perl 5.649

Perl 5.650

Perl 5.651

Perl 5.652

Perl 5.653

Perl 5.654

Perl 5.655

Perl 5.656

Perl 5.657

Perl 5.658

Perl 5.659

Perl 5.660

Perl 5.661

Perl 5.662

Perl 5.663

Perl 5.664

Perl 5.665

Perl 5.666

Perl 5.667

Perl 5.668

Perl 5.669

Perl 5.670

Perl 5.671

Perl 5.672

Perl 5.673

Perl 5.674

Perl 5.675

Perl 5.676

Perl 5.677

Perl 5.678

Perl 5.679

Perl 5.680

Perl 5.681

Perl 5.682

Perl 5.683

Perl 5.684

Perl 5.685

Perl 5.686

Perl 5.687

Perl 5.688

Perl 5.689

Perl 5.690

Perl 5.691

Perl 5.692

Perl 5.693

Perl 5.694

Perl 5.695

Perl 5.696

Perl 5.697

Perl 5.698

Perl 5.699

Perl 5.700

Perl 5.701

Perl 5.702

Perl 5.703

Perl 5.704

Perl 5.705

Perl 5.706

Perl 5.707

Perl 5.708

Perl 5.709

Perl 5.710

Perl 5.711

Perl 5.712

Perl 5.713

Perl 5.714

Perl 5.715

Perl 5.716

Perl 5.717

Perl 5.718

Perl 5.719

Perl 5.720

Perl 5.721

Perl 5.722

Perl 5.723

Perl 5.724

Perl 5.725

Perl 5.726

Perl 5.727

Perl 5.728

Perl 5.729

Perl 5.730

Perl 5.731

Perl 5.732

Perl 5.733

Perl 5.734

Perl 5.735

Perl 5.736

Perl 5.737

Perl 5.738

Perl 5.739

Perl 5.740

Perl 5.741

Perl 5.742

Perl 5.743

Perl 5.744

Perl 5.745

Perl 5.746

Perl 5.747

Perl 5.748

Perl 5.749

Perl 5.750

Perl 5.751

Perl 5.752

Perl 5.753

Perl 5.754

Perl 5.755

Perl 5.756

Perl 5.757

Perl 5.758

Perl 5.759

Perl 5.760

Perl 5.761

Perl 5.762

Perl 5.763

Perl 5.764

Perl 5.765

Perl 5.766

Perl 5.767

Perl 5.768

Perl 5.769

Perl 5.770

Perl 5.771

Perl 5.772

Perl 5.773

Perl 5.774

Perl 5.775

Perl 5.776

Perl 5.777

Perl 5.778

Perl 5.779

Perl 5.780

Perl 5.781

Perl 5.782

Perl 5.783

Perl 5.784

Perl 5.785

Perl 5.786

Perl 5.787

Perl 5.788

Perl 5.789

Perl 5.790

Perl 5.791

Perl 5.792

Perl 5.793

Perl 5.794

Perl 5.795

Perl 5.796

Perl 5.797

Perl 5.798

Perl 5.799

Perl 5.800

Perl 5.801

Perl 5.802

Perl 5.803

Perl 5.804

Perl 5.805

Perl 5.806

Perl 5.807

Perl 5.808

Perl 5.809

Perl 5.810

Perl 5.811

Perl 5.812

Perl 5.813

Perl 5.814

Perl 5.815

Perl 5.816

Perl 5.817

Perl 5.818

Perl 5.819

Perl 5.820

Perl 5.821

Perl 5.822

Perl 5.823

Perl 5.824

Perl 5.825

Perl 5.826

Perl 5.827

Perl 5.828

Perl 5.829

Perl 5.830

Perl 5.831

Perl 5.832

Perl 5.833

Perl 5.834

Perl 5.835

Perl 5.836

Perl 5.837

Perl 5.838

Perl 5.839

Perl 5.840

Perl 5.841

Perl 5.842

Perl 5.843

Perl 5.844

Perl 5.845

Perl 5.846

Perl 5.847

Perl 5.848

Perl 5.849

Perl 5.850

Perl 5.851

Perl 5.852

Perl 5.853

Perl 5.854

Perl 5.855

Perl 5.856

Perl 5.857

Perl 5.858

Perl 5.859

Perl 5.860

Perl 5.861

Perl 5.862

Perl 5.863

Perl 5.864

Perl 5.865

Perl 5.866

Perl 5.867

Perl 5.868

Perl 5.869

Perl 5.870

Perl 5.871

Perl 5.872

Perl 5.873

Perl 5.874

Perl 5.875

Perl 5.876

Perl 5.877

Perl 5.878

Perl 5.879

Perl 5.880

Perl 5.881

Perl 5.882

Perl 5.883

Perl 5.884

Perl 5.885

Perl 5.886

Perl 5.887

Perl 5.888

Perl 5.889

Perl 5.890

Perl 5.891

Perl 5.892

Perl 5.893

Perl 5.894

Perl 5.895

Perl 5.896

Perl 5.897

Perl 5.898

Perl 5.899

Perl 5.900

Perl 5.901

Perl 5.902

Perl 5.903

Perl 5.904

Perl 5.905

Perl 5.906

Perl 5.907

Perl 5.908

Perl 5.909

Perl 5.910

Perl 5.911

Perl 5.912

Perl 5.913

Perl 5.914

Perl 5.915

Perl 5.916

Perl 5.917

Perl 5.918

Perl 5.919

Perl 5.920

Perl 5.921

Perl 5.922

Perl 5.923

Perl 5.924

Perl 5.925

Perl 5.926

Perl 5.927

Perl 5.928

Perl 5.929

Perl 5.930

Perl 5.931

Perl 5.932

Perl 5.933

Perl 5.934

Perl 5.935

Perl 5.936

Perl 5.937

Perl 5.938

Perl 5.939

Perl 5.940

Perl 5.941

Perl 5.942

Perl 5.943

Perl 5.944

Perl 5.945

Perl 5.946

Perl 5.947

Perl 5.948

Perl 5.949

Perl 5.950

Perl 5.951

Perl 5.952

Perl 5.953

Perl 5.954

Perl 5.955

Perl 5.956

Perl 5.957

Perl 5.958

Perl 5.959

Perl 5.960

Perl 5.961

Perl 5.962

Perl 5.963

Perl 5.964

Perl 5.965

Perl 5.966

Perl

不同抽象层级计算机(虚拟机器)

战德臣

哈尔滨工业大学 教授.博士生导师
教育部大学计算机课程教学指导委员会委员



**Research Center on Intelligent
Computing for Enterprises & Services,
Harbin Institute of Technology**

计算机语言促进了计算机处理能力的不断增强



不同抽象层级计算机(虚拟机器)

(2)你想编程序但又不涉及硬件吗？

计算机语言促进了计算机处理能力的不断增强



计算机语言促进了计算机处理能力的不断增强



计算机语言促进了计算机处理能力的不断增强



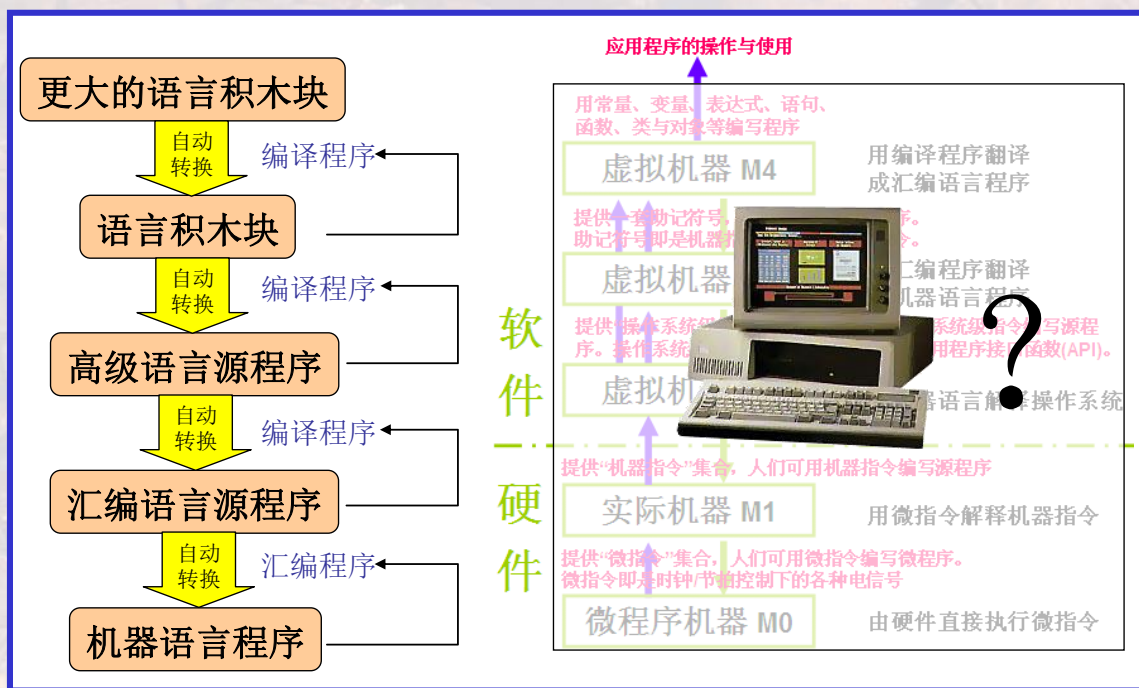
计算机语言促进了计算机处理能力的不断增强



计算机语言促进了计算机处理能力的不断增强



基本目标：理解如何编写计算机可以执行的程序



基本思维：高级语言与汇编语言 → 语言与编译器 → 高级语言程序的构成要素 → 不同层面的计算机

协议与编解码器

--另一种抽象-自动化机制示例

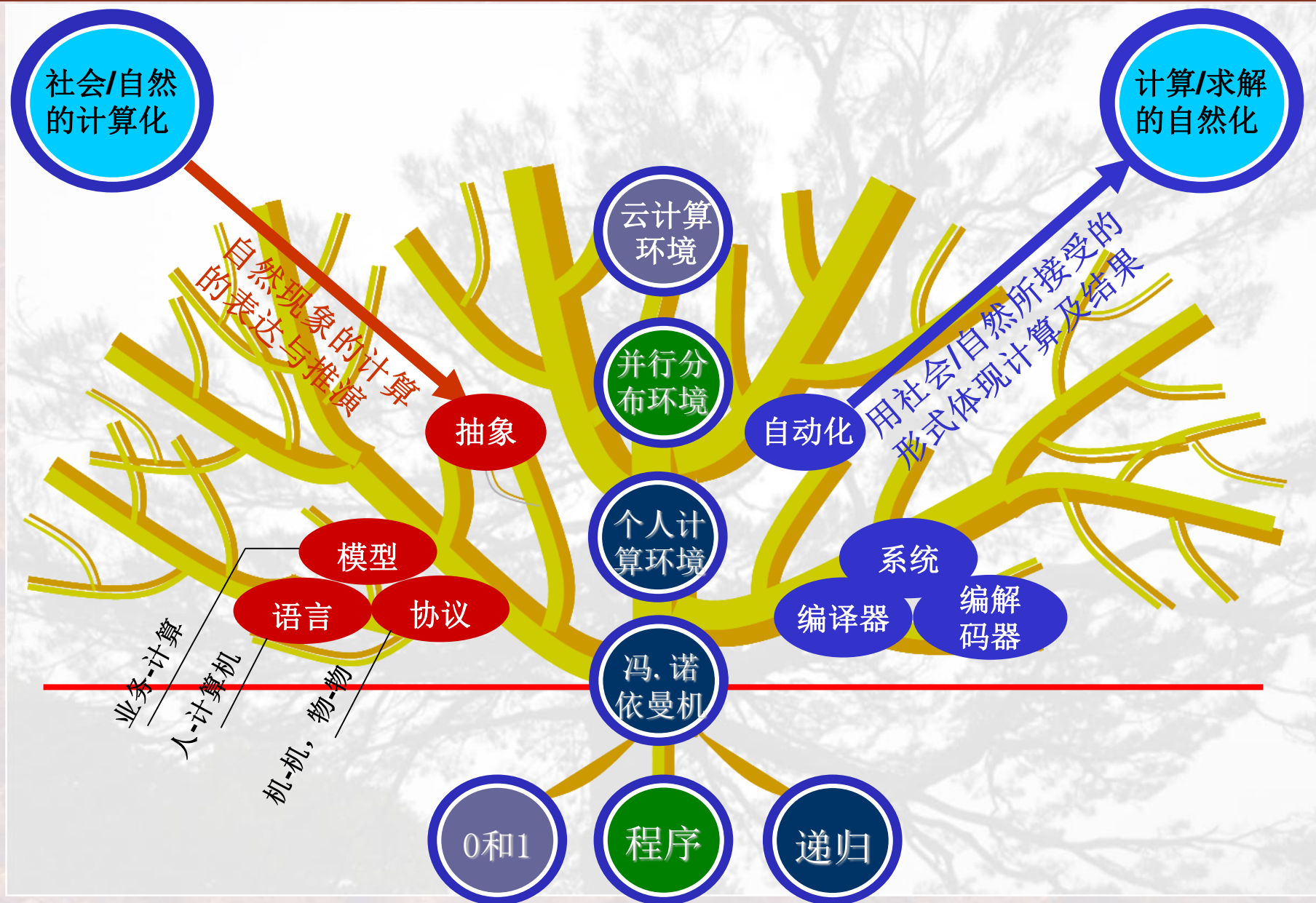
战德臣

哈尔滨工业大学 教授.博士生导师
教育部大学计算机课程教学指导委员会委员



Research Center on Intelligent
Computing for Enterprises & Services,
Harbin Institute of Technology

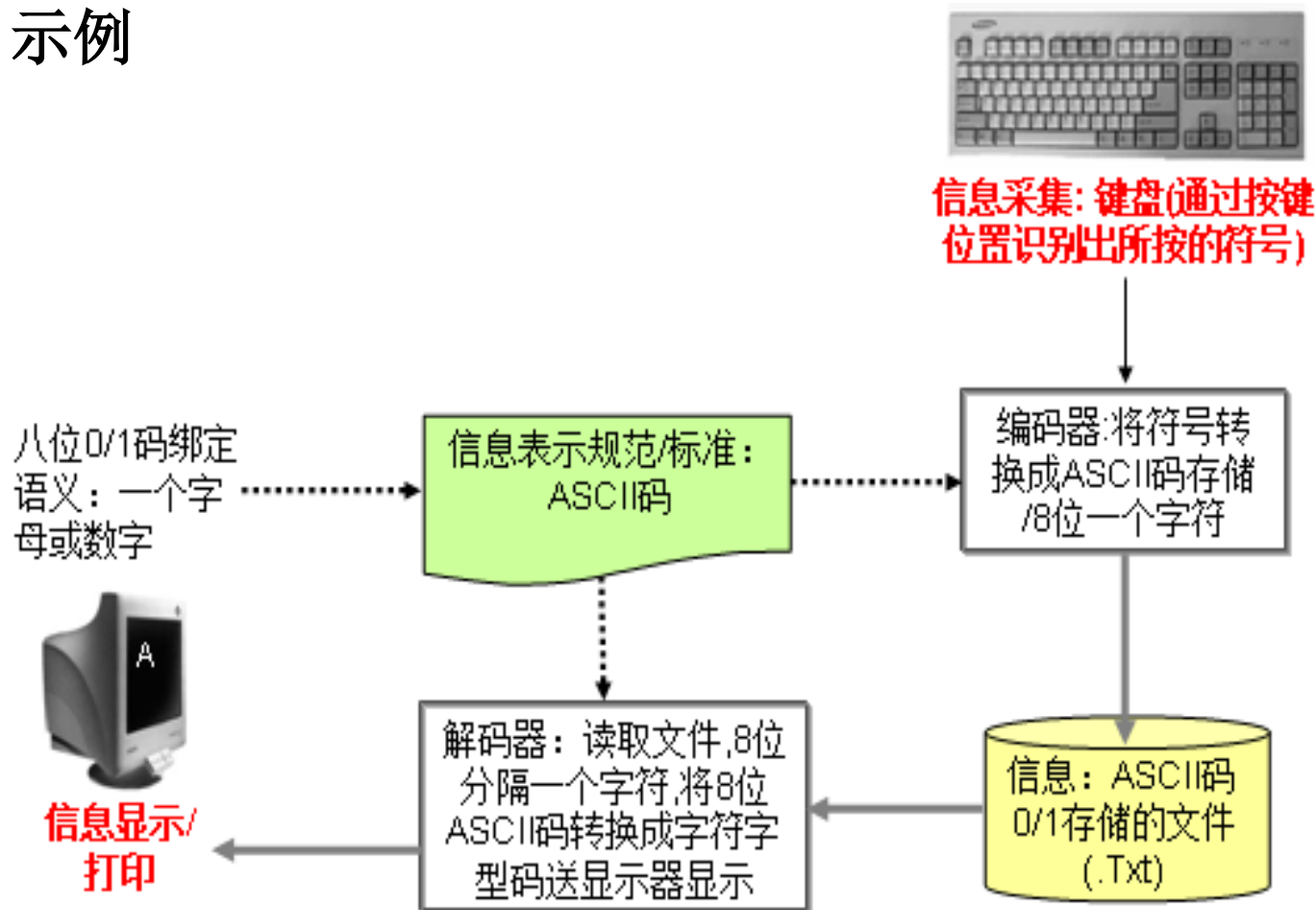
(1)你记得计算之树中的不同抽象层次吗？



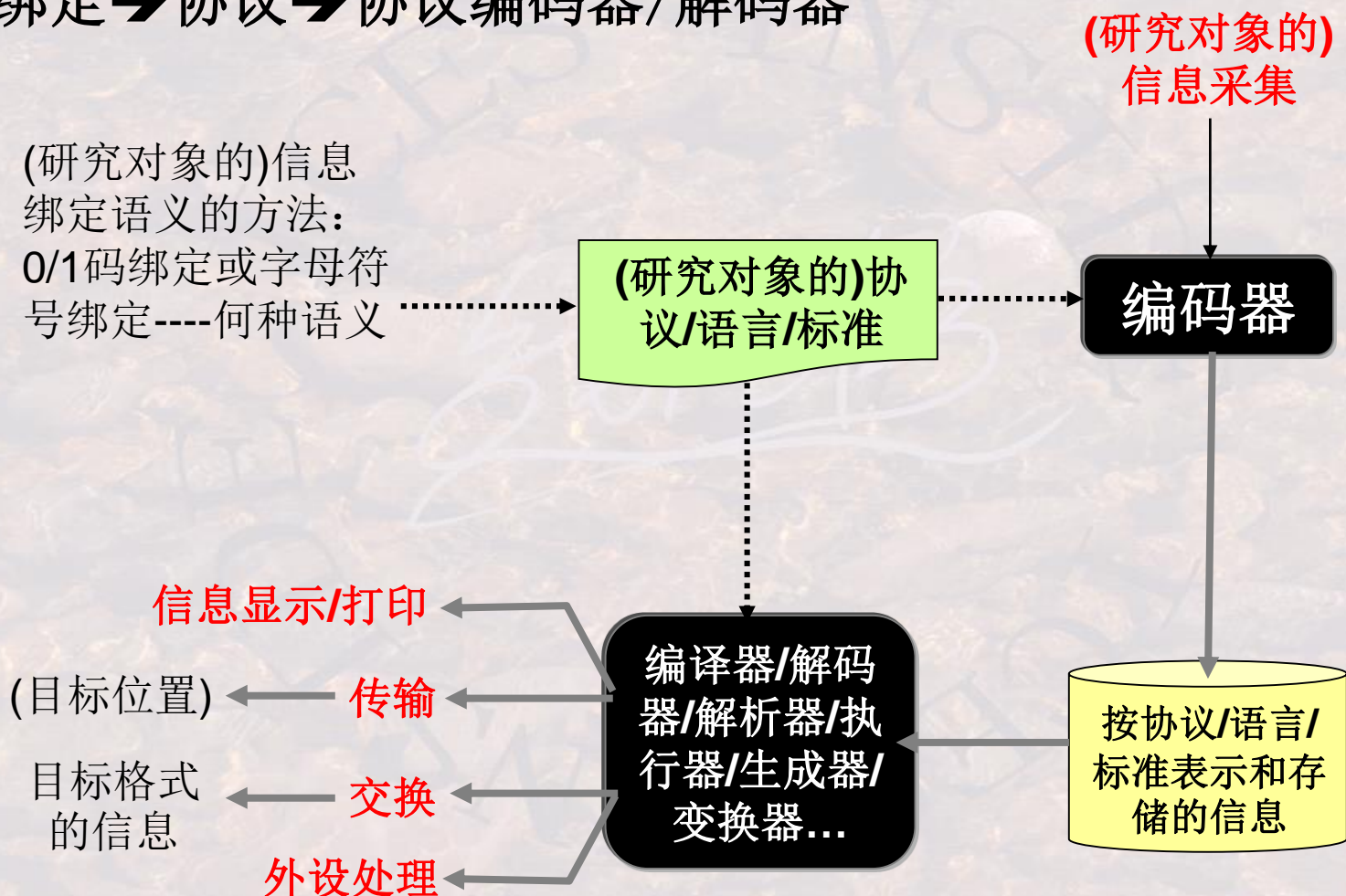
协议及其应用示意：键盘输入/与显示器显示处理

一般而言，
“协议”是为
交流信息的双
方(计算机)能
够正确实现信
息交流而建立
的一套规则、
标准或约定。

示例



信息绑定→协议→协议编码器/解码器



协议与编解码器体现了信息处理的一般性思

协议与编解码器/解码器示例----多媒体处理

音频信息的表示与处理

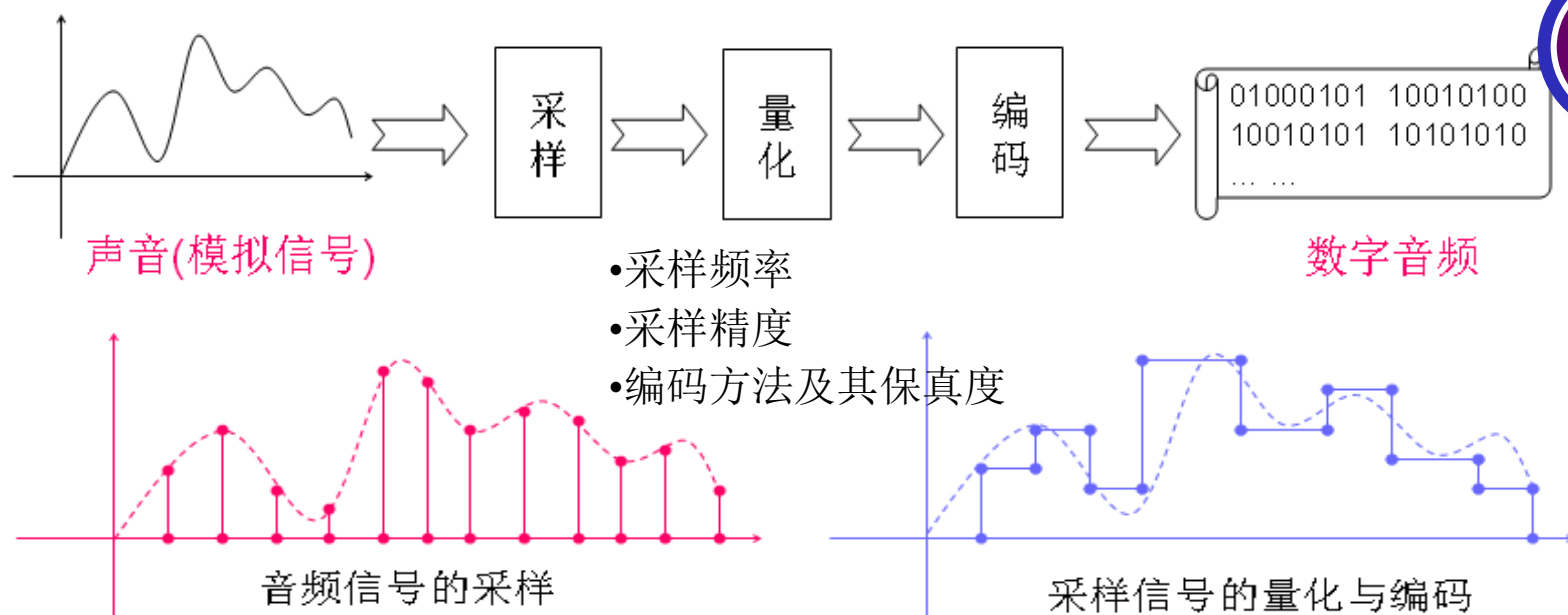
◆**音频编码**：音频是连续的模拟信号，需经采样、量化和编码后形成数字音频后，进行数字处理。所谓采样是指按一定的采样频率对连续音频信号做时间上的离散化，即对连续信号隔一定周期获取一个信号点的过程。而量化是将所采集的信号点的数值区分成不同位数的离散数值的过程。而编码则是将采集到的离散时间点的信号的离散数值按一定规则编码存储的过程。

采样

量化

编码

解码



协议与编解码器/解码器示例----多媒体处理

图像表示与处理

◆**位图图像**：将图像划分成均匀的由单元点构成的网格，每个单元点称为像素。每个像素可由1位或多位表示，1位只能表示黑白图像，8位能表示灰度图像，24位则能表示彩色图像。单位尺寸内的像素数目被称为图像的分辨率，由水平像素数目×垂直像素数目来表示。



像素表达颜色的不同，
需要编码的位数不同。

黑白-1位(0,1)

256级灰度-8位(0-255)

16色彩色-4位(0-16)

256色彩色-8位(0-255)

24位真彩色-24位

(红0-255、绿0-255、蓝0-255三元色)

采样

量化

编码

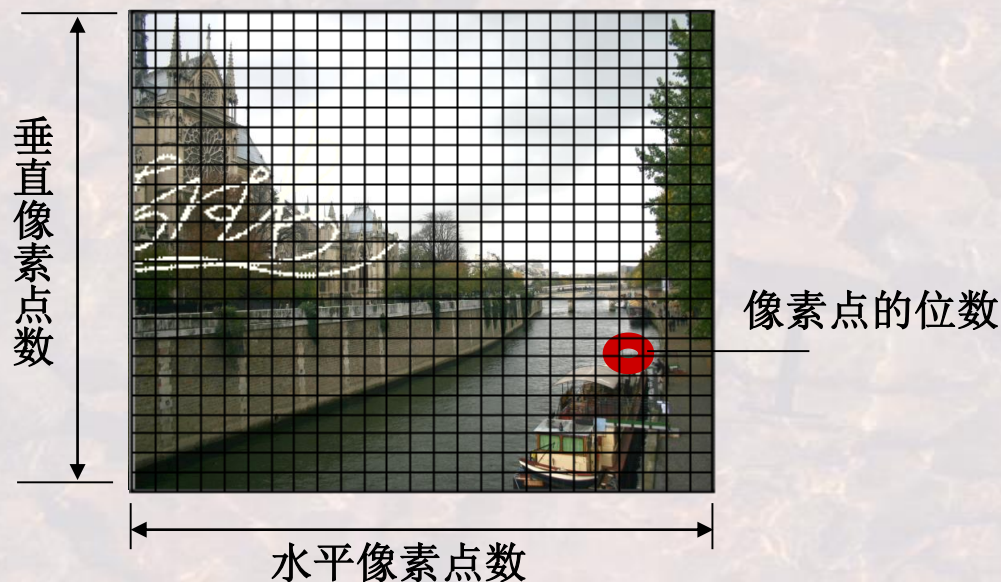
解码

协议与编解码器/解码器示例----多媒体处理

◆ **图像编码：** 由于位图图像的存储量大(水平像素数目×垂直像素数目×每像素位数)，通常都需要进行压缩存储，不同的压缩采用了不同的图像编码。典型的有：

◆ **JPEG：** 国际标准化组织(ISO)和国际电报电话咨询委员会(CCITT)联合成立的“联合照片专家组”于1991年3月提出了JPEG标准(Joint Photographic Experts Group)。

◆ 其他常用编码格式有：BMP，GIF，TIFF，...



采样

量化

编码

解码

你知道图像的编码器、解码器是什么吗？你有使用过吗？

协议与编解码器/解码器示例----多媒体处理

视频信号的表示与处理

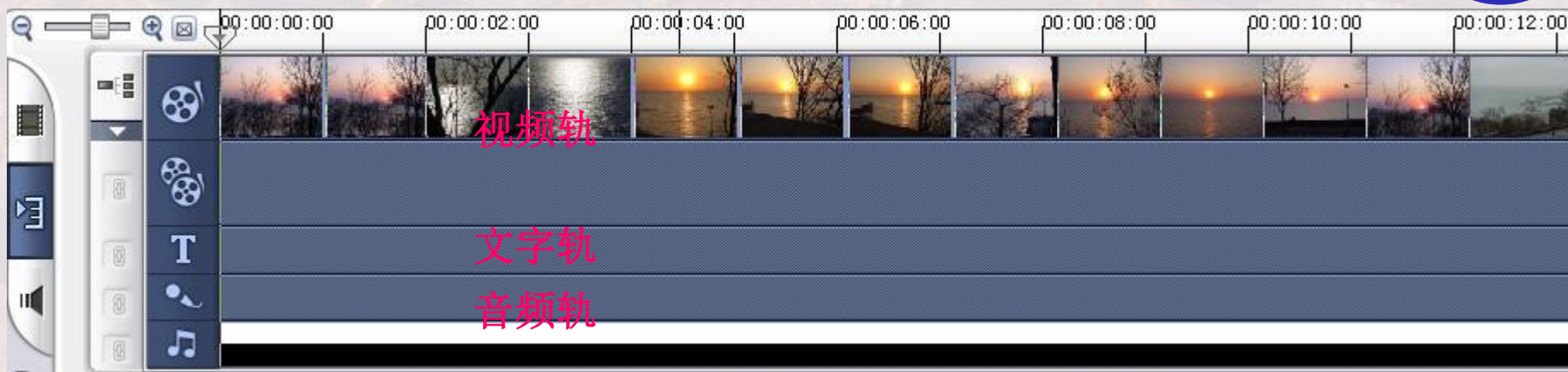
◆视频：是时间序列的动态图像(如25帧/秒)，也是连续的模拟信号，需要经过采样、量化和编码形成数字视频，保存和处理。同时，视频还可能由视频、音频及文字经同步后形成的。因此视频处理相当于按照时间序列处理图像、声音和文字及其同步问题。

采样

量化

编码

解码

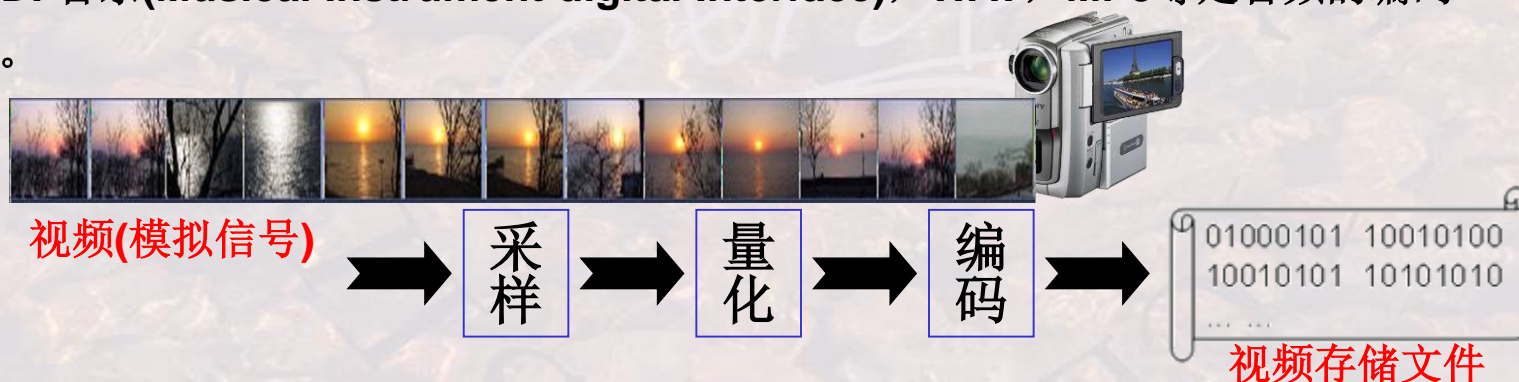


协议与编解码器/解码器示例----多媒体处理

视频信号的表示与处理

◆视频编码： MPEG是Moving Pictures Experts Group(动态图象专家组)的缩写。提出了四个版本： MPEG-I(VCD: Video CD) 、 MPEG-II (DVD:Digital Versatile Disk) 、 MPEG-III、 MPEG-IV(多媒体)。

◆MIDI 音乐(Musical Instrument digital Interface), WAV, MP3等是音频的编码标准。

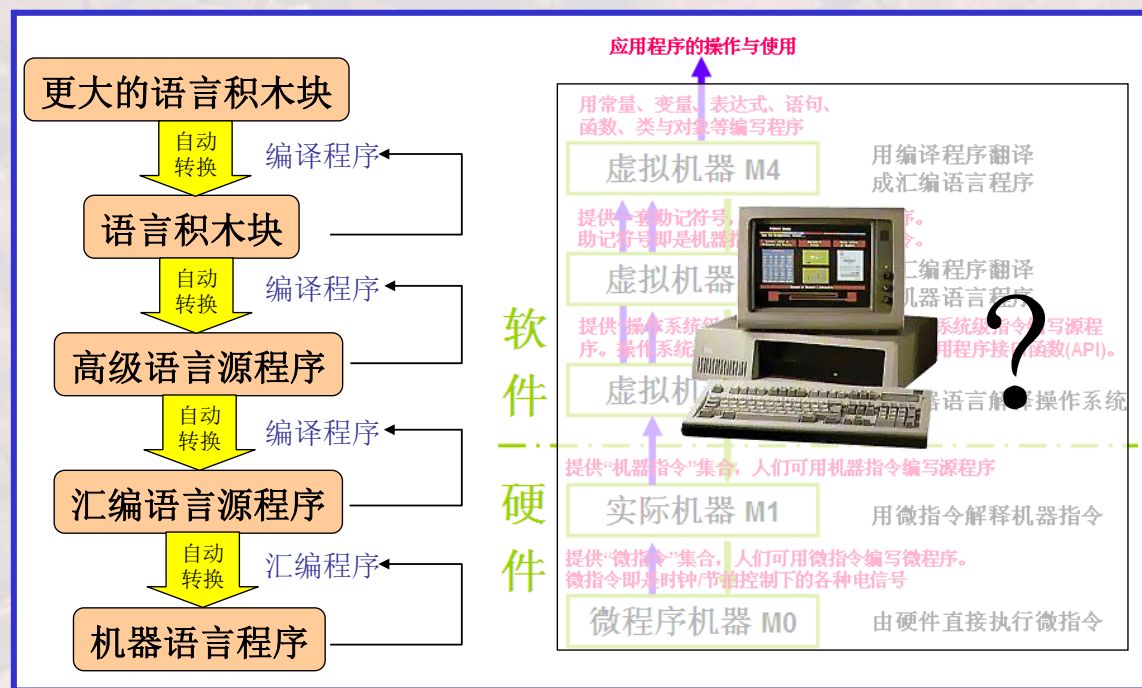


- 数据速率
- 压缩比
- 关键帧



基本目标： 理解如何编写计算机可以执行的程序

执行零壹编程符号，
汇编语言高级语言。
不同层次虚拟机器，
分层变换编译真谛。
各级协议编解码器，
道理相通扩展容易。



基本思维： 高级语言与汇编语言 → 语言与编译器 → 协议与编解码器/解法器 → 不同层面的计算机；
相关课程： 《高级语言程序设计》 《汇编语言程序设计》 《编译原理》 《计算机网络》 《操作系统》 《Internet》；

分层次抽象-自动化机制示例

--操作系统对设备的分层次管理

战德臣

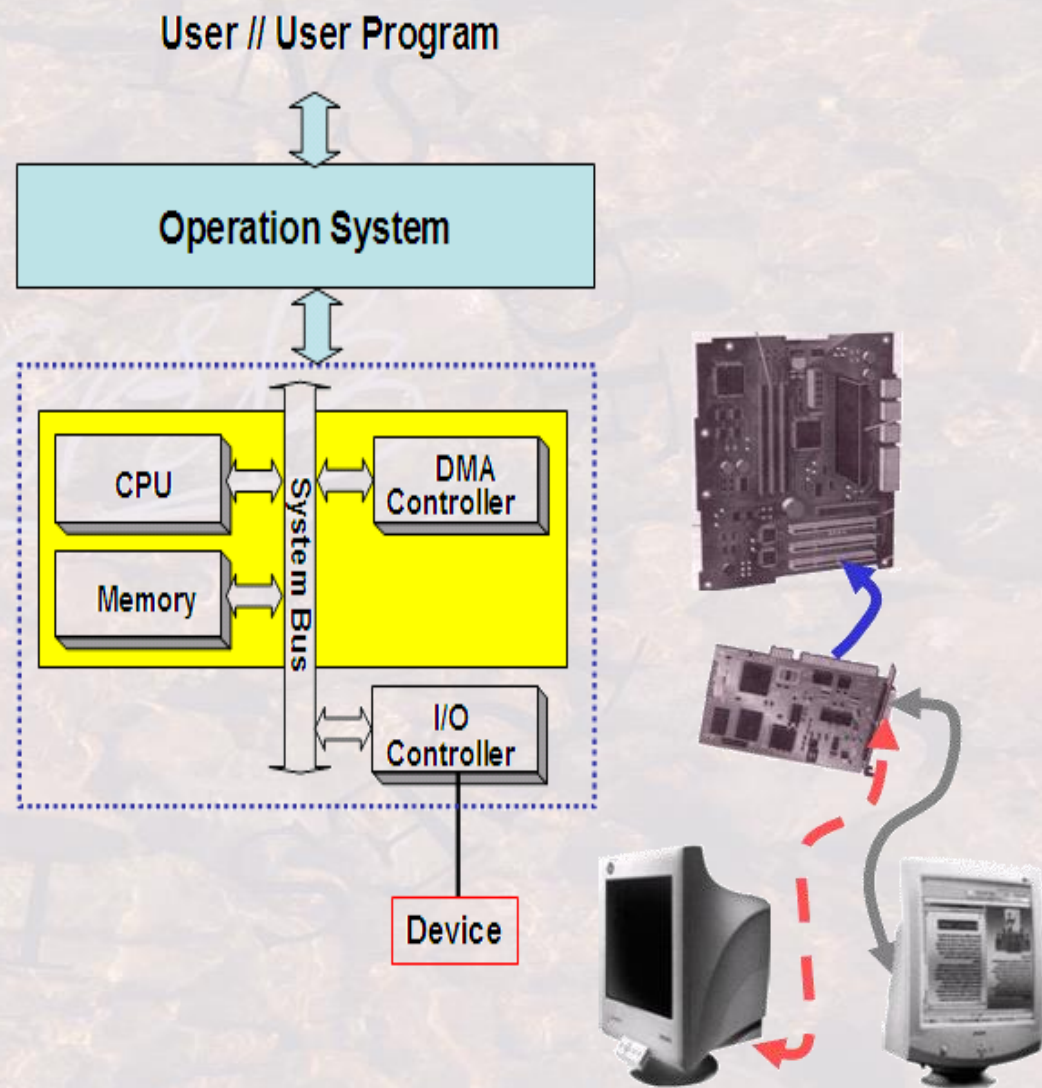
哈尔滨工业大学 教授.博士生导师
教育部大学计算机课程教学指导委员会委员



Research Center on Intelligent
Computing for Enterprises & Services,
Harbin Institute of Technology

分层次抽象-自动化机制示例--操作系统对设备的分层次管理

(1)应用程序、操作系统、硬件和设备之间的关系？

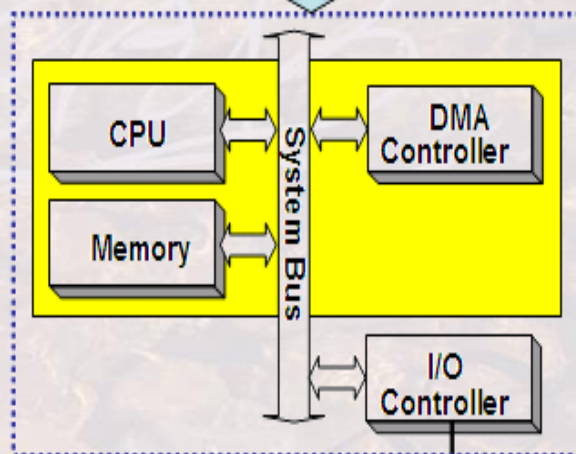



```
#include <stdio.h>
Main()
{ //...
    Printf("Hello, World!\n");//在屏幕上显示Hello World.
    //...
}
```

User // User Program

Operation System

OS怎样控制不同厂家的同类别设备？



厂商1的显示器
(厂商1的设备
操作指令集合)

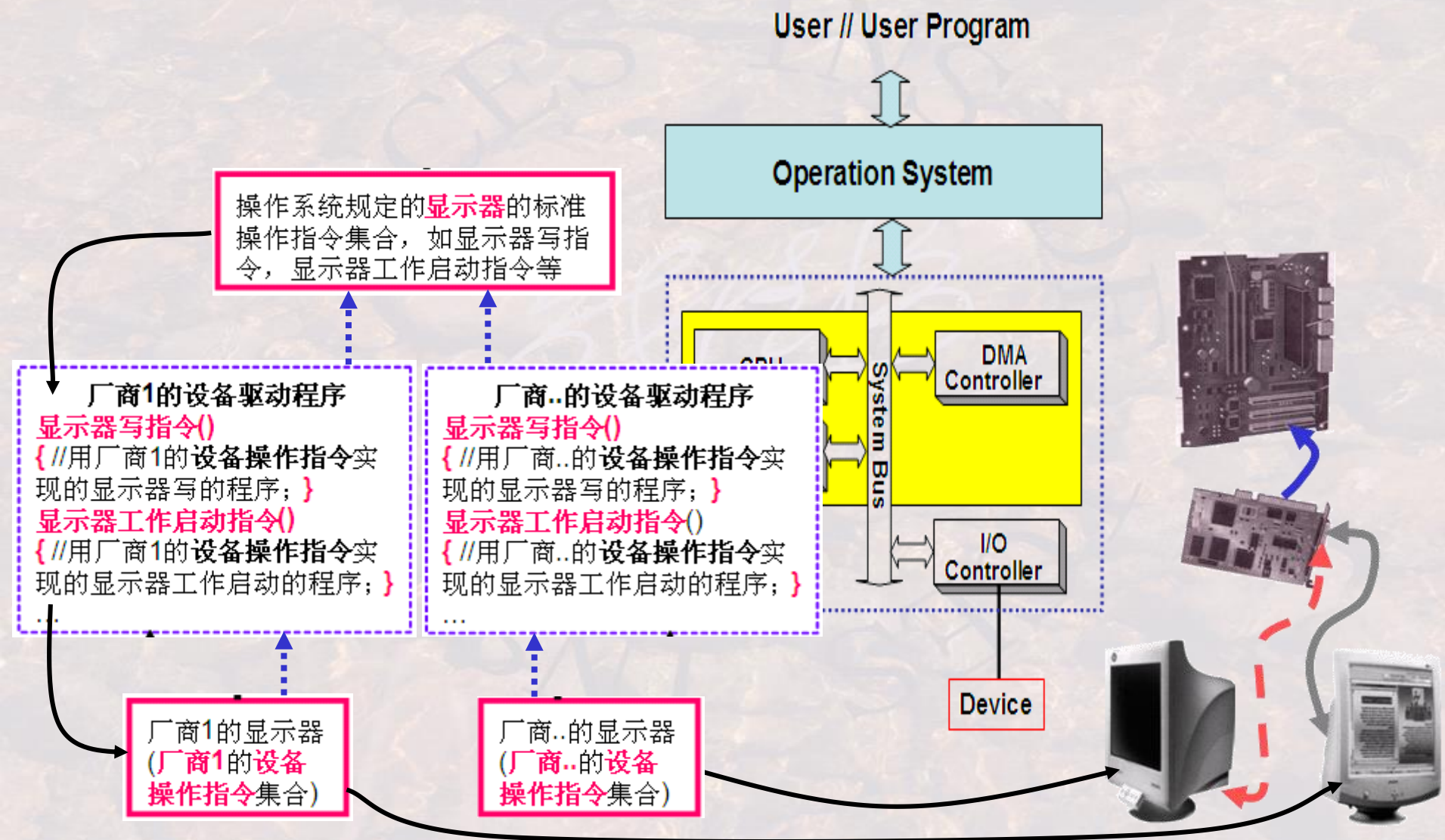
厂商..的显示器
(厂商..的设备
操作指令集合)

Device



分层次抽象-自动化机制示例--操作系统对设备的分层次管理

(2)如何适应不同厂商的同类设备?



(3) 如何使应用程序不考虑设备的具体操作细节？

```
#include <stdio.h>
Main()
{ //...
  Printf("Hello, World!\n");//在屏幕上显示Hello World.
  //...
}
```

User // User Program

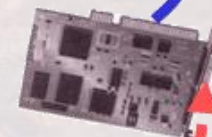
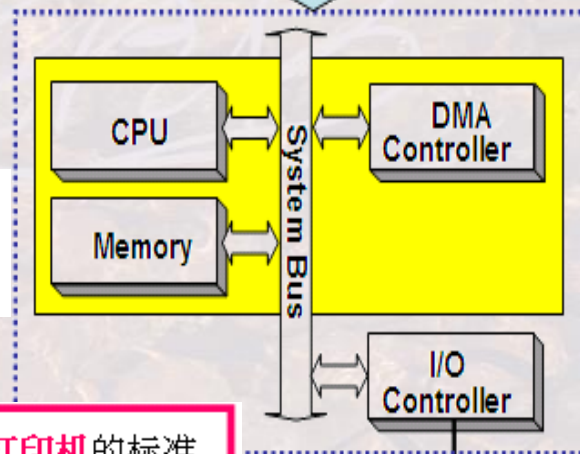
Operation System

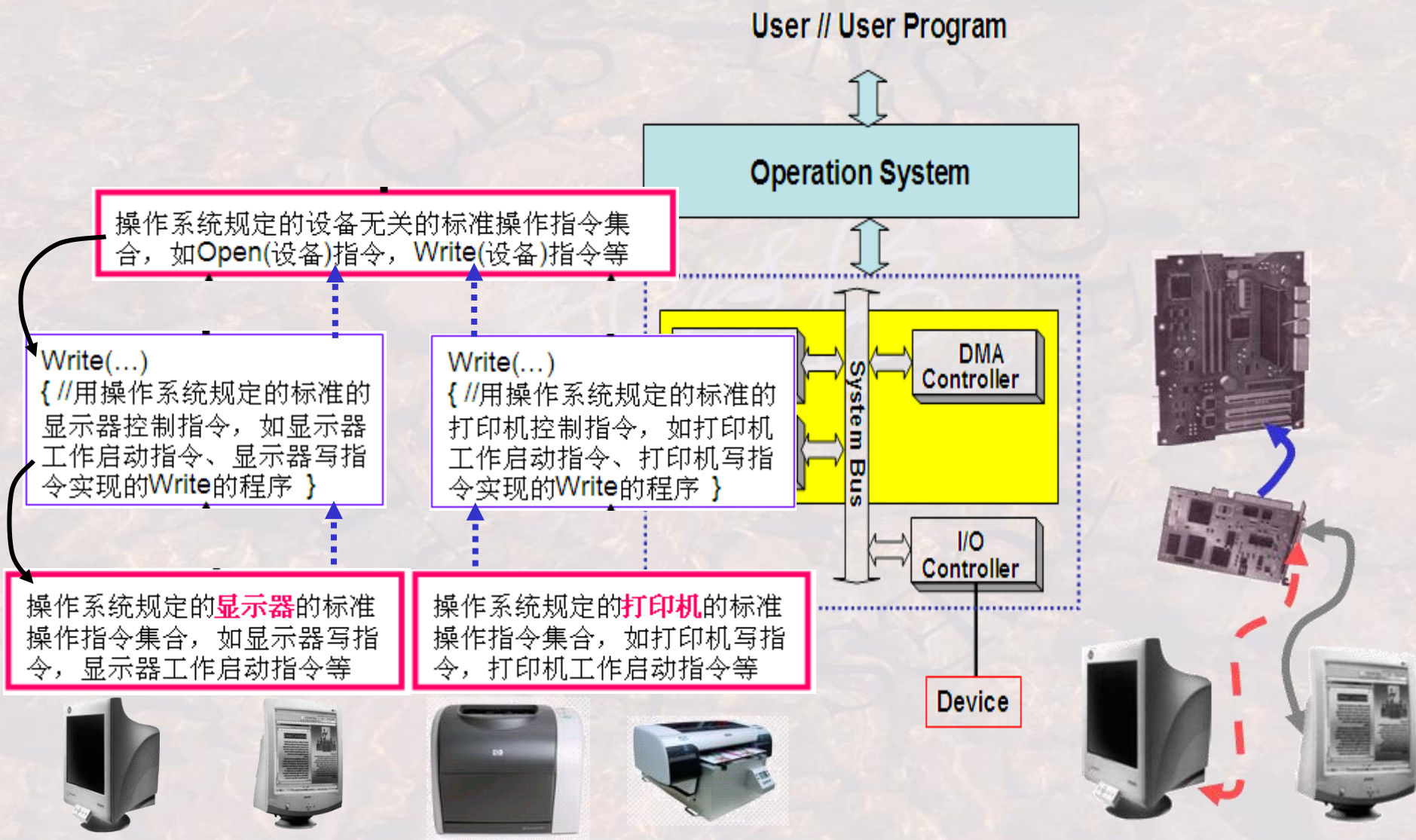
OS怎样让用户不考虑设备的具体操作细节？

OS怎样控制不同类别的设备？

操作系统规定的**显示器**的标准操作指令集合，如显示器写指令，显示器工作启动指令等

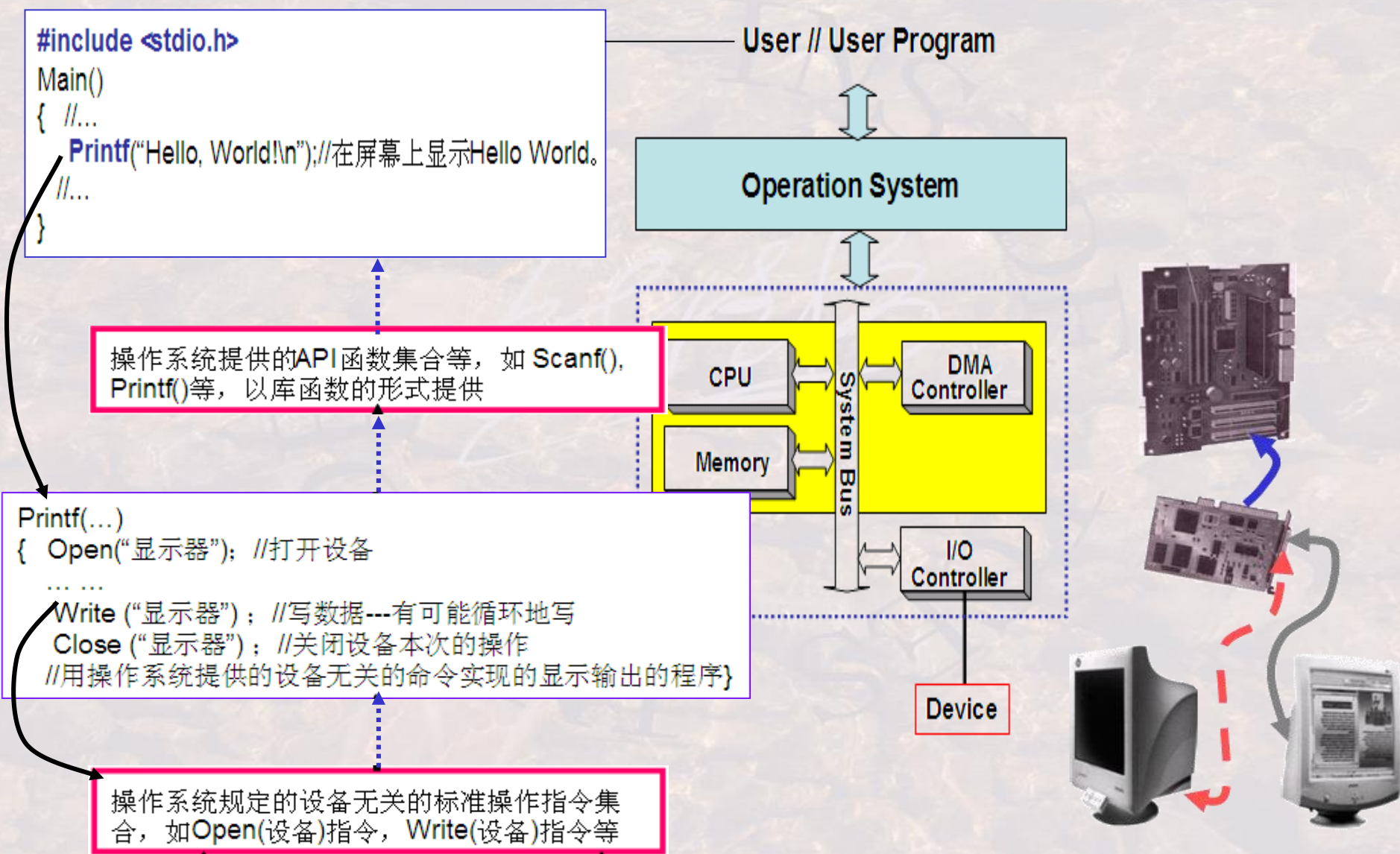
操作系统规定的**打印机**的标准操作指令集合，如打印机写指令，打印机工作启动指令等





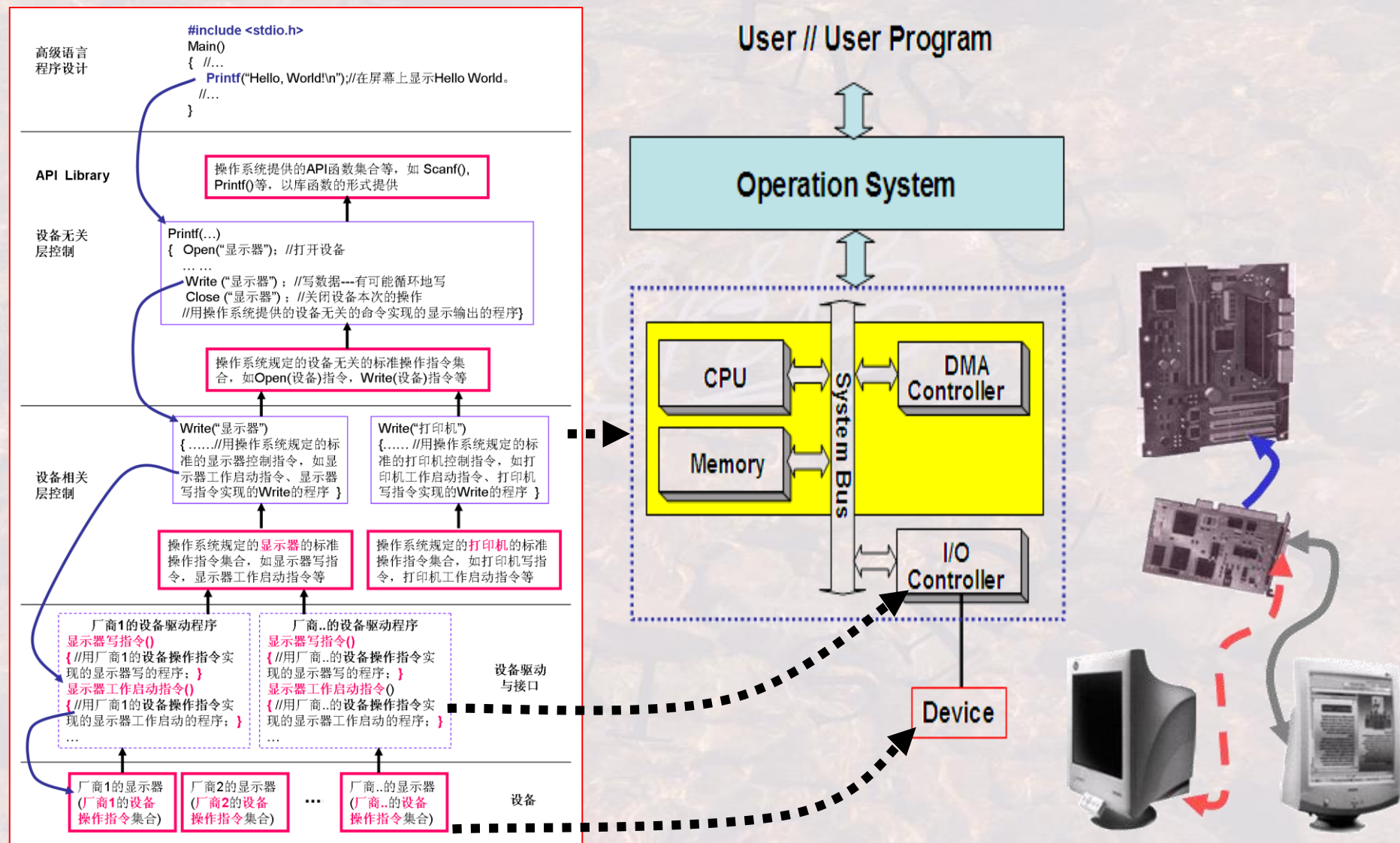
分层次抽象-自动化机制示例--操作系统对设备的分层次管理

(3) 如何使应用程序不考虑设备的具体操作细节?



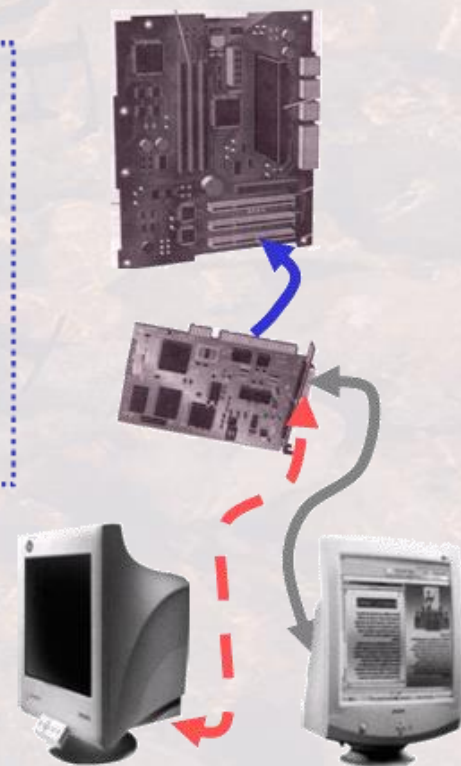
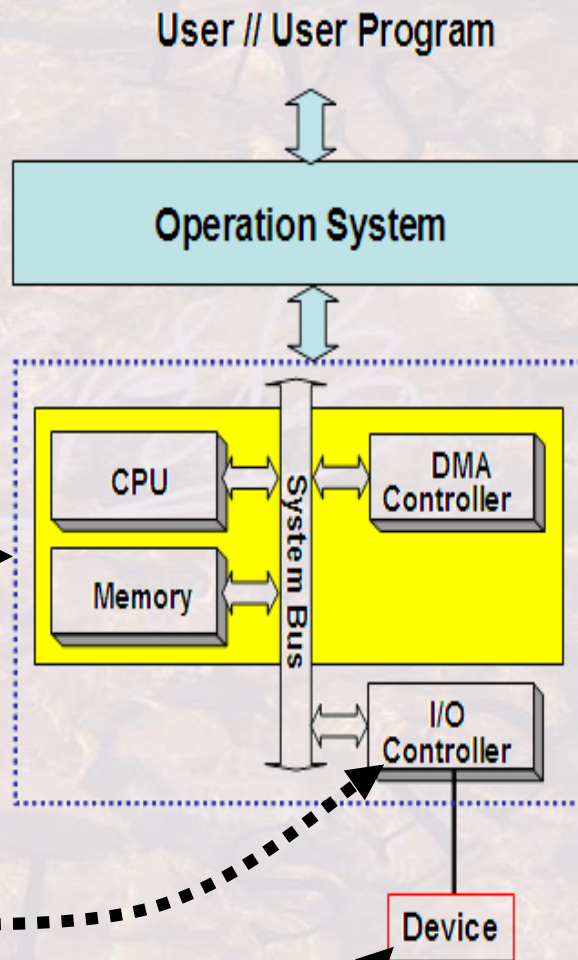
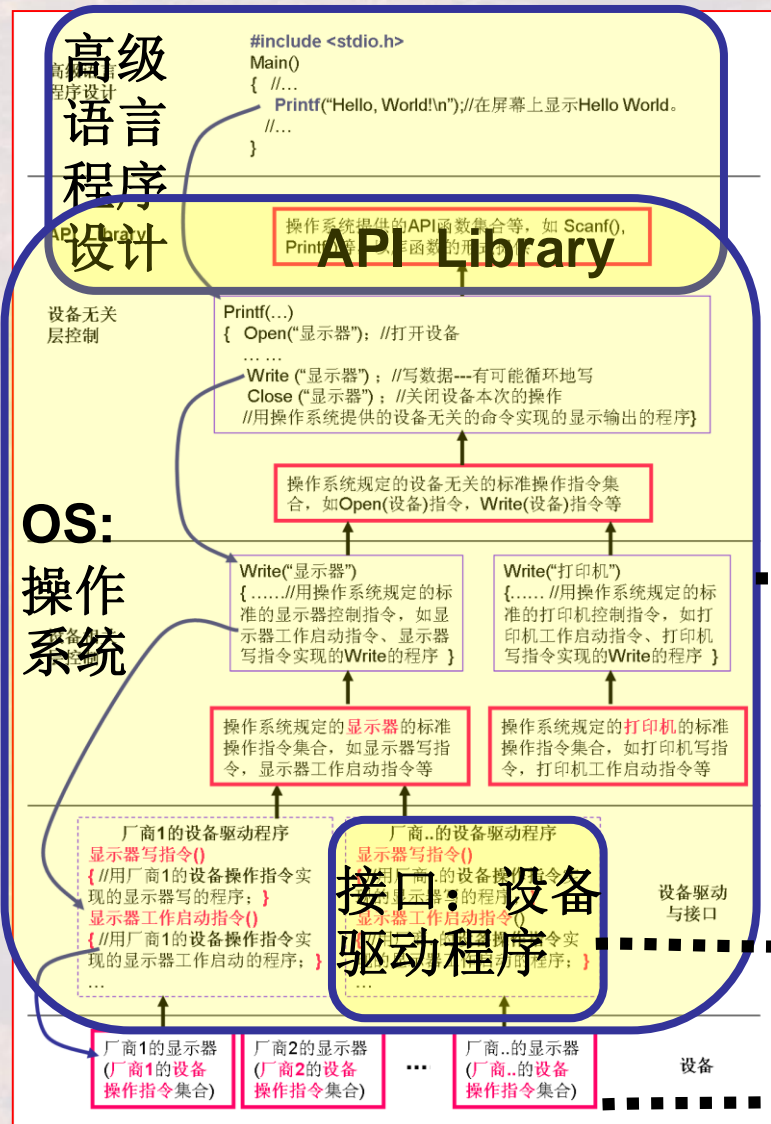
分层次抽象-自动化机制示例--操作系统对设备的分层次管理

(4)分层次管理一览？



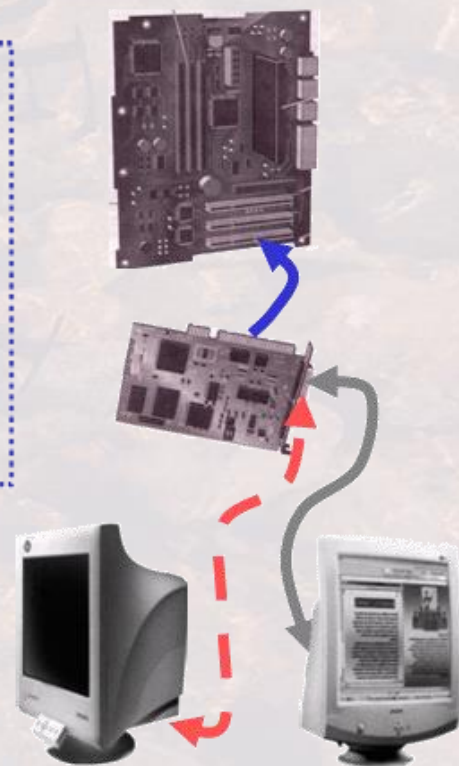
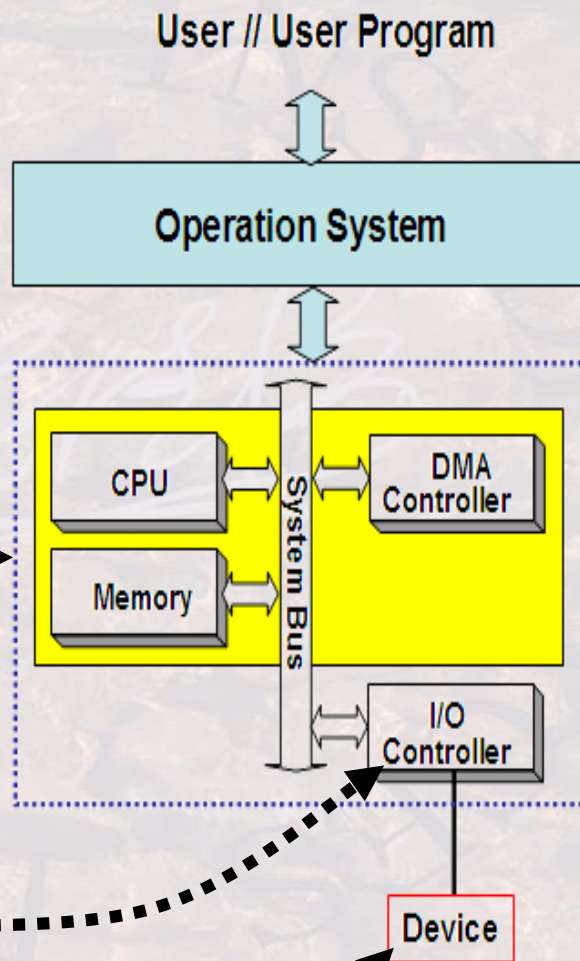
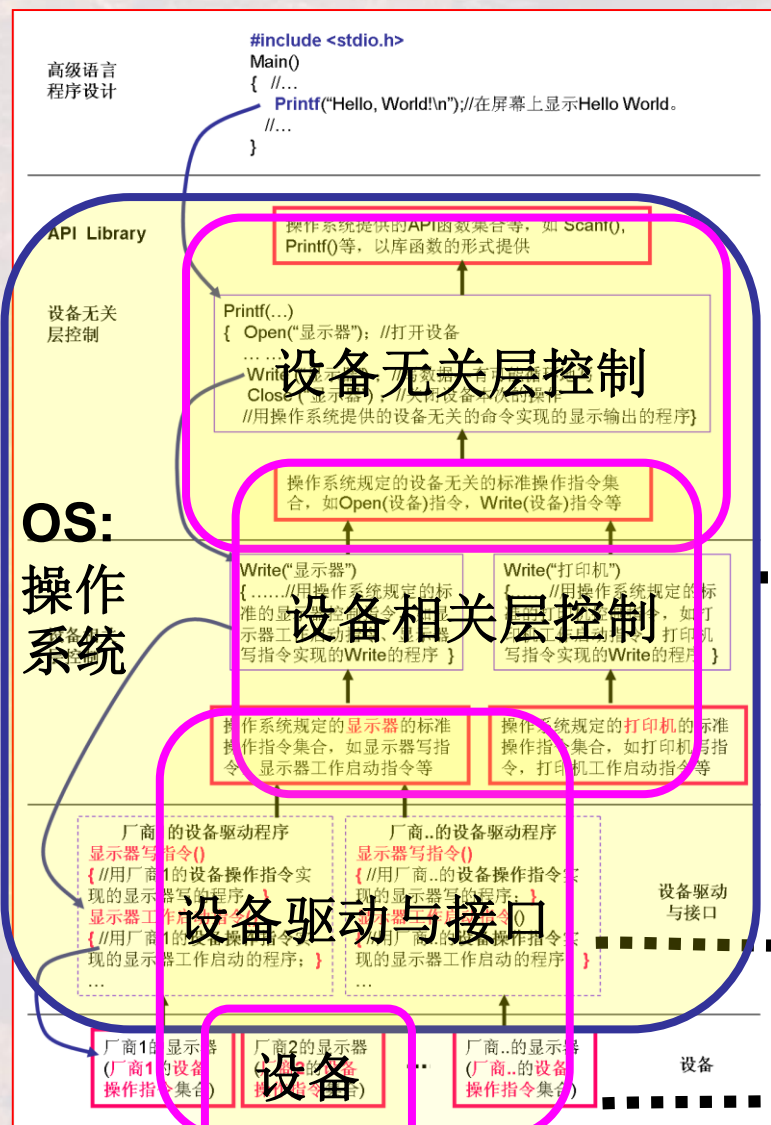
分层次抽象-自动化机制示例--操作系统对设备的分层次管理

(4)分层次管理一览？



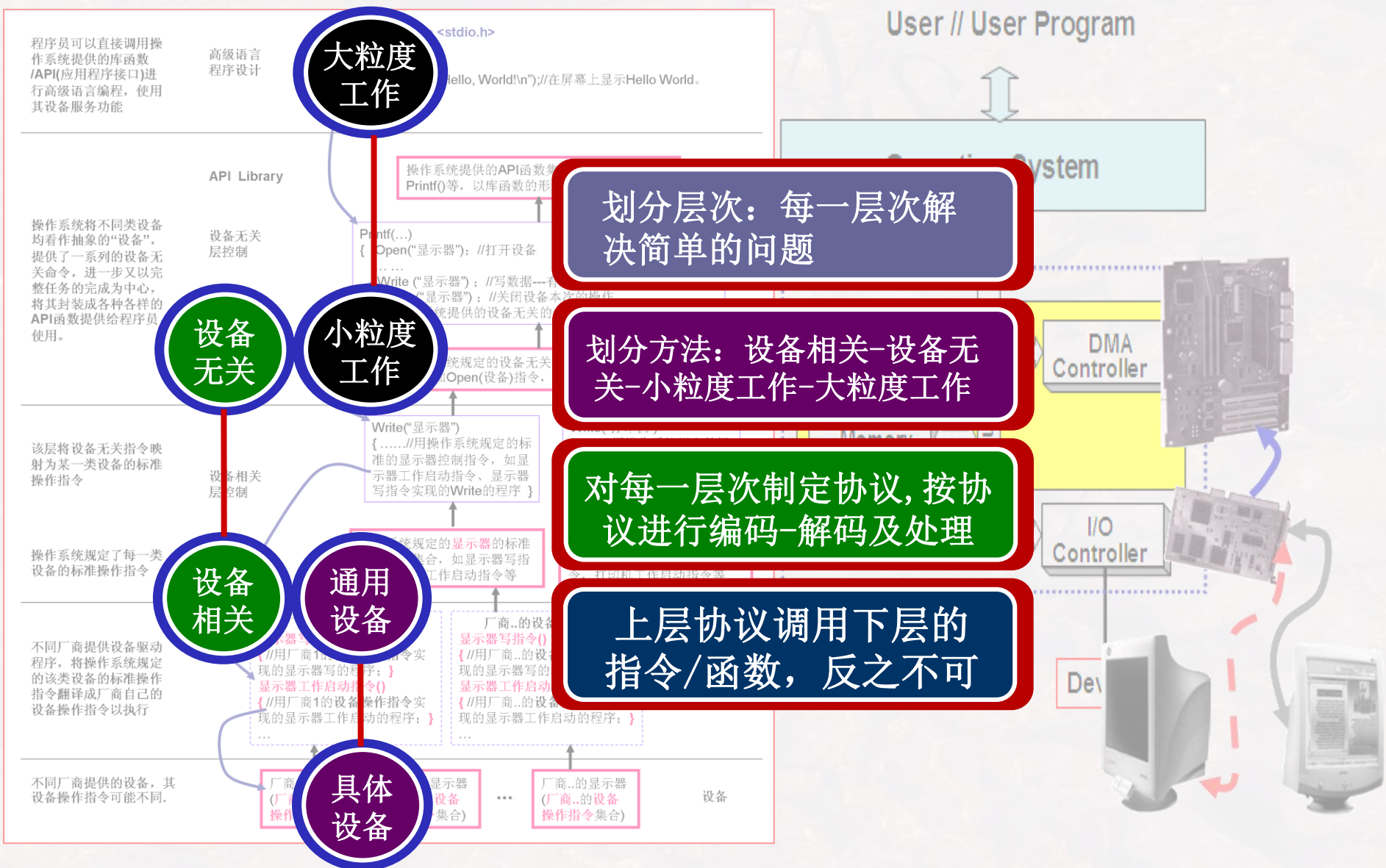
分层次抽象-自动化机制示例--操作系统对设备的分层次管理

(4)分层次管理一览？



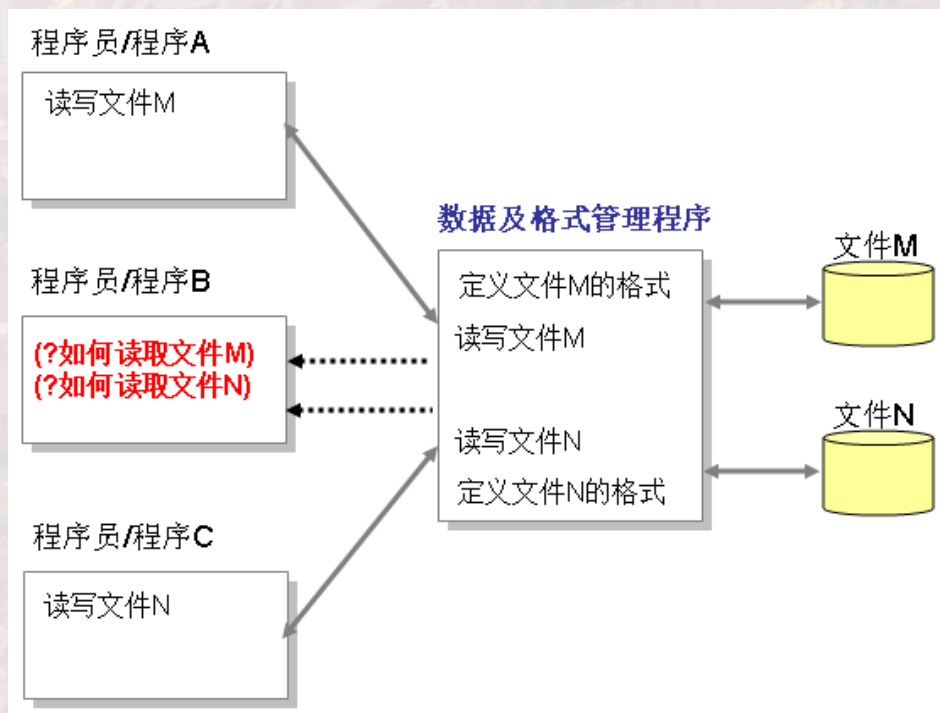
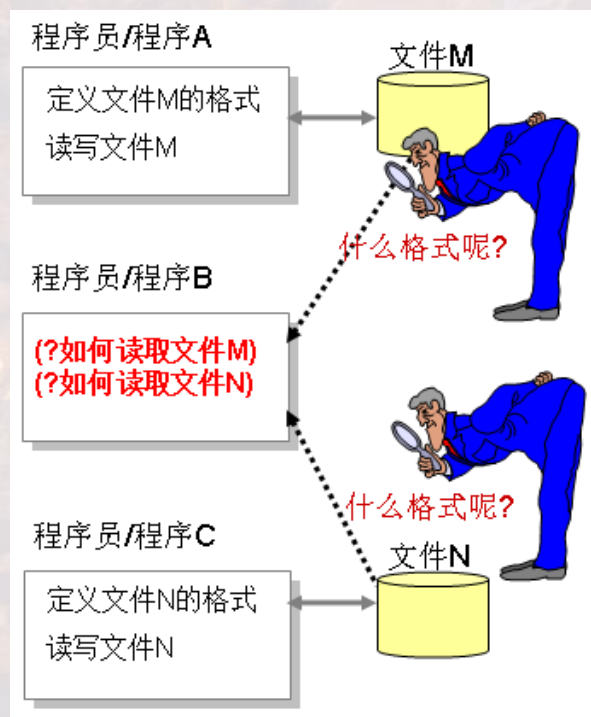
分层次抽象-自动化机制示例--操作系统对设备的分层次管理

(5)分层次抽象-自动化的启示?



(6) 计算学科分离-分层思维的其他示例?

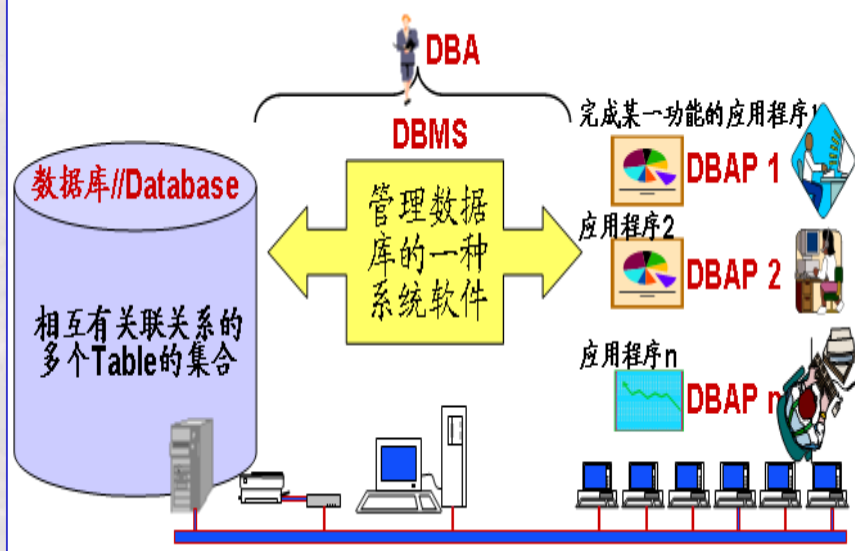
分离的思想——将共性的内容分离出来以单独的软件来管理，通过逐渐剥离共性内容使复杂问题的可解决内容越来越多，进而到最终求解的思想。



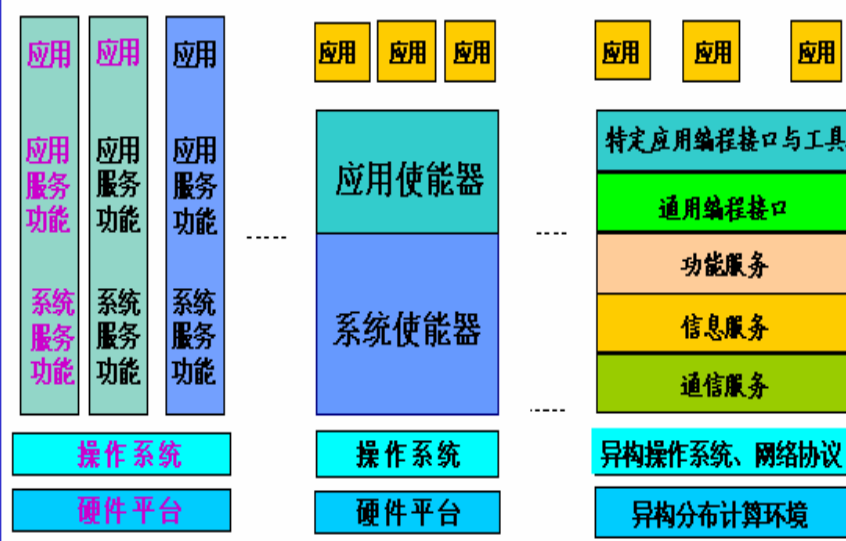
(6) 计算学科分离-分层思维的其他示例？

分离的案例——从应用软件中分离出“数据库管理系统”，形成独立的系统软件产品；从应用软件中不断分离形成“中间件”系统软件。

示例: DBMS

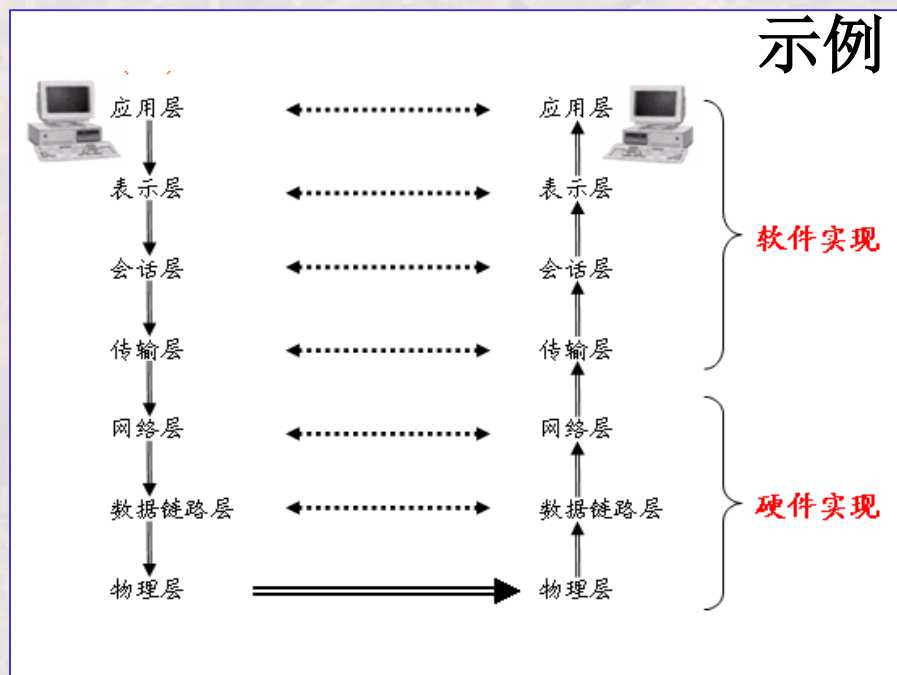
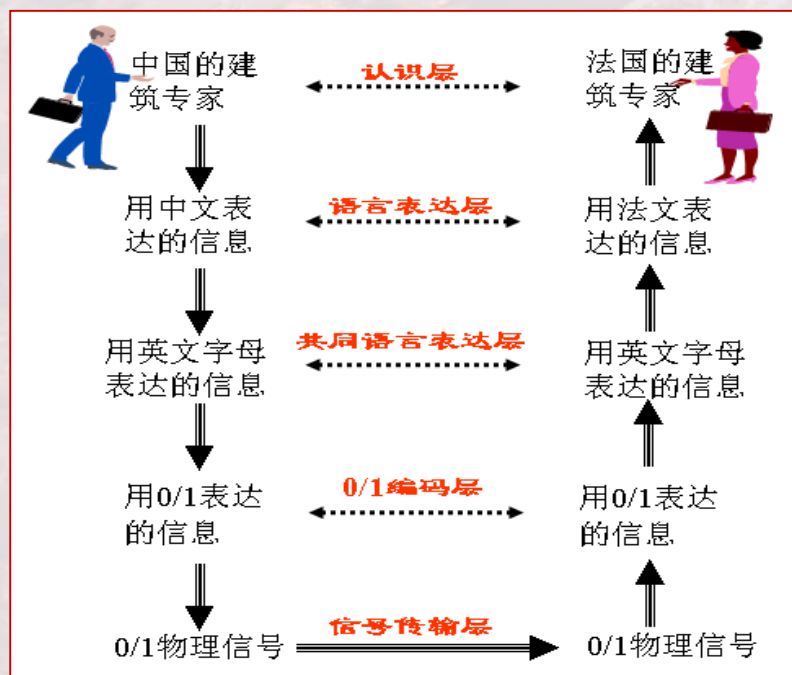


示例:中间件



(6) 计算学科分离-分层思维的其他示例？

分层的思想---将复杂的问题分层求解。将一个复杂问题分为不同层面，比如从宏观到微观的若干层，从概念到实现的若干层等，每一层相对来讲比较简单，可清晰定义每一层的协议/标准并编制相应的处理程序。然后再建立高层向低层的转换关系，从而实现复杂问题求解。



分层次抽象-自动化机制示例--操作系统对设备的分层次管理

(7)小结?



- 解决复杂系统问题的基本思维模式
- 计算学科经典的信息处理思维模式