2020/04/28 第19课 结构体、共用体、枚举的使用

笔记本: 0

创建时间: 2020/4/28 星期二 15:12

作者: ileemi

标签: 共用体, 结构体, 枚举

结构体

• <u>union</u>

• enum

结构体

同类型结构体可以互相赋值

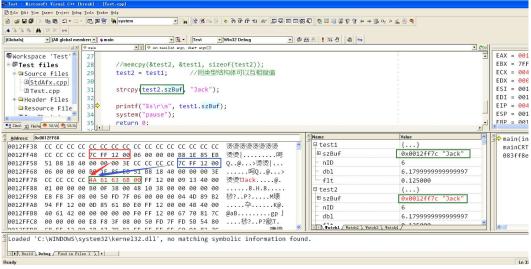
```
struct tagTest
   char *szBuf;
   int nID;
   double dbl;
   float flt;
int main(int argc, char* argv[])
   struct tagTest test1 = {
       "Tom",
       6. 18,
   struct tagTest test2;
    test2 = test1;
   system("pause");
```

```
▼ < Test
                                        ▼ Win32 Debug
                                                      int na
®Workspace 'Test'
               24
               25
                     struct tagTest test2;
■ Test files
               26
 ⊕ ⊜Source Files
   ≝StdAfx.cpp
               27
                     //memcpy(&test2, &test1, sizeof(test2));
test2 = test1; //同类型结构体可以互相赋值
               28
   ∄Test.cpp
               29
                     system("pause");
 ■ 

Header Files
               30
31
                     return 0;
  □Resource File
                  }
               32
Classi FileVie VAVii 🧌 VA Ou
Address: 0x0012FF68
                                                          ⊞ test1
{...}
                                                          ⊞ test2
烫烫烫烫烫烫烫烫烫烫
烫烫烫烫烫烫烫烫烫烫
烫烫烫烫 ,'B....
..<mark>呺Q..@...>.0</mark>烫
,'B......呺Q..@
...>.0烫....@.
0012FFA8 94 FF 12 00 0D 85 61 80 E0 FF 12 00 00 4B 40 00
                                          .....卆.....K@.
                                                          ▼ ▼atch1 / Watch2 / Watch3 / Watch4 /
Loaded 'C:\WINDOWS\system32\kernel32.dll', no matching symbolic information found.
Build Debug Find in Files 1
  #include <stdio.h>
  #include <string.h>
  #include <stdlib.h>
  struct tagTest
      char *szBuf;
      int nID;
      double db1;
      float flt;
  int main(int argc, char* argv[])
      char szBuf[] = "Tom";
      struct tagTest test1 = {
           szBuf,
          6. 18,
           0.125f
      struct tagTest test2;
       test2 = test1;
                          //同类型结构体可以互相赋值
       修改结构体test2中的成员szBuf后,test1中的成员szBuf也会受影响
       两者指向的是同一个地方
      strcpy(test2.szBuf, "Jack");
```

```
printf("%s\r\n", test1.szBuf); //Jack
system("pause");
return 0;
}

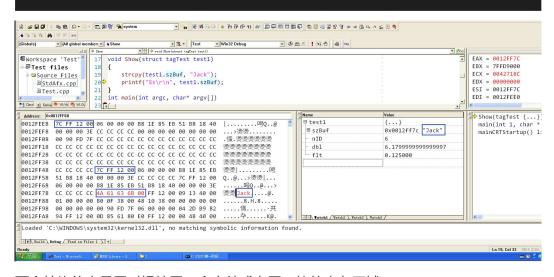
Zest = Normonoft Varial CV. (breal) = (Test.spp)
| B | Dia Rut Pojet | Rody | Feb | End | End | Ed)
```



函数参数的传递,无论任何类型都是从实参->形参,进行传递的

函数参数的传递,传递指针,传递的就是指针的一份拷贝,传递结构体,就是结构体的一份拷贝,传递其它类型的数据,就是其它类型的一份拷贝。 当结构体类型做函数参数传递的时候:

void Show(struct tagTest test1)
{
 strcpy(test1.szBuf, "Jack");
 printf("%s\r\n", test1.szBuf);



两个结构体变量同时操控同一个文件或者同一块的内存区域

```
struct tagTest test2;
//memcpy(&test2, &test1, sizeof(test2));
```

```
/*
    如果存在指针成员,这时候赋值了指针的值,而不是指针目标的值(浅拷
贝)
    */
    test2 = test1;
```

避免两个结构体变量中的指针成员同时引用同一个资源(一个指针成员修改指向数据内容,另一个指针也会受到影响)的方法:

- 1、有条件的情况,可以让每个结构体变量中的指针成员各自拥有独立的资源。**深拷贝**(好比篮球场上大伙抢一个篮球,一个人把篮球搞坏后,其他人不能玩,如果有条件下每个人都有一个篮球,就不会收到改影响),将结构体成员指针更改为数组。
- 2、引用计数,所有结构体变量中的指针成员同时引用同一个资源,但是需要记录引用的情况,满足合理条件则真正处理改资源。

微软系统大量使用 引用计数方法,很多情况下没有空间条件

好比Windows下的可执行文件装载到内存创建进程,可执行文件需要引用

kernel32.dll(Windows中非常重要的32位动态链接库文件,属于内核级文件), uese32.dll gdi32.dll **引用**kernel32.dll

```
kernel32.dll ->(引用) ntdll.dll
uese32.dll ->(引用) kernel32.dll
gdi32.dll ->(引用) kernel32.dll
```

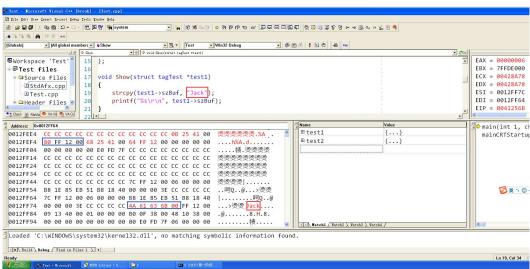
将函数的参数设置为结构体指针,可以解决浅拷贝带来的问题,间接访问,直接修改 实参

```
#include <stdio.h>
#include <stdib.h>
#include <stdlib.h>

struct tagTest
{
    char *szBuf;
    int nID;
    double dbl;
    float flt;
};

void Show(struct tagTest *test1) //传递时复制的是指针(4字节)
{
    /*
    问接访问,直接修改实参
    -> 符号,有问接访问,导致直接访问并修改实参
    */
    strcpy(test1->szBuf, "Jack");
    printf("%s\r\n", test1->szBuf);
}
```

```
int main(int argc, char* argv[])
   char szBuf[] = "Tom";
   struct tagTest test1 = {
      szBuf,
      6,
      6. 18,
      0.125f
   struct tagTest test2;
   test2 = test1; //同类型结构体可以互相赋值
   修改结构体test2中的成员szBuf后, test1中的成员szBuf也会受影响
   Show(&test1);
   printf("%s\r\n", test1.szBuf);
   system("pause");
```



给函数传递一个指向结构体的指针,远比传递一个结构体要小的多,将其压到堆栈上的效率会提高很多,传递指针在访问结构体成员的时候,必须使用间接访问来访问,结构体越大,把指向它的指针传递给函数的效率就越高。

函数的参数参数为数组名时,传递的是数组的首地址 函数的参数参数为结构体时,传递的是结构体中所有成员的引用 类型定义不占用空间,不产生编译器行为,写给编译器看的

产生变量的时候,产生编译器行为,编译器为其分配空间

结构体的嵌套,当子结构体仅为主结构服务时,可将其至于主结构内,也可将其设为 私有。

```
#include "stdafx.h"
#include <string.h>
struct tagTest
   char *szBuf;
   int nID;
   struct tagDateOfBirth
       int nYear;
       int nMonth;
        int nDay;
   }Dob;
    double dbl;
    float flt;
int main(int argc, char* argv[])
    struct tagTest test1 = {
    struct tagTest :: tagDateOfBirth dob = {
        2020, 4, 28
    test1.Dob.nMonth = 5;
    dob. nMonth = 5;
    dob. nDay = 1;
    printf("%d %d %d\r\n", dob.nYear, dob.nMonth, dob.nDay);
    system("pause");
```

为了分类管理程序,可以使用结构体嵌套,调用对用的成员结构时,需要注意层级关系,可使用::。

```
struct tagScore
   int nType; //标记
   char cLevel;
   float fPoint;
   int nPoint;
   char szText[16];
void ShowScore(struct tagScore *pScore)
   switch (pScore->nType)
   case TY_CHAR:
       printf("%c\r\n", pScore->cLevel);
   case TY_FLOAT:
       printf("%f\r\n", pScore->fPoint);
   case TY_INT:
       printf("%d\r\n", pScore->nPoint);
   case TY_TEXT:
       printf("%s\r\n", pScore->szBuf);
int main(int argc, char* argv[])
   struct tagScore s1;
   s1.nType = TY_CHAR;
   s1.cLevel = 'A';
   ShowScore(&s1);
   struct tagScore s2;
   s2.nType = TY_FLOAT;
```

```
s2. fPoint = 88.5f;
                                                          ShowScore (&s2);
                                                          struct tagScore s3;
                                                          s3.nType = TY_INT;
                                                          s3. nPoint = 77;
                                                         ShowScore(&s3);
                                                         struct tagScore s4;
                                                          s4.nType = TY\_TEXT;
                                                          strcpy(s4. szBuf, "Hahaha");
                                                          ShowScore (&s4);
                                                          system("pause");
                                                          return 0;
  A lext2 - Licrosoft Visual C++ (bresk) - (lext2.cpp)
Dile Lit Liw Jacort Brijert Lebog Jacks Linder Kolp
Dile Lit Liw Jacort Brijert Linder Jacks Linder Kolp
Dile Lit Live Jacort Dile Dile Linder Kolp
Dile Lit Linder Linder Linder Linder Kolp
Dile Lit Linder Li
                                                                                                                                                                                                          (Globals) ■ (All global member
| Closed | Close | Clo
                                                                                                                                                    Test2 yWin32 Ockup y 多區兰 ! 知 ① 备 ing
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         EAX = 00000008
EBX = 7FFDE000
ECX = 0042AA80
EDX = 0042AA80
ESI = 00000000
EDI = 0012FF80
EIP = 0040118D
                                                                                                                                    struct tagScore s4;
s4.nType = TY_TEXT;
strcpy(s4.szBuf, "Hahaha");
ShowScore(&s4);
                                                                                                                                                                                                                                                                            CC 表示内存中暂时未使用的空间,内存空间存在浪费情况
 main(int 1, char * * 0x0
mainCRTStartup() line 20
KERNEL32! 7c817067()
                                                                                                                                                                                                                                                                                                                                                                               = s2
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       ( ....)
                                                                                                                                                                                                                                                                                                                                                                                   ■ s2

- nType

- cLevel

- fPoint

- nPoint

⊞ szBuf
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         1
-52 '?
88.5000
-858993460
0x0012ff50 "烫烫烫烫烫
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        ❷英;◎♦團蟲♥〓
                                                                                                                                                                                                                                                                                                                                                                                 ⊞ s3
  Loaded 'C:\WINDOWS\system32\kernel32.dll', no matching symbolic information found
      Build Debug Find in Files 1 \ 4
```

Version 2:

```
char szText[16];
void ShowScore(struct tagScore *pScore)
   switch (pScore->nType)
   case TY_CHAR:
        printf("%c\r\n", *(char *)pScore->szText);
   case TY_FLOAT:
        printf("%f\r\n", *(float *)pScore->szText);
   case TY_INT:
       printf("%d\r\n", *(int *)pScore->szText);
   case TY_TEXT:
       printf("%s\r\n", pScore->szText);
int main(int argc, char* argv[])
   struct tagScore s1;
   s1.nType = TY\_CHAR;
   *(char *)s1.szText = 'A';
   ShowScore(&s1);
   struct tagScore s2;
   s2. nType = TY_FLOAT;
   *(float *) s2. szText = 88.5f;
   ShowScore (&s2);
   struct tagScore s3;
   s3.nType = TY_INT;
   *(int *)s3. szText = 77;
   ShowScore(&s3);
   struct tagScore s4;
   s4.nType = TY_TEXT;
   strcpy(s4. szText, "Hahaha");
   ShowScore(&s4);
   system("pause");
```

Version 2.1:

```
#include <string.h>
struct tagScore
    int nType; //标记
   char szText[16];
void ShowScore(struct tagScore *pScore)
   switch (pScore->nType)
   case TY_CHAR:
       printf("%c\r\n", *(char *)pScore->szText);
   case TY_FLOAT:
       printf("%f\r\n", *(float *)pScore->szText);
   case TY_INT:
        printf("%d\r\n", *(int *)pScore->szText);
   case TY_TEXT:
       printf("%s\r\n", pScore->szText);
int main(int argc, char* argv[])
   struct tagScore s1;
   s1.nType = TY CHAR;
   *(char *)s1.szText = 'A';
   ShowScore(&s1);
   s1.nType = TY_FLOAT;
```

```
*(float *) s1. szText = 88.5f;
                              ShowScore (&s1);
                               s1.nType = TY INT;
                              *(int *)s1.szText = 77;
                              ShowScore (&s1);
                               s1.nType = TY\_TEXT;
                               strcpy(s1.szText, "Hahaha");
                               ShowScore (&s1);
                               system("pause");
                               return 0;
🚵 😅 🖬 🗗 % 🗞 🙉 🕰 🗨 - 🖪 🗩 👺 🦬 system
                                                                                                                                        浅龙 | 4
                                                                                                                                 Test2
| Mal global members | wms | w
                                (All global members 🕶 🖣 main
                                                                                                                                                                          ▼ Win32 Debug
                                                                                                                                                                                                                                   - ②曲玉!140 | 4 | №
                                                                                                        int main(int argo, char* argv[
                                                                                         s1.nType = TY_FLOAT;
*(float *)s1.szText = 88.5f;
ShowScore(&s1);
                                                                                        s1.nType = TY_INT;
*(int *)s1.szText = 77;
ShowScore(&s1);
CXX0017: Error: symbol
                                                                                                                                                                                                                                                             s2
                                                                                                                                                                                                                                                              s3
                                                                                                                                                                                                                                                                                                                                CXX0017: Error: symbol
                                                                                                                                                                                                                                                              s4
                                                                                                                                                                                                                                                                                                                                CXX0017: Error: symbol
  0012FFFC 00 00 00 00 41 63 74 78 20 00 00 00 01 00 00 00 ....Actx ......
                                                                                                                                                                                                                                                     ▼atch1 / Watch2 / Vatch3 / Watch4 /
Loaded 'C:\WINDOWS\system32\kernel32.dll', no matching symbolic information found.
```

union

4 M Build Debug Find in Files 1

C语言提供一种数据存储类型: 语法糖 union

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define TY_CHAR 0
#define TY_FLOAT 1
#define TY_INT 2
#define TY_TEXT 3

//语法糖
//选择其所有成员中块头最大的类型字节数,作为其实际的占用空间
union unTest
```

```
float flt;
                                                    double dbl;
                    int main(int argc, char *argv[])
                                                  union unTest untest;
                                                     编译器眼中,取共用体类型的地址将其解释为当前成员类型的指针,取内容
                                                    *(char *)&untest = 'A';
                                                    untest. n = 999;
                                                  *(int *) & untest = 999;
                                                  untest. flt = 6.18f;
                                                  *(float *) &untest = 6.18f;
                                                  untest. db1 = 5.20;
                                                  *(double *)&untest = 5.20f;
                                                    system("pause");
   | Mind class members - Create CRC+ Memb | No | Test2 | WMn32 Debag | No | Wmn32 Debag | Wmn32 Debag
  本海海崎県民民会
unTest MAI class members Mo men
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              ### Mame | Washington | Wilderson | Wilderson | Wilderson | Washington | Washington | Wilderson | Washington | Washington | Wilderson | Washington | Washington | Washington | Washington | Wilderson | Washington |
```

使用共用体对上面的程序再加以修改:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define TY_CHAR 0
```

```
struct tagScore
   int nType; //标记
       char cLevel;
       float fPoint;
       int nPoint;
       char szText[16];
   }uscore;
void ShowScore(struct tagScore *pScore)
   switch (pScore->nType)
   case TY_CHAR:
       printf("%c\r\n", pScore->uscore.cLevel);
   case TY FLOAT:
       printf("%f\r\n", pScore->uscore.fPoint);
   case TY_INT:
       printf("%d\r\n", pScore->uscore.nPoint);
   case TY_TEXT:
       printf("%s\r\n", pScore->uscore.szText);
int main(int argc, char *argv[])
   struct tagScore s1;
   s1.nType = TY_CHAR;
   s1. uscore. cLevel = 'A';
   ShowScore(&s1);
   s1.nType = TY_FLOAT;
   s1.uscore.fPoint = 88.5f;
   ShowScore(&s1);
```

```
s1.nType = TY_INT;
s1.uscore.nPoint = 77;
ShowScore(&s1);

s1.nType = TY_TEXT;
strcpy(s1.uscore.szText, "Hahaha");
ShowScore(&s1);

system("pause");
return 0;
}
```

enum

仅仅只能代替符号化**整数的宏**,符号化整型常量会给其成员(枚举常量)安排一个唯一的整数值可以说是一个功能单一,受控制的宏。

在定义枚举变量时,它的值可以赋值,在不进行强转的情况下只能赋值枚举常量值, 编译器会做类型检查

```
enum eType
{

TY_CHAR,

TY_FLOAT,

TY_INT,

TY_TEXT
};

enum eType etype;

etype = 0; //编译不通过

etype = TY_CHAR;

...
```

使用枚举对上面的例子再次进行改进:

```
struct tagScore
    int nType; //标记
       char cLevel;
       float fPoint;
       int nPoint;
       char szText[16];
    }uscore;
       TY_CHAR,
       TY_FLOAT,
       TY_INT,
       TY_TEXT
   }etype;
//输出数据
void ShowScore(struct tagScore *pScore)
   switch (pScore->nType)
   case tagScore::TY_CHAR:
       printf("%c\r\n", pScore->uscore.cLevel);
   case tagScore::TY_FLOAT:
        printf("%f\r\n", pScore->uscore.fPoint);
   case tagScore::TY_INT:
       printf("%d\r\n", pScore->uscore.nPoint);
   case tagScore::TY_TEXT:
       printf("%s\r\n", pScore->uscore.szText);
int main(int argc, char *argv[])
   struct tagScore s1;
   s1.nType = tagScore::TY_CHAR;
   s1. uscore. cLevel = 'A';
```

```
ShowScore(&s1);

s1.nType = tagScore::TY_FLOAT;
s1.uscore.fPoint = 88.5f;
ShowScore(&s1);

s1.nType = tagScore::TY_INT;
s1.uscore.nPoint = 77;
ShowScore(&s1);

s1.nType = tagScore::TY_TEXT;
strcpy(s1.uscore.szText, "Hahaha");
ShowScore(&s1);

system("pause");
return 0;
}
```

变体原理: 一个vt 控制类型,一个共用体控制解释方式,上层封装variant_t

tagVARIANT ComVariant variant_t VT_I2