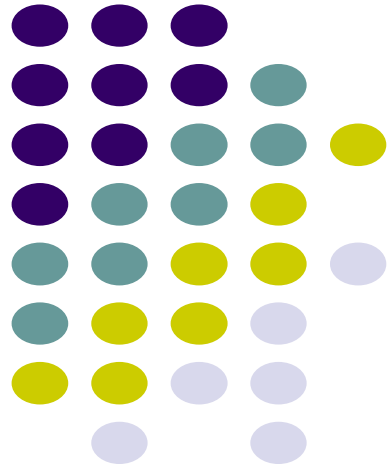
《计算机系统基础（四）：编程与调试实践》

数据类型的转换



数据类型的转换

整数之间的数据类型转换

整数和浮点数之间的转换

C语言中的自动类型转换

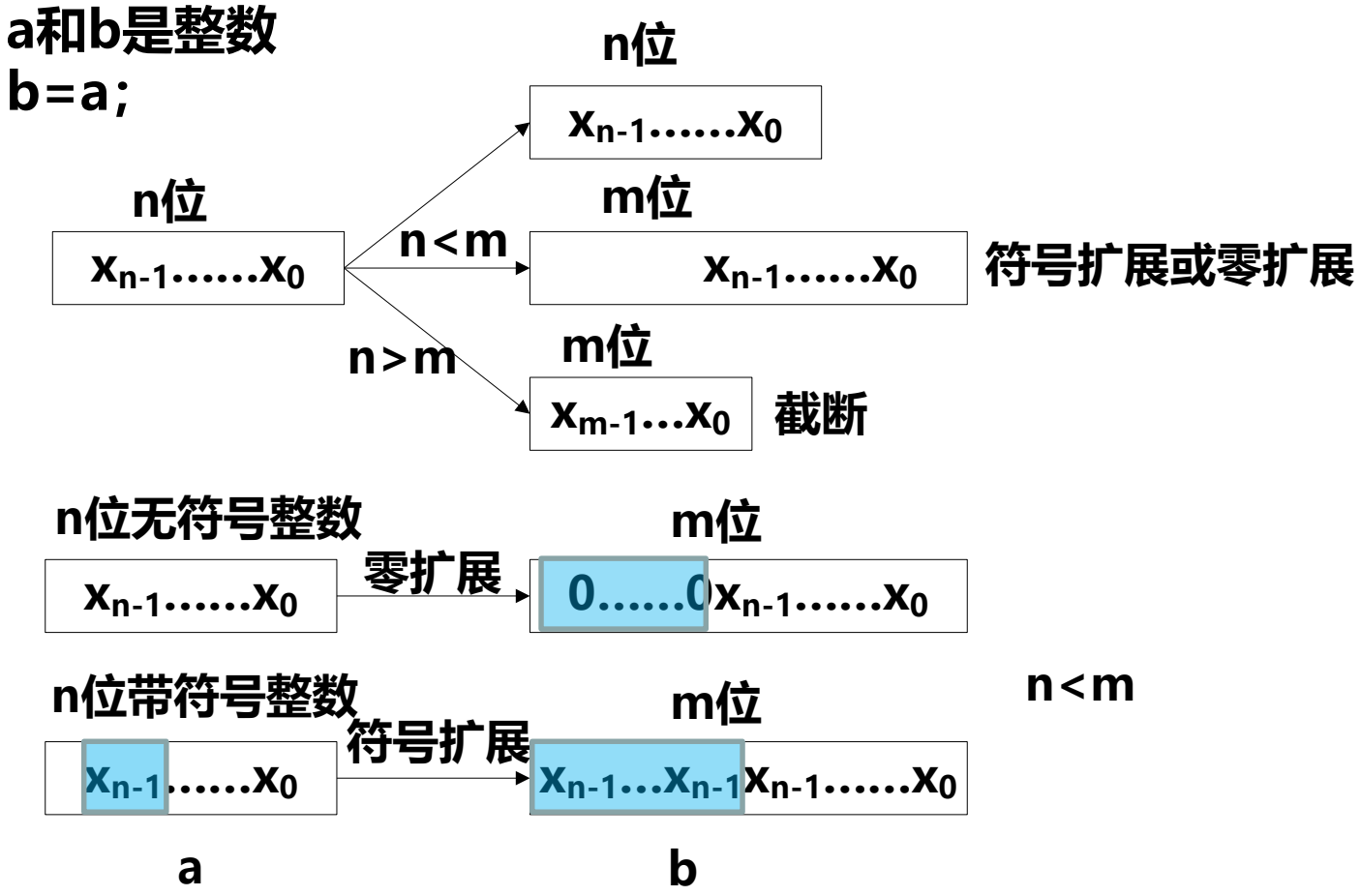
整数之间的数据类型转换

```
#include "stdio.h"
void main()
{   short           si=-100;
    unsigned short usi=si;
    int             i=usi;
    unsigned        ui=usi;
    int             i1=si;
    unsigned        ui1=si;
    int             i2=0x12348765;
    short           si2=i2;
    unsigned short usi2=i2;
    int             i3=si2;
    int             i4=4294967296;
    printf("si=%d,usi=%u,i=%d,ui=%u,i1=%d,ui1=%u\n",si,usi,i,ui,i1,ui1);
    printf("i2=%d,si2=%d,usi2=%u,i3=%d,i4=%d \n", i2,si2,usi2,i3,i4);
}
```

1. 这些赋值运算执行后，赋值运算左右两侧变量的值相等吗?
2. 程序运行过程中，各变量存储的机器数分别是什么?
3. 程序中i2赋值给si2，si2赋值给i3，i2和i3的值相等吗?
4. int型数据的范围是:
   -2147483648~2147483647
   i4的值是多少?

整数之间的数据类型转换

C语言中，整数的赋值不是在真值上的复制，而是在机器数上的赋值。



C语言中的“=”是赋值运算符，不同于数学上的等于符号“=”。

整数与浮点数之间的转换

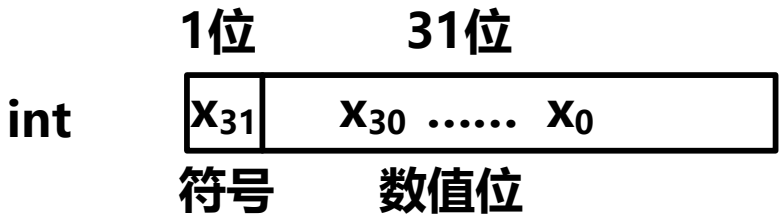
```
#include "stdio.h"
int main()
{   int      i1=0x7fffffff,        i2, itemp;
    float  f1=0x987654321, f2,ftemp;
    ftemp=i1;
    i2=ftemp;         //i2=(int)(float)i1;
    itemp=f1;
    f2=itemp;         //f2=(float)(int)f1;
    printf("i1=%d,i2=%d,f1=%f,f2=%f\n", i1,i2,f1,f2);
}
```

1. 代码运行过程中，各变量存储的机器数分别是什么?
2. i1和i2的值相同吗？为什么?
3. f1和f2的值相同吗？为什么?

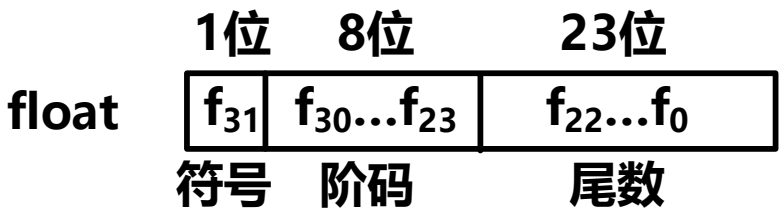
整数与浮点数之间的转换

整数与浮点数之间的转换，是在编码上的转换。

带符号整数：补码



浮点数：float、double IEEE 754标准



整数与浮点数之间的转换

i1=0x7fff ffff; ftemp=i1; i2=ftemp ;

i1: 0x7fff ffff 补码

↓ = 0111 1111 1111 1111 1111 1111 1111 1111B

1.11 1111 1111 1111 1111 1111 1111 1111×2^{30} 真值

+1 （尾数入操作）

↓ ≈10.0×2^{30}=1.0×2^{31}

ftemp : 0 1001 1110 000 0000 0000 0000 0000 0000B float

↓ 31+127=128+16+8+4+2

1.0×2^{31} 真值

↓ =1000 0000 0000 0000 0000 0000 0000 0000B

i2: 1000 0000 0000 0000 0000 0000 0000 0000B 补码

=0x8000 0000

补码→float编码→补码，整数与浮点之间的转换不是机器数上的复制，而是编码上的转化。在int→float转换中，可能会有精度的损失。

整数与浮点数之间的转换

```
#include "stdio.h"
int main()
{   int      i1=0x7fffffff,        i2, itemp;
    float  f1=0x987654321, f2,ftemp;
    ftemp=i1;
    i2=ftemp;         //i2=(int)(float)i1;
    itemp=f1;
    f2=itemp;         //f2=(float)(int)f1;
    printf("i1=%d,i2=%d,f1=%f,f2=%f\n", i1,i2,f1,f2);
}
```

1. 代码运行过程中，各变量存储的机器数分别是什么?
2. i1和i2的值相同吗？为什么?
3. f1和f2的值相同吗？为什么?

整数与浮点数之间的转换

f1=0x987654321；itemp=f1; f2=itemp;

0x987654321=1001 1000 0111 0110 0101 0100 0011 0010 0001B

=1.001 1000 0111 0110 0101 0100 0011 0010 0001×2^{35}

f1: 0 1010 0010 001 1000 0111 0110 0101 0100 float

=0x51187654 35+127=128+32+2

1.001 1000 0111 0110 0101 0100×2^{35} 真值

=1001 1000 0111 0110 0101 0100 0000 0000 0000B

itemp: 1000 0000 0000 0000 0000 0000 0000 0000B 补码

-111 1111 1111 1111 1111 1111 1111 1111B

+1

-1000 0000 0000 0000 0000 0000 0000 0000B 真值

=-1.0×2^{31}

f2: 1 1001 1110 000 0000 0000 0000 0000 0000B float

=0xcf000000 31+127=128+16+8+4+2

float编码→补码→float编码，整数与浮点之间的转换不是机器数上的复制，而是编码上的转化。在float→int转换中，可能会有溢出问题。

整数与浮点数之间的转换

总结：

1. **整数与浮点数转换时，是在编码格式上的转换。**
2. **在int ↔ float转换中，可能会有精度损失、溢出、小数丢弃等问题导致的数据不一致。**
3. **不同编译系统采用的编译优化有差异，同一程序在不同系统上运行，得到的结果可能不一样。**

C语言中的自动类型转换

已知 f(n) = $\sum_{i=0}^{n} 2^i = 2^{n+1}-1 = \overbrace{11\cdots\cdots1}^{n+1}$ B，计算 f(n)的C语言函数f1如下。

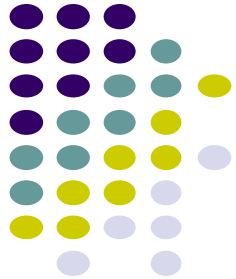
```
int f1( unsigned int n )
 {  int sum = 1, power = 1;
    int i;
    for (  i = 0; i <= n - 1; i ++ )
    {   power *= 2;
        sum += power;
     }
    return sum;
 }
```

1. 执行f1(0)时，为什么会出现死循环?
2. 为了得到正确的值，应该如何修改函数f1?

数据类型的转换

总结：

1. **整数与整数之间的转换是在机器数上的复制**
2. **整数与浮点数之间的转换是在编码上的转换**
3. **一个运算表达式中有不同数据类型时，C语言会自动进行类型转换**



谢谢！