C语言中数据的隐式转换和显式转换

1. 不同数据类型间的赋值

1.1 int 转换成 char

Int 转换成 char 时,直接获取 int 的低 8bit 值。 0x12345678 直接获取低位 0x78 。

```
| Interpretation of the print of the print
```

Int 转换成 char, 低 8bit 的最高位直接作为符号位。 0x123456A8 直接获取低位 0xA8 。直接作为负数处理。

Printf(); 在打印时会自动把负数扩充为 32bit。

```
13 ⊟int _tmain(int argc, _TCHAR* argv[])
14
15
        cout << end1:
16
                   数据的隐式转换和显式转换"<<endl;
        cout << "
17
18
        int nOrg = 0x123456A8;
19
20
        char cH = (char)nOrg ;
21
        printf("\r\n cH Value : %d; Hex Value : 0x%x; Sizeof Value %d", cH, cH, sizeof (cH));
22
23
                      C:\Users\Ensense\Documents\Visual Studio 2010\Projects\test\Debug\test.exe
24
        int a ;
25
        cin>>a ;
                           数据的隐式转换和显式转换
26
        return 0;
27 }
                        cH Value :-88 ; Hex Value :0xffffffa8 ; Sizeof Value 1
28
```

调用 sizeof 计算出 cH 的长度是 1 个字节。可以佐证上面的 0xffffffa8 是 printf();自动扩充的。

直接把 cH 赋值为-1 , printf();打印出来的也是 0xffffffff,更是证明了这里是 prinf();问题。不 去多考虑了。

PS:应该是 printf()把所有数据都扩充到 32bit,正数的高位是 0,所以没显示,负数是 1,就都打印出来了,感兴趣的自行研究,我对 printf()知之甚少,不做深究。

1.2 int 转换成 unsigned char

```
□int _tmain(int argc, _TCHAR* argv[])
{
    cout<<endl;
    cout<</f>
    数据的隐式转换和显式转换"<<endl;

    int nOrg = 0x12345678;

    unsigned char cH = nOrg;
    printf("\r\n cH Value :%d; Hex Value :0x%x", cH, cH);

    C:\Users\Ensense\Documents\Visual Studio;

数据的隐式转换和显式转换
    cH Value :120; Hex Value :0x78
```

Int 转换成 unsigned char 时,直接获取 int 的低 8bit 值。0x12345678 直接获取低位 0x78 。

Int 转换成 unsigned char 时,直接获取 int 的低 8bit 值。0x12345678 直接获取低位 0x78 。 unsigned char 型没有符号位,都是正数。

1.3 char 转换成 int

```
int _tmain(int argc, _TCHAR* argv[])
4
5
        cout << end1;
6
7
                 数据的隐式转换和显式转换"<<endl;
       cout<<"
8
       char cH = 78;
9
       int nOrg = cH;
!1
!2
!3
       printf("\r\n cH Value :%d : Hex Value :0x%x : Sizeof Value %d" ,nOrg,nOrg,sizeof(nOrg));
4
                                                                              _ O X
   C:\Users\Ensense\Documents\Visual Studio 2010\Projects\test\Debug\test.exe
:5
:6
        数据的隐式转换和显式转换
!7
!8
9
    cH Value :78 ; Hex Value :0x4e ; Sizeof Value 4
}_⊟int _tmain(int argc, _TCHAR* argv[])
      cout << end1;
                 数据的隐式转换和显式转换"<<endl;
      char cH = -35;
      int nOrg = cH;
      printf("\r\n cH Value :%d : Hex Value :0x%x : Sizeof Value %d" ,nOrg,nOrg,sizeof(nOrg));
                                                                              _ 0 X
    C:\Users\Ensense\Documents\Visual Studio 2010\Projects\test\Debug\test.exe
        数据的隐式转换和显式转换
     cH Value :-35 ; Hex Value :0xffffffdd ; Sizeof Value 4
```

char 转换成 int 时,直接在高位补充符号位。数据大小本身不变。

1.4 char 转换成 unsigned int

```
int _tmain(int argc, _TCHAR* argv[])
{
    cout<<endl;
    cout<<" 数据的隐式转换和显式转换"<<endl;

    char cH = -78;

    unsigned short nOrg = cH;

    printf("\r\n cH Value :%d; Hex Value :0x%x; Sizeof Value %d", nOrg, nOrg, sizeof(nOrg));

    C:\Users\Ensense\Documents\Visual Studio 2010\Projects\test\Debug\test.exe

    数据的隐式转换和显式转换
    cH Value :65458; Hex Value :0xffb2; Sizeof Value 2_
```

负数这里我改成 unsigned short 了,用来避免 printf();溢出什么的。但是从结果还是能看出来:

char 在扩充成 unsigned int 时,高位补充的是符号位。

```
3 _= int _tmain(int argc, _TCHAR* argv[])
4
5
       cout << end1;
6
7
8
       cout<<"
                 数据的隐式转换和显式转换"<<endl;
       unsigned char cH = 0xA8 ;
9
       unsigned short nOrg = cH;
2
       printf("\r\n cH Value :%d : Hex Value :0x%x : Sizeof Value %d" ,nOrg,nOrg,sizeof(nOrg));
                                                                                - - X
    C:\Users\Ensense\Documents\Visual Studio 2010\Projects\test\Debug\test.exe
5
6
7
        数据的隐式转换和显式转换
8
9
     cH Value :168 ; Hex Value :0xa8 ; Sizeof Value 2
```

unsigned char 扩充成 int 时,高位补充的是 0;

1.5 总结

对于多字节的数转换成少字节的数时,直接取多字节数的低位相应字节。如果转换后的 数据是有符号的,最高位作为符号位。

对于少字节的数扩充到多字节的数。如果少字节的数有符号,则扩充后的数高位扩充符号位,如果少字节的数无符号,则高位字节扩充 0.

2. 计算中的数据转换

2.1 先看现象

```
13 ∃ int _tmain(int argc, _TCHAR* argv[])
   14
   15
           cout << end1;
           cout<<
                     数据的隐式转换和显式转换"<<endl;
   16
   17
   18
           unsigned char cH = 50;
                                             C:\Users\Ensense\Documents\Visual St
   19
   20
           unsigned short nOrg = cH;
   21
                                                 数据的隐式转换和显式转换
   22
           float s ;
   23
                                              cH Value :16.000000
           s = cH/3;
   24
   25
           printf("\r\n cH Value :%f " ,s);
   26
   27
   28
   29
           int a ;
   30
           cin>>a ;
   31
           return 0;
   32 }
   33
   34
   35 □void zzz(void)
   36
       {
   37
0 % + 4
误列表
🕽 0 个错误 📗 🚹 1 个警告 📗 🚺 0 个消息
                           文件
1 warning C4244: "=" :
                          test.cpp
   从 "int" 转换到 "float" , 可
   能丢失数据
```

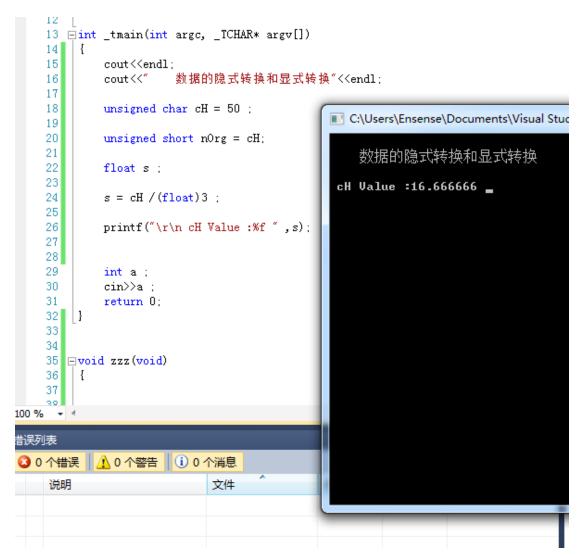
C 语言在只有整形数据参与计算时,默认把数据扩充到 int,然后计算,计算结果默认为 int 。所有 50/3 = 16 。 16 再转换成 float 型,是 16.000000 。 出现计算错误和数据精度丢失警告。

```
12
    13 ☐ int _tmain(int argc, _TCHAR* argv[])
    14 {
    15
           cout << endl;
                    数据的隐式转换和显式转换"<<endl;
    16
           cout<<
    17
           unsigned char cH = 50;
    18
                                           C:\Users\Ensense\Documents\Visual Stuc
    19
    20
           unsigned short nOrg = cH;
    21
                                               数据的隐式转换和显式转换
    22
           float s ;
    23
                                            cH Value :16.000000
    24
           s = (float)(cH/3);
    25
           printf("\r\n cH Value :%f " ,s);
    26
    27
    28
    29
           int a ;
           cin>>a ;
    30
    31
           return 0;
    32 }
    33
    34
    35 ⊟void zzz(void)
    36
       {
    37
    30
100 % 🕶 🖪
错误列表
说明
                          文件
```

强行把计算结果显式的转换成 float 型,虽然警告消失了,但在转换发生在计算之后, 计算结果已经是 16 了,再转换还是 16.000000 。

```
13 ∃int _tmain(int argc, _TCHAR* argv[])
  14
  15
           cout << endl;
  16
           cout<<
                     数据的隐式转换和显式转换"<<endl;
  17
  18
           unsigned char cH = 50 ;
                                            C:\Users\Ensense\Documents\Visual $
  19
   20
           unsigned short nOrg = cH;
                                                数据的隐式转换和显式转换
  21
  22
           float s ;
  23
                                            cH Value :16.666666
  24
           s = cH/3.0;
  25
           printf("\r\n cH Value :%f " ,s);
  26
  27
  28
  29
           int a ;
   30
           cin>>a ;
  31
           return 0;
  32
      | }
  33
  34
  35 ⊟void zzz(void)
  36 {
  37
0 %
吳列表
🕽 0 个错误 📗 🗘 1 个警告 📗 🚺 0 个消息
   说明
                          文件
1 warning C4244: "=":
                          test.cpp
  从 "double" 转换
  到 "float" ,可能丢失数据
```

C 语言中直接定义的小数默认保存为 double 。 有 double 型数据参与计算时,所有数据先扩充为 double 型,再计算结果。计算结果保存为 double 型。Double 型数据可以保存小数,所以结果精度未丢失,但是 double 和 float 类型不符,会警告。



当计算中有 float 型数据参与时,会自动先把所有数据扩充到 float 型,在计算结果,结果保存为 float 型。

上图显式的把 3 转换成 float 型。最后结果正确,无警告。

```
13 ☐ int _tmain(int argc, _TCHAR* argv[])
    14
    15
             \verb"cout! <\!\!< \verb"end1";
                      数据的隐式转换和显式转换"<<endl;
    16
             cout<<
    17
    18
            unsigned char cH = 50;
    19
                                                 C:\Users\Ensense\Documents\Visual
    20
            unsigned short nOrg = cH;
    21
                                                      数据的隐式转换和显式转换
    22
            float s ;
    23
                                                  cH Value :16.666666 _
            s = (float) cH / 3;
    24
    25
            printf("\r\n cH Value :%f " ,s);
    26
    27
    28
    29
             int a ;
    30
             cin>>a ;
    31
             return 0;
    32 }
    33
    34
    35 ⊟void zzz(void)
    36 {
    37
100 % 🕶 🖪
错误列表
② 0 个错误 ▮ 1 0 个警告 ▮ 1 0 个消息
     说明
                             文件
                                             行
```

当计算中有 float 型数据参与时,会自动先把所有数据扩充到 float 型,在计算结果,结果保存为 float 型。

上图显式的把 cH 转换成 float 型。最后结果正确,无警告。

```
12 |
 13 ⊟int _tmain(int argc, _TCHAR* argv[])
 14
          \texttt{cout} \leq \texttt{end1};
 15
                     数据的隐式转换和显式转换"<<endl;
 16
          cout<<
 17
 18
          unsigned char cH = 50 ;
                                              C:\Users\Ensense\Documents\Visual S
 19
          unsigned short nOrg = cH;
 20
 21
                                                  数据的隐式转换和显式转换
 22
          float s :
 23
                                              cH Value :16.666666
          s = (float)cH / (double)3;
 24
 25
          printf("\r\n cH Value :%f " ,s);
 26
 27
 28
 29
          int a ;
          cin>>a ;
 30
 31
          return 0;
 32
     | }
 33
 34
 35 ⊟void zzz(void)
 36
      {
 37
%
列表
0 个错误 📗 🚹 1 个警告 📗 🛈 0 个消息
  说明
                           文件
 warning C4244: "=" :
                          test.cpp
 从 "double" 转换
 到"float",可能丢失数据
```

当同时有 float 和 double 型数据参与计算时,数据会自动扩充到 double 再计算。结果为 double 型。所以有警告。

2.2 总结

C语言在做计算时,对于所有整形数 (int 和 char) 都扩充到 int 型然后计算,此举是为了防止数据溢出。

在 C 语言中直接定义的小数数据类型(比如 3.0)是 double 型。

有 double 型参与计算时,会把其他数也扩充为 double。

没有 double 类型,有 float 型数据参与计算时,会把所有数据扩充为 float。