计算机程序设计

刘龙军 副教授

2025年9月9日

目录

一、数据类型

计算机如何表示数据?

记数法—X进制

进制	元素	规则	例子
10进制	{0, 1, 2, . . . , 9}	逢十进一	9+1=10
60进制	{0、1、2, . . . , 59}	逢六十进一	上午11:59:59
2进制	{0, 1}	逢二进一	1+1=10
8进制	{0, 1, 2, . . . , 7}	逢八进一	7+1=10
16进制	{0, 1, . . . , 9, A, B, C, D, E, F}	逢十六进一	F+1=10

Windows 计算器


计算器
查看(V) 编辑(E) 帮助(H)
标准型(T) Alt+1
科学型(S) Alt+2
程序员(P) Alt+3
统计信息(A) Alt+4
历史记录(Y) Ctrl+H
数字分组(I)
基本(B) Ctrl+F4
单位转换(U) Ctrl+U
日期计算(D) Ctrl+E
工作表(W)
0
M-
√
%
1/x
1
2
3
-
0
.
+
=


计算器
查看(V) 编辑(E) 帮助(H)
0
0000 0000 0000 0000 0000 0000 0000 0000
63 47 32
0000 0000 0000 0000 0000 0000 0000 0000
31 15 0
十六进制
十进制
八进制
二进制
四字
双字
字
字节
Mod A MC MR MS M+ M-
( ) B ← CE C ± √
RoL RoR C 7 8 9 / %
Or Xor D 4 5 6 * 1/x
Lsh Rsh E 1 2 3 -
Not And F 0 . +
=

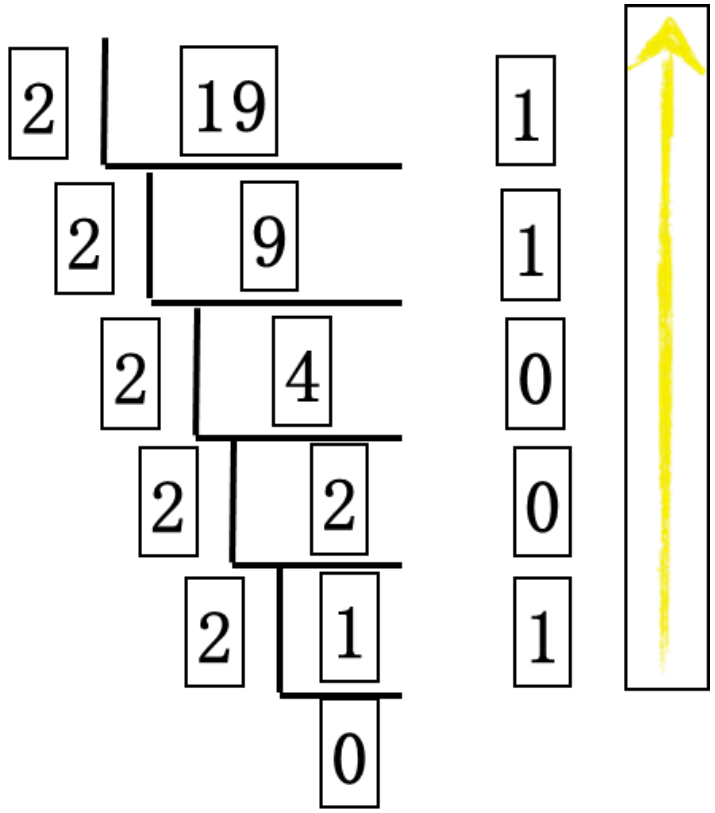
二进制

- 世界上只有10种人，1种人懂二进制， 1种人不懂二进制（读法）
- 0010110,　　0101,　　11010,　　01010
- **为什么二进制适用于计算机？**
- 二进制数在电气元件中容易实现、容易运算，在电子学中具有两种稳定状态以代表0和1，如电压的高和低、电灯的亮和灭、电容的充电和放电、脉冲的有和无、晶体管的导通和截止等。**(计算机就适合高速处理简单、有规律、重复性的“劳动”)**

二进制与十进制的互换

- **十进制 ----→ 二进制**
- $(19)_{10} = (10011)_2$
- **二进制 -----> 十进制**
- **$(10011)_2$ =**

$$1\times2^4+0\times2^3+0\times2^2+1\times2^1+1\times2^0 = (19)_{10}$$



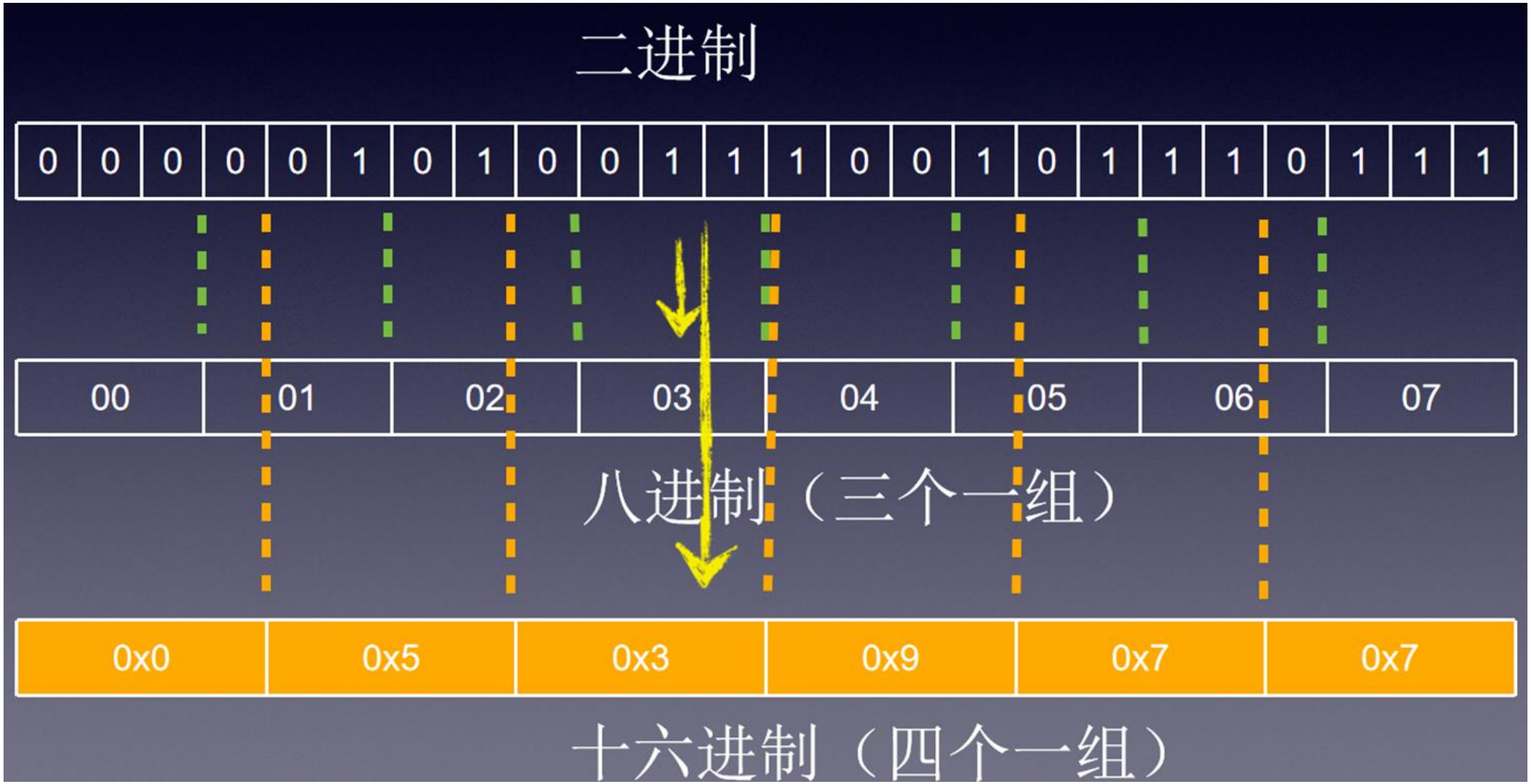
C语言中的八进制、十六进制

- 八进制：
  - 以数字“0”开始的整数
  - 011和11不同，即$(11)_8=1\times8+1=(9)_{10}$ （**如果转2进制再转10进制呢？**）
- 十六进制：
  - 以“0x”或“0X”开始的整数
  - A～F和a~f用来表示十进制的10~15: 0xFF
  - 0x11，即$(11)_{16}=1\times16+1=(17)_{10}$ **（如果转2进制再转10进制呢？）**

Int a = 0b01011（早期编译器不支持二进制）；

int a = 02673； int a= 0x3a； int a= 1001； 程序例子

二进制与八进制、十六进制的换算



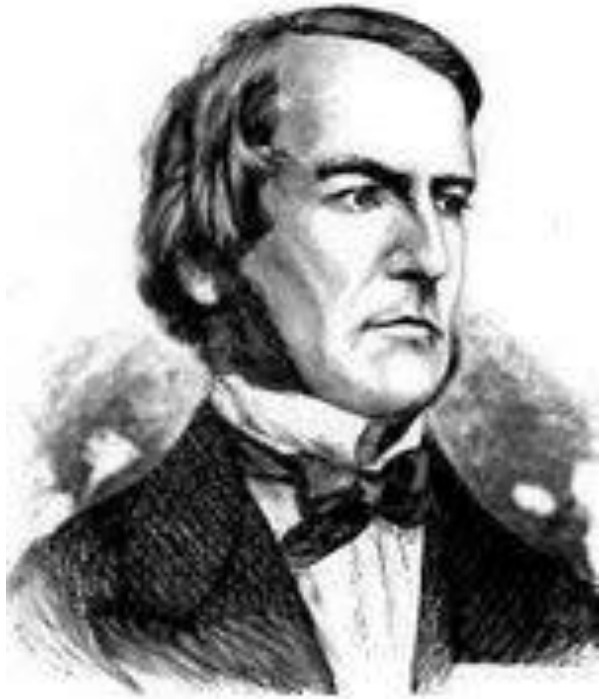
数据的单位---与实际结合

说明	换算关系	读法
bit	最小单位{0,1}	比特、bit
Byte	1 Byte = 8 bit	字节、bit
KB	1 KB = 1024 B	千、kilobit
MB	1 MB = 1024 KB	兆、megabit
GB	1 GB = 1024 MB	吉、gigabit
TB	1 TB = 1024 GB	太、terabit
PB	1 PB = 1024 TB	拍、petabit
EB	1 EB = 1024 PB	艾、eksabit

你的硬盘与U盘分别是多大？能存储多少1或0？

一个bit有多大

- 只能是“0”或者“1”，这叫二进制
- 二进制诠释了计算机的哲学
- 种类众多的复杂事务都是若干种简单事物构成



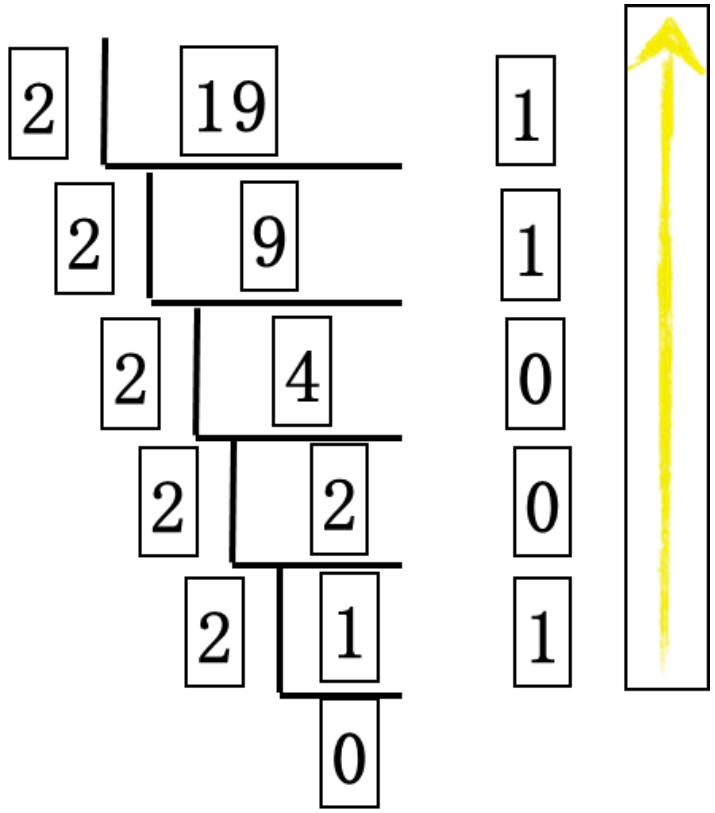
George Boole

二进制与十进制的互换

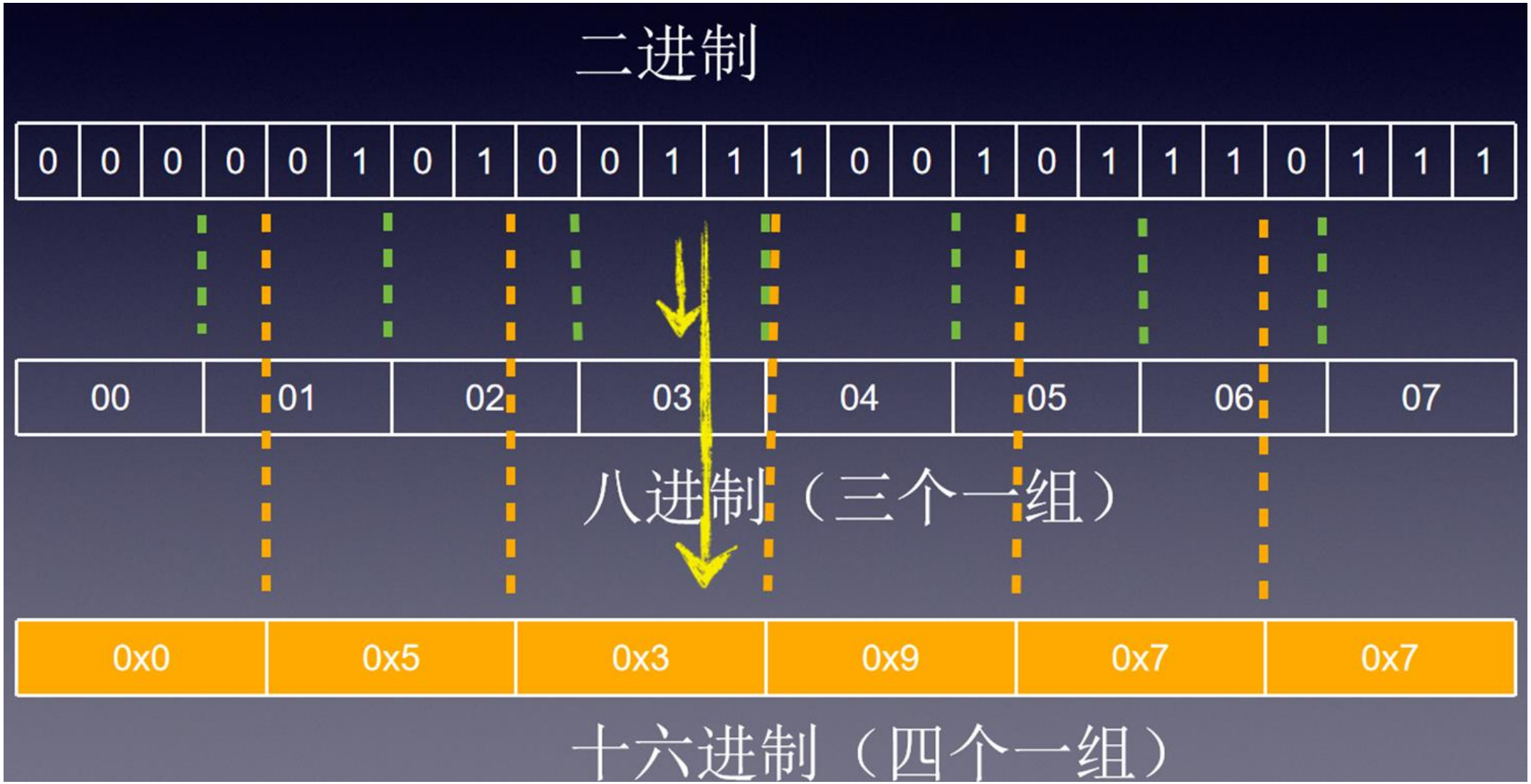
- **十进制 ----→ 二进制**
- $(19)_{10} = (10011)_2$
- **二进制 -----> 十进制**
- **$(10011)_2$ =**

$$1\times2^4+0\times2^3+0\times2^2+1\times2^1+1\times2^0 = (19)_{10}$$

（八进制 8，十六进制 16）

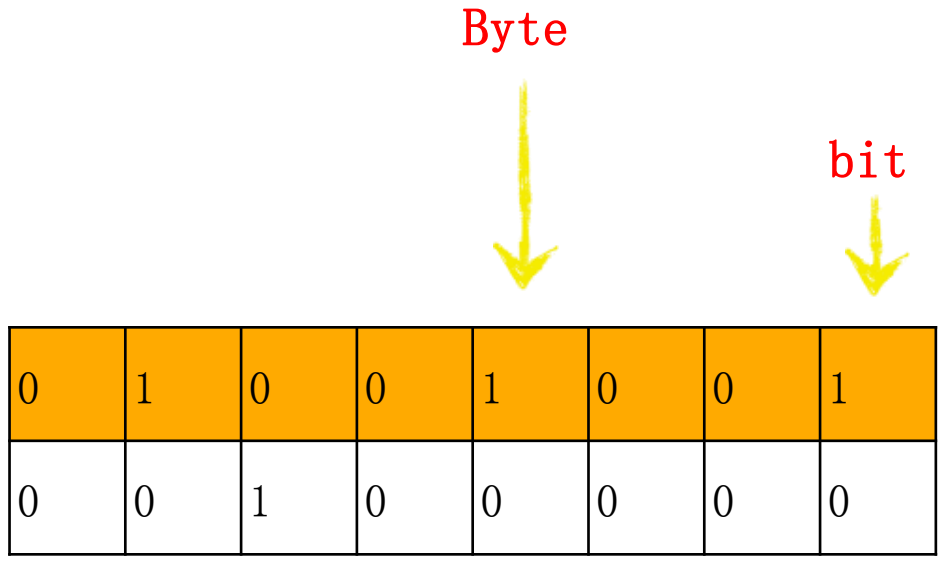


二进制与八进制、十六进制的换算



一个Byte有多大---思考一下存完数据 咋样存字符 汉字

- 可以表示数字0～255
- 保存一个字符（英文字母、数字、符号）
  - ASCII码
- 两个字节保存一个汉字
  - GB2312，6763字
  - GB18030，27533字
  - BIG5，13000字



ASCII码(American Standard Code for Information Interchange)：美国信息交换标准代码

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEI

一个Byte有多大

- 可以表示数字0～255
- 保存一个字符（英文字母、数字、符号）
  - ASCII码
- 两个字节保存一个汉字
  - GB2312，6763字
  - GB18030，27533字
  - BIG5，13000字

两个字节保存汉字

0	1	0	0	1	0	0	1
0	0	1	0	0	0	0	0

ACSII码翻译

- 某男生在某女神朋友圈发回复老被女神无视。

于是该男生用神秘数字开始轰炸该女神在朋友圈发的自拍照：

0b1010011010011010100001010100100101010000101011010000010010000101。终于引起女神注意，女神百思不得其解，只好求助闺蜜。闺蜜赐一程序遂解谜团，该女神顿时觉得这男生很神秘！

ASCII对照表

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

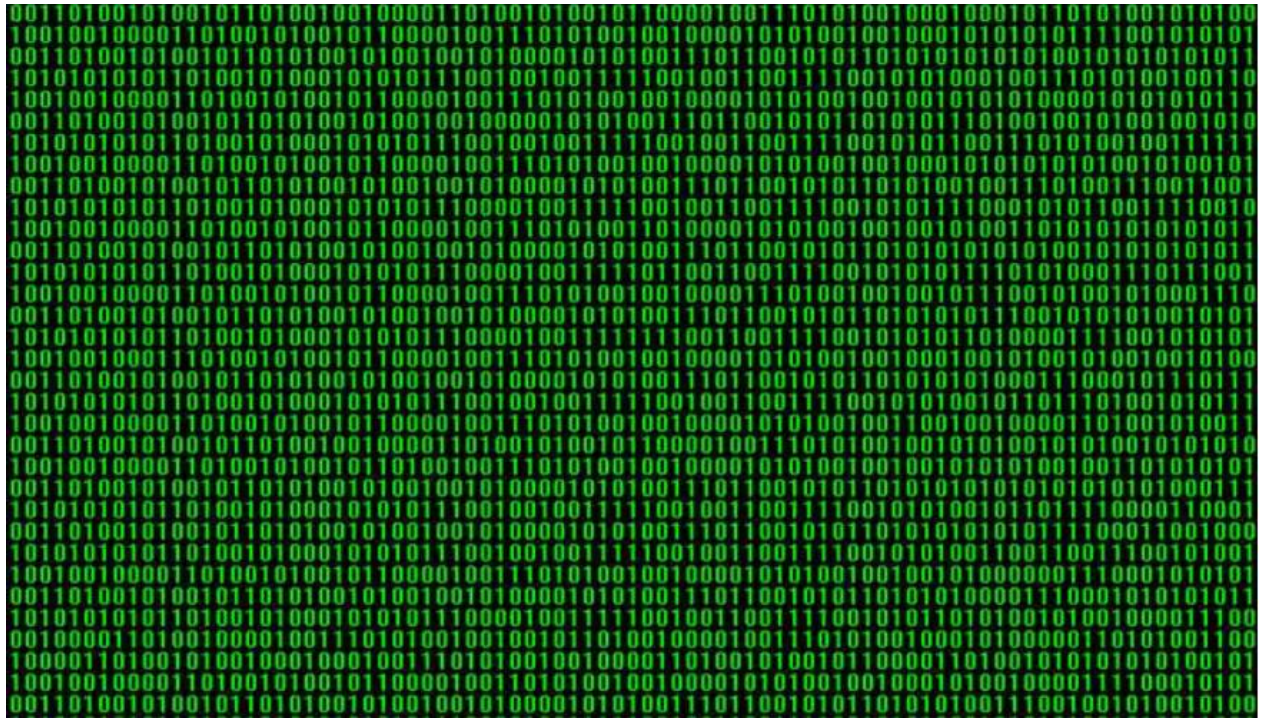
ACSII码翻译


```
1010011010011010100001010100100101010000101011010000100100010 1
5   3   4   D   4   1   5   2   5   4   2   B   4   2   4   5
```

```
#include <iostream>
int main(){
    using namespace std;
    char c0 = char(0x53);
    char c1 = 0x4d;
    char c2 = 0x41;
    char c3 = 0x52;
    char c4 = 0x54;
    char c5 = 0x2b;
    char c6 = 0x42;
    char c7 = 0x45;
    char c8 = 0x41;
    char c9 = 0x55;
    char c10 = 0x54;
    char c11 = 0x59;
    char c12 = 0x3d;
    char c13 = 0x55;
    //cout<<oct;
    cout<<c0<<c1<<c2<<c3<<c4<<c5<<c6
    <<c7<<c8<<c9<<c10<<c11<<c12<<c13<<endl;
    return 0;
}
```

数据在内存中的存储模型

- 想想 图片、视频 等其它数据在计算机里面怎么存储？



- 你们看到的美丽图片、美景等其实在计算机眼里都是０１０１

上节课内容回顾

- 程序观其大略：
  - 把程序里面的参数改改、输入输出改改等，看看是否出错，是否出现其它现象
- 好的代码风格：整齐！缩进！空格（Tab）空行！注释！命名
- 写程序、上机、调试、实践出真知、动手、练会、上网搜
- 避免一看就懂、一做不会，（脱机直接码出来）
- 练习盲打、非字母部分练熟、记忆快捷键（搜索），抄写代码、输出！
- 数据、进制转换、单位
- 快速写出该数的十六进制转换 0b00001010011100101110111

程序处理的基本对象

变量和常量（从计算机底层组织看起）

上一节课回顾

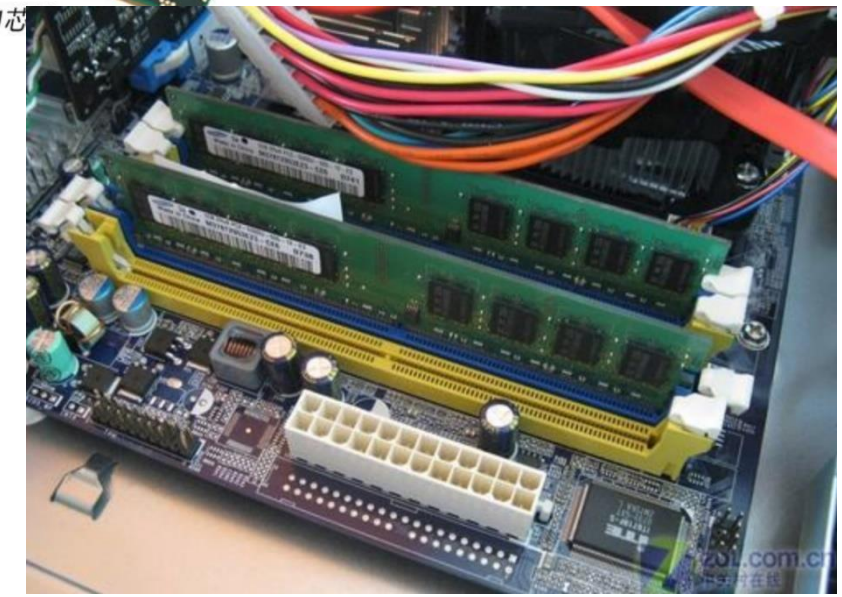
- 以本门课为 计算机系列课的“引导”，
- 不“就书论书”
- 按照上课扩展的内容进行适当扩展，AI专业不同基础的学生，本学年都适用。
- 多积累计算机相关知识，而不是只拿着计算机去打游戏。

上课 教学 VS 网上视频； 扩展的基础贯穿

AI应用、引导

计算机硬件

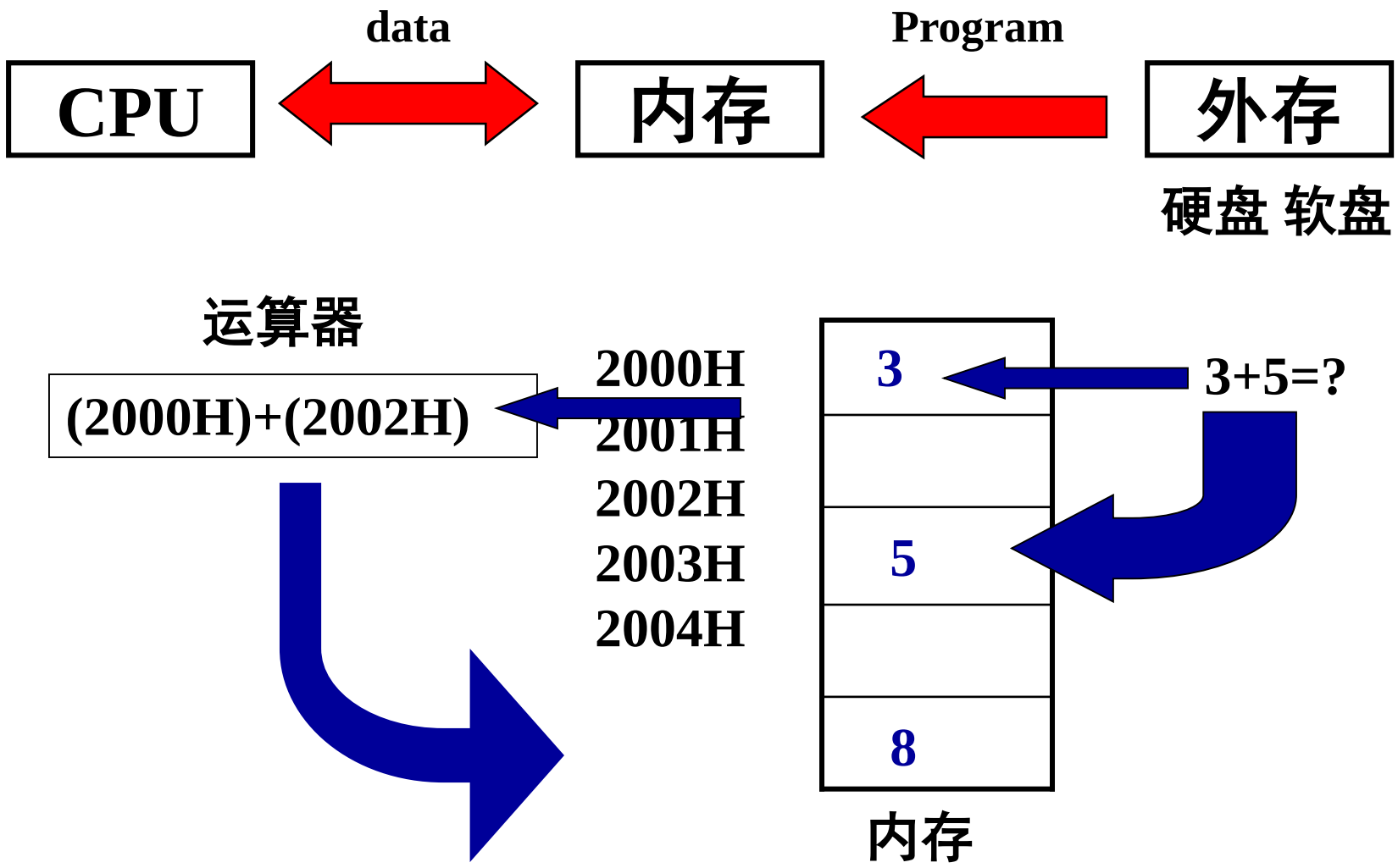
CPU芯



- 代码是存在哪里的? 变量在哪里？如何运行的？ 硬件、程序结合

思考！ 机器人系统

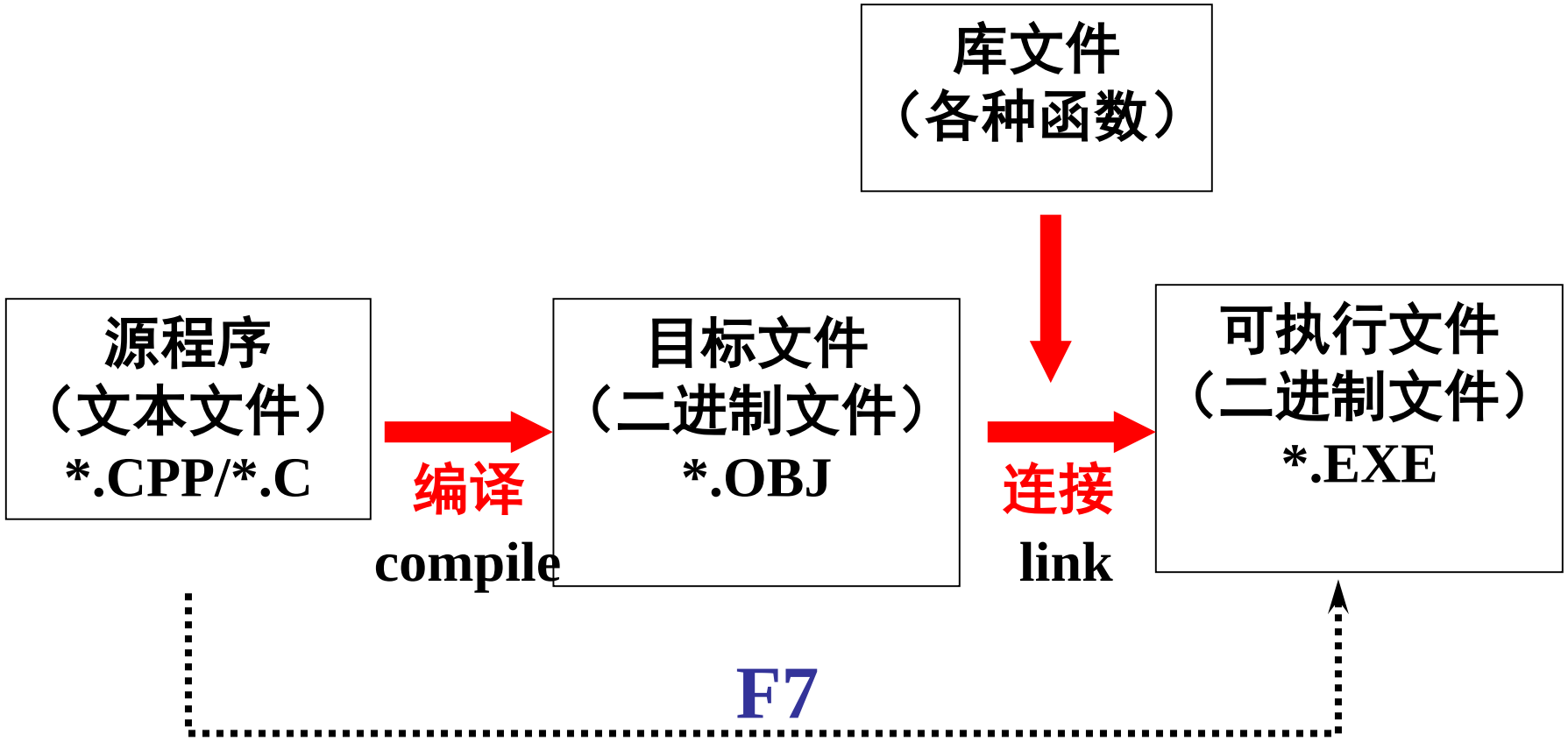
存储程序并执行的过程


data
Program
CPU
内存
外存
硬盘 软盘
运算器
(2000H)+(2002H)
2000H
2001H
2002H
2003H
2004H
3
5
8
3+5=?
内存

变量

- 其值可以改变的量称为**变量**。
- 目的：在程序运行过程中用来**保存待处理的数据**，保存**中间**或最终结果
- **本质：从内存中划拉出来一块空间，这个空间对应了一个内存地址，这个复杂的地址需要一个名字方便操作**
- **先声明后使用**

编译过程


库文件
（各种函数）
源程序
（文本文件）
.CPP/.C
编译
compile
目标文件
（二进制文件）
*.OBJ
连接
link
可执行文件
（二进制文件）
*.EXE
F7

变量命名

- 命名规则：变量名只能由字母、数字和下划线三种字符组成，且第一个字符必须是字母或下划线
  - 合法变量名：a, a1, a_1, _a
  - 不合法变量名：a+b, $123, 1.5b, 3GB
- 变量名区分大小写
  - sum, SUM代表不同的变量名
- 变量名要直观，能表达它的功能，英文单词的缩写（代码国际化！）

变量的注意事项

- 变量需要“先定义，后使用”
- 为什么？

  程序需要根据变量类型分配内存单元

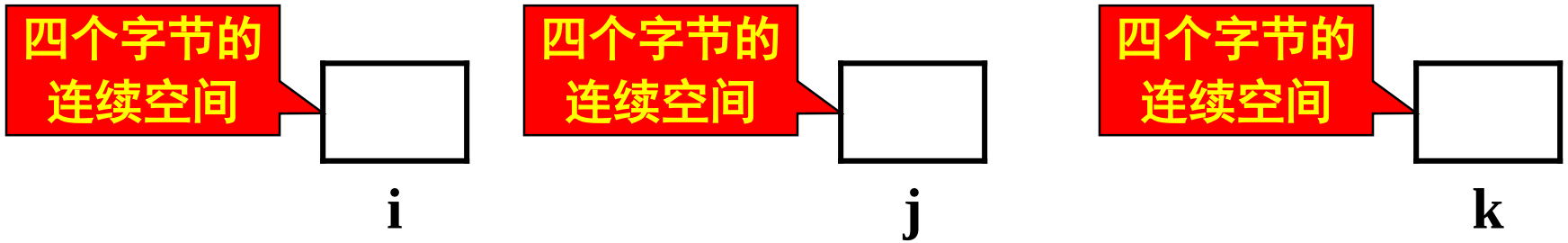
  保证变量名使用正确

  便于检查程序当中的操作运算是否合法

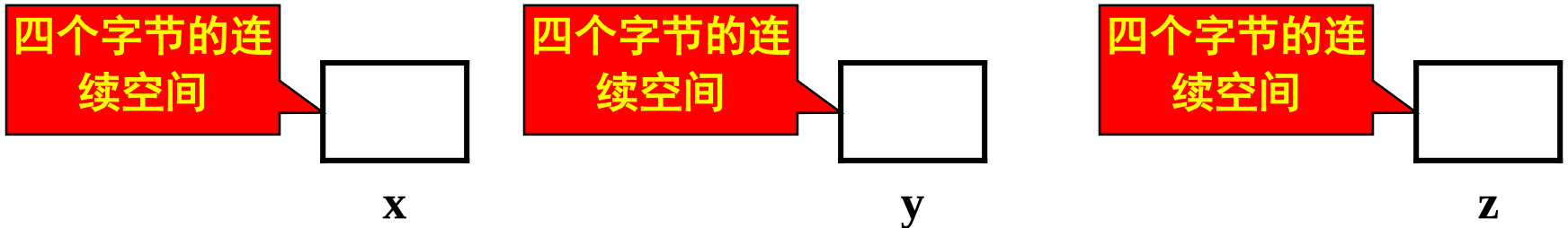
```
int main()
{
  int data;

  date=100;

  return 0;
}
```

```
int main()
{
  int   a=1;
  float b=1.1;
  float c=1.2;

  a = b % c;

  return 0;
}
```

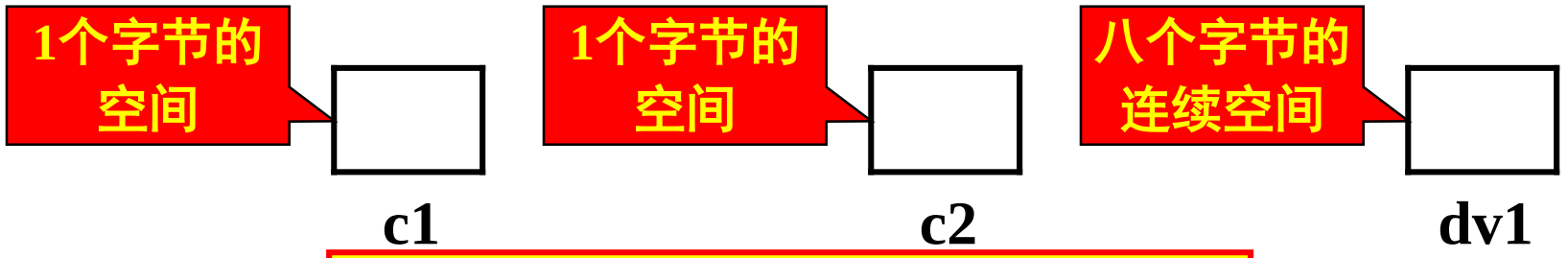
```
int i, j, k;          //定义了三个整型变量i,j,k
```




```
float  x,y,z;         //定义了三个实型变量x,y,z
```




```
char   c1,c2;  //说明了二个字符型变量c1,c2
double  dv1;  //说明了一个双精度型变量dv1
```

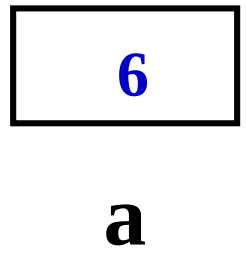


开辟空间后, 空间中为随机值

变量赋初值

- 在定义变量的**同时**给变量赋值，即在内存中开辟出一个空间后马上给此空间赋值。
- 但这个空间的值**并不是固定不变的**，在程序的运行中**一样可以改变**。

```
int   a=4;     //定义语句，在开辟空间后马上为空间赋值
      a=6;     //重新为该空间赋值
```




```
char   a='\x64', b='d';
 a='A';  b='\n';
```

```
int   a1=6,   a2=98;
a1=011;   a2=121;
```

常量

- 常量：不接受程序修改的固定值。
- 常量有三类：
  - 数
  - 字符
  - 字符串

数常量

数常量	合法的表示
整数常量	• 十进制：123，0，-5 • 八进制：015 • 十六进制：0x13
浮点数常量	0.123 ， 123E-3

字符常量

- 单引号括起来的一个字符，如‘a’、‘A’
- 字符常量**其实是一个整数**，其值就是**该字符在所属字符集中的编码值**
- 注意：单引号是**定界符**，它并不是字符常量的一部分

```
// ascii.cpp --  type conversion 2
// from char to int

#include <iostream>

int main()
{
    char ch = 'A';

    std::cout <<ch<<'\t'<<int(ch)<<std::endl;

    return 0;
}
```

查看ASCII码表，想想大小写怎么转换？

字符常量

- 字符型数据实际上是作为**整型数据**在内存中存储的。
- 计算机是以字符编码的形式处理字符的，因此，我们在计算机内部是**以ASCII码的形式表示所有字符的**。所以**7位**二进制数即可表示出一个字符，我们用**一个字节的容量（8位）存储一个字符**。

例如：字符A的ASCII码为0x41或65，在内存中表示为：

0	1	0	0	0	0	0	1

在程序中表示为：

```
char   grade ;//定义一个字符型的变量空间(1个字节)
grade='A';  //必须用 ‘ ’ 表示，否则易与标识符混同
```

‘ ’内括起来的字符表示该字符的ASCII码。

字符常量

进一步，由于在内存中的形式与整型数据相同，所以，可以直接用其整型值给变量赋值。

```
char   grade;
grade=65;
```

以下的赋值形式均是等同的。

```
grade='A';      grade=65 ;      grade=0x41;      grade=0101;
```

```
#include<iostream.h>
void main(void)
{
  char a,b;
  a='A';  //输入ASCII码
  b=65;  //输入十进制数
  cout<<"a="<<a<<endl;
  cout<<"b="<<b<<endl;
}
```

输出：

```
a=A
b=A
```

即在内存中的表示均是相同的

0	1	0	0	0	0	0	1

转义字符常量

- C/C++语言专门设置了一种以反斜杠开头的字符常量用来表示一些无法直接写出的特殊字符

转义字符	含义
\n	回车换行（endl）
\t	横向制表符，类似于Tab
\\	反斜杠“\”
\’	单引号
\”	双引号
\?	问号

转义字符虽然包含2个或多个字符，但它只代表一个字符。编译系统在见到字符“\”时，会接着找它后面的字符，把它处理成一个字符，在内存中只占一个字节。

字符串常量

- 字符串常量：用一对双引号括起来的字符序列， 如“XJTU”，“Love C++”
  - 正确： char c = ‘a’；
    char str[9] = “Love C++”；
  - 不正确：char c = “a”；
- 系统会在字符串末尾加一个字符串结束标志 ‘\0’

字符串常量

C	H	I	N	A	\0

实际上内存是对应字符的ASCII码形式

0x43	0x48	0x49	0x55	0x41	\0

01000011	01001000	01001001	01010101	01000001	00000000

'a'在内存中占一个字节

a

"a"占两个字节

a	\0

01100001

01100001	00000000

符号常量

西安交通大学 XI'AN JIAOTONG UNIVERSITY

宏定义define:

- 让常量表现的更加人性化
- 实现“一改多改” （如果程序中同名会咋样？）

```
// const1.cpp -- show how predefine works
#include <iostream>
using namespace std;
#define PRICE 30

int main()
{
    int num,total;
    num=10;
    total=num*PRICE;
    cout <<"total=" << total << endl;
    return 0;
}
```


total=300

符号常量

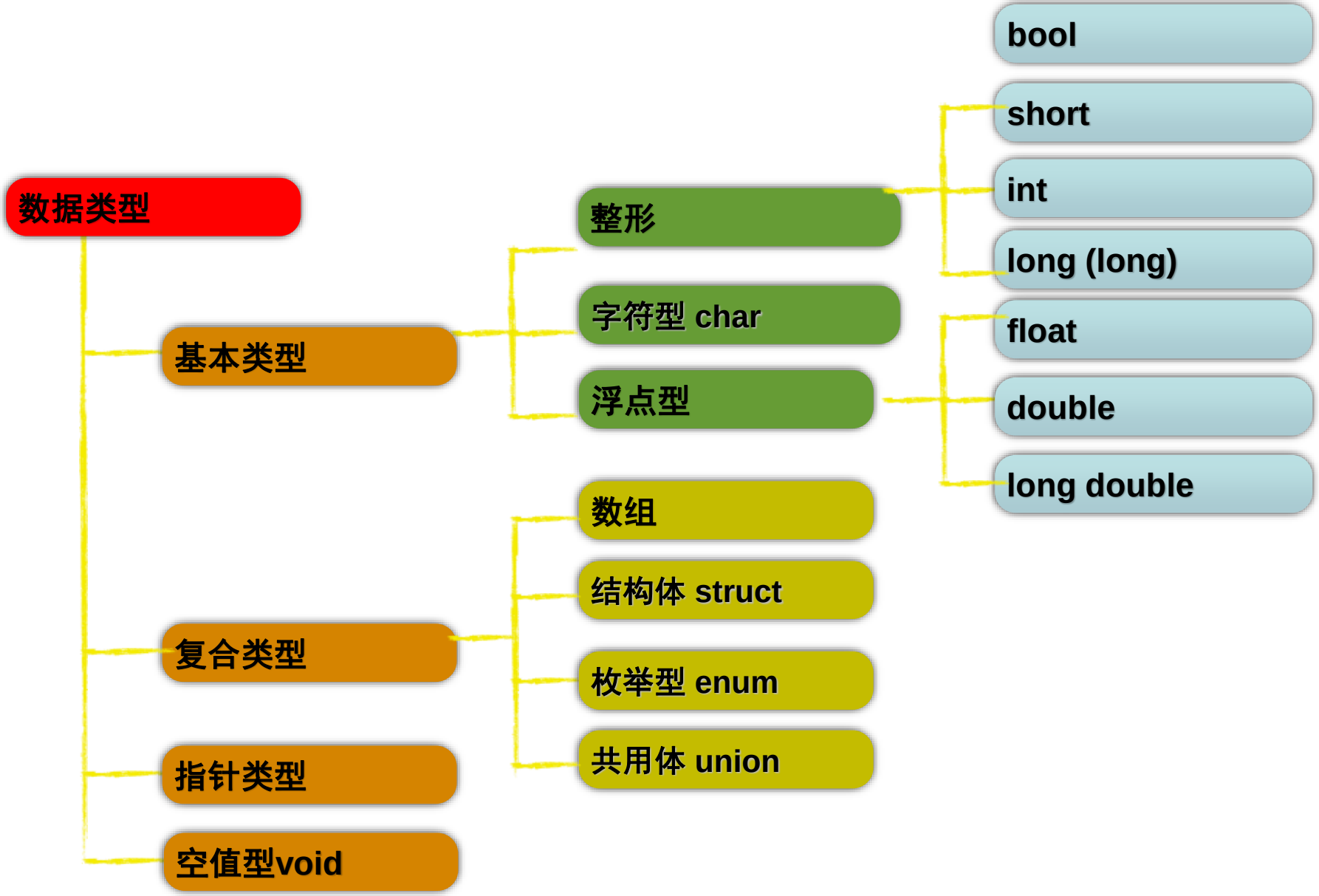
- Const
- 注意区别和用法

```
// const2.cpp -- example of const
#include <iostream>
using namespace std;
int main()
{
    int num,total;
    const int Price=30;
//  const int Price;//value undefined
//  Price=30; //too late
    num=10;
    total=num*Price;
    cout <<"total=" << total << endl;
    return 0;
}
```

常量与变量

- 常量：在程序运行过程中，其值一直保持不变的量为常量。
- 常量也区分不同的类型：30，40 为整型，30.0，40.0为实型，编辑器只是根据其表面形式来判断其类型。
- 变量：在程序运行过程中，其值可以改变的量为变量。
- 变量在程序的执行中能够赋值，发生变化。变量有一个名字，并在使用之前要说明其类型，一经说明，就在内存中占据与其类型相应的存储单元。

数据类型


数据类型
基本类型
整形
bool
short
int
long (long)
字符型 char
浮点型
float
double
long double
复合类型
数组
结构体 struct
枚举型 enum
共用体 union
指针类型
空值型void

基本数据类型

基本类型标识符	含义	长度(Byte)
int	整数	2或4
float	单精度浮点数	4
double	双精度浮点数	8
char	字符	1

基本数据类型的长度

```
// limits.cpp -- some integer limits
#include <iostream>
int main()
{
    using namespace std;
    int n_int = INT_MAX;        // initialize n_int to max int
value
    short n_short = SHRT_MAX;
    long n_long = LONG_MAX;
    long long n_llong = LLONG_MAX;
   // sizeof operator yields size of type or of variable
    cout << "int is " << sizeof(int) << " bytes." << endl;
    cout << "short is " << sizeof(short) << " bytes." << endl;
    cout << "long is " << sizeof(long) << " bytes." << endl;
    cout << "long long is " << sizeof(long long) << " bytes." <<
endl;

    cout << "Maximum values:" << endl;
    cout << "int: " << n_int << endl;
    cout << "short: " << n_short << endl;
    cout << "long: " << n_long << endl;
    cout << "long long: " << n_llong << endl << endl;
   cout << "Minimum int value = " << INT_MIN << endl;
    cout << "Bits per byte = " << CHAR_BIT << endl;
   return 0;
}
```

```
int is 4 bytes.
short is 2 bytes.
long is 8 bytes.
long long is 8 bytes.

Maximum values:
int: 2147483647
short: 32767
long: 9223372036854775807
long long: 9223372036854775807

Minimum int value = -2147483648
Bits per byte = 8
```

数据类型的修饰符

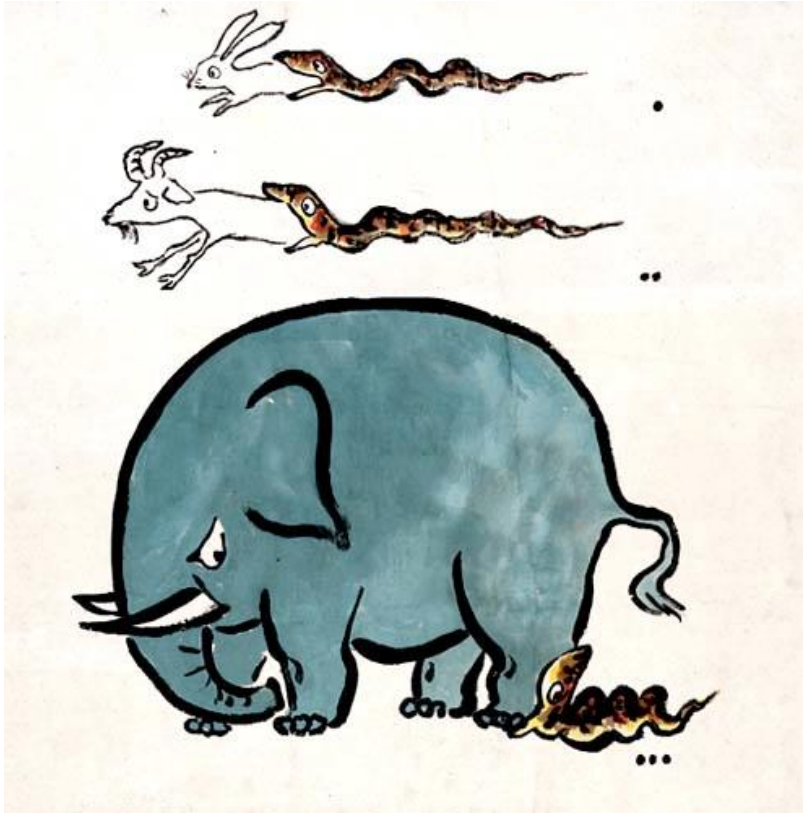
修饰符	含义	长度(Byte)
short	short int，短整数，简写为short	2或4
long	long int，长整数，简写为long	4
	long double，高精度浮点数	12或16
signed	用来修饰char或任意整型（int、short和long），说明他们是有符号的整数（正整数、0、负整数）。一般缺省都是有符号的，所以这个修饰符通常省略	
unsigned	用来修饰char、int、short和long，说明他们总是正值或0	

数据表示范围

- 整型变量：分为有符号型与无符号型。
- 有符号型：
  - short 在内存中占两个字节，范围为-2^{15}~（$2^{15}-1$）
  - int 在内存中占四个字节，范围为-2^{31}~（$2^{31}-1$）
  - long在内存中占四个字节，范围为-2^{-31}~$2^{31}-1$
- 无符号型：最高位不表示符号位
  - unsigned short 在内存中占两个字节，范围为0~$2^{16}-1$
  - unsigned int 在内存中占四个字节，范围为0~$2^{32}-1$
  - unsigned long在内存中占四个字节，范围为0~$2^{32}-1$

超出取值范围会怎样？

- int的取值范围为-2147483648～2147483647
- 如果我们给他一个小于-2147483648或大于2147483647的数会如何呢？



变量的声明和初始化

```
char c = ‘\\’ ;

int i = 0;

long num = 0;

float eps = 1.0e-5;

char str[9] = “Love C++” ;
```

- 大家想一想：变量如果不赋初值，会如何？

类型转换

- 自动转换———计算机完成（隐式转换）
- 强制转换———程序员手工完成（显式转换）

```
int a;
double b = 9.80665;

a = b;                  //自动转换
a = (int)b * 4;         //强制转换
a = (int)(b * 4);       //强制转换
```

大家想一想：每一步返回a的值是多少？

类型转换

```
#include <iostream>
/*自动类型转换和强制类型转换演示*/

int main(){
  using namespace std;
  int f=5;

  cout<<f;
  cout<<f/2;
     cout << float(f/2);
  count<<float(f) / 2;

  return 0;
}
```



```
f                = 5
f / 2            = 2
(float)(f/2)     = 2.00
(float)f/2       = 2.50
```

自动类型转换

- 两个不同数据类型的运算结果，是**两种类型中取值范围更大的那种**
- 将取值范围小的类型转为取值范围大的类型是安全的，反之是不安全的
  - 如果大类型的值在小类型可以容纳的范围之内，则平安无事
  - 但是，浮点数转为整数，会丢失小数部分（去尾，非四舍五入）
  - 反之，转换后的结果必然是错误的，具体结果与机器和实现方式有关。尽量避免如此使用

强制类型转换

(类型)表达式
类型(表达式)

0	1	0	0	1	0	0	1
0	0	1	0	0	0	0	0
0	1	0	0	1	1	0	0
0	1	1	0	1	1	1	1
0	1	1	1	0	1	1	0
0	1	1	0	0	1	0	1
0	0	1	0	0	0	0	0
0	1	0	1	0	1	0	1

- 强转很灵活，你必须知道你在做什么！！！
- 经常用强转来解决很多warning.

打印华氏温度与摄氏温度对照表

```
#include <iostream>
/* 对fahr = 0, 20, ..., 300
 打印华氏温度与摄氏温度对照表 */
int main(){
    using namespace std;
    int  fahr, celsius;
    int  lower, upper, step;

    lower = 0;       // 温度表的下限
    upper = 300;     // 温度表的上限
    step  = 20;      // 步长
    fahr = lower;

    while (fahr <= upper) {
        celsius = 5* (fahr-32)/9;  //改成celsius = 5/9*
(fahr-32)；会如何？
        cout<<fahr<<'\t'<<celsius<<endl;
        fahr = fahr + step;
    }
   // system("PAUSE");
    return 0;
}}
```



```
0   -17
20    -6
40    4
60    15
80    26
100 37
120 48
140 60
160 71
180 82
200 93
220 104
240 115
260 126
280 137
300 148
```

更为精确的程序

```
//fc2.cpp

#include <iostream>
/* 对fahr = 0, 20, ..., 300
 打印华氏温度与摄氏温度对照表 */
int main(){
    using namespace std;
    float  fahr, celsius;
    int  lower, upper, step;

    lower = 0;       // 温度表的下限
    upper = 300;     // 温度表的上限
    step  = 20;      // 步长
    fahr = lower;

    while (fahr <= upper) {
        celsius = (5.0/9.0) * (fahr-32);
        cout<<fahr<<'\t'<<celsius<<endl;
        fahr = fahr + step;
    }
    return 0;
}
```

```
0   -17
20    -6
40    4
60    15
80    26
100 37
120 48
140 60
160 71
180 82
200 93
220 104
240 115
260 126
280 137
300 148
```

课后作业

- 继续完善计算器程序的设计，可计算整型、浮点型加减乘除，及字符加密解密，字符串输出等功能。
- 要求：代码风格友好

  程序界面友好

本章结束！

上一节课回顾

- **程序与硬件内存模型的映射**

  写出下列表达式的内存存储表示，假设起始地址都是0x58；

  int score = 89; char grade = ‘A’ ; char grade = “A”; char name[5] = “Harry” ;

- **基本概念学习：**

  变量、变量命名规则、变量赋初值、变量定义，（先定义后使用）

  常量、数常量、字符常量、转义字符、字符串、符合常量

  数据类型、数据类型修饰符、数据范围等

学习方法：

归纳法、思维导图式、练习

手册学习

简单输入输出练习