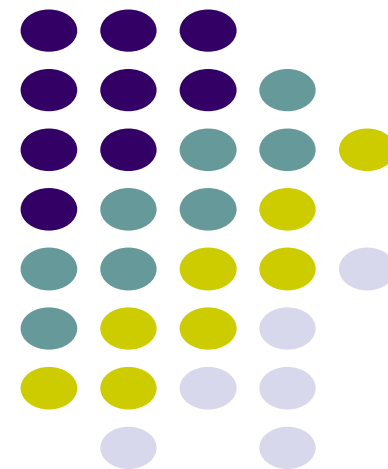


# 《计算机系统基础（四）：编程与调试实践》

## 真值与机器数



# 真值与机器数

整数的编码

浮点数的编码

# 整数的编码

---

带符号整数: char、short、int、long

无符号整数: unsigned

```
#include "stdio.h"
void main( )
{   int ai = 100, bi = 2147483648, ci = -100;
    unsigned au = 100, bu = 2147483648, cu = -100;
    printf("ai=%d, bi=%d, ci=%d \n", ai, bi, ci);
    printf("au=%u , bu=%u, cu=%u \n", au, bu, cu);
}
```

上述代码的执行结果是什么？

# 带符号整数和无符号整数的编码

分析：

真值			机器数	
十进制	二进制	十六进制	编码	说明
100	0110 0100	64H	00000064H	
2147483648		80000000H	80000000H	
-100	-0110 0100	-64H	FFFFFF9CH	补码

# 带符号整数和无符号整数的编码

分析:

真值			机器数	
十进制	二进制	十六进制	编码	说明
100	0110 0100	64H	00000064H	
2147483648		80000000H	80000000H	
-100	-0110 0100	-64H	FFFFFF9CH	补码

变量	类型	机器数	真值
bi	int	80000000H	-2147483648
cu	unsigned	FFFFFF9CH	4294967196

总结:

带符号整数: 补码

符号

数值

无符号整数: 二进制编码

数值

# 浮点数的编码

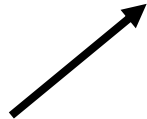
```
#include "stdio.h"
void main( )
{
    int ai = 100, bi = -100;
    float af = 100, bf = -100;
    printf("ai=%d, bi=%d \n", ai, bi );
    printf("af=%f , bf=%f \n", af, bf );
}
```

上述代码运行时，各变量的机器数分别是什么？

IA-32中的寄存器：定点寄存器组、浮点寄存器栈、多媒体扩展寄存器组

IA-32中指令类型：x86指令、x87浮点处理指令、MMX指令、SSE指令

IA-32中的浮点处理架构：x87 架构、SSE架构



# 真值与机器数

---

**总结：**

**带符号整数：采用补码的表示，如int类型**

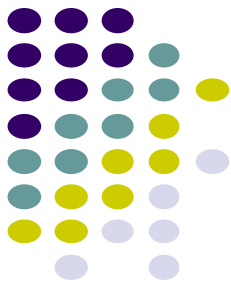
符号	数值
----	----

**无符号整数：二进制的表示，如unsigned int 类型**

数值
----

**浮点数：采用IEEE 754标准，如float类型**

符号	阶码	尾数
----	----	----



谢谢！