

示例代码：

```
1 // 4. 单重虚继承有虚函数.cpp : 定义控制台应用程序的入口点。
2 //
3
4 #include "stdafx.h"
5 #include <iostream>
6 using namespace std;
7
8 // 家具类
9 class CFurniture {
10 public:
11     CFurniture() {
12         printf("CFurniture::CFurniture()\r\n");
13         m_nFurniture = 1;
14         m_nFurniture2 = 2;
15         m_nFurniture3 = 3;
16     }
17
18     ~CFurniture() {
19         printf("CFurniture::~~CFurniture()\r\n");
20         m_nFurniture = 0;
21     }
22
23     virtual void sleep() {
24         printf("CFurniture::sleep()\r\n");
25     }
26
27     void sit() {
28         printf("CFurniture::sit()\r\n");
29     }
30
31     int m_nFurniture;
32
33 protected:
34     int m_nFurniture2;
35
36 private:
37     int m_nFurniture3;
38 };
```

```
39
40 // 床类
41 class CBed : virtual public CFurniture {
42 public:
43     CBed() {
44         printf("CBed::CBed()\r\n");
45         m_nBed = 2;
46     }
47
48     ~CBed() {
49         printf("CBed::~~CBed()\r\n");
50         m_nBed = 0;
51     }
52
53     virtual void sleep() {
54         printf("CBed::sleep()\r\n");
55     }
56
57     void sit() {
58         printf("CBed::sit()\r\n");
59     }
60
61 private:
62     int m_nBed;
63 };
64
65 int main()
66 {
67
68     CFurniture fur;
69     CBed bed;
70
71     int nSizeFur = sizeof(CFurniture);
72     cout << nSizeFur << endl; // 16
73     int nSizeBed = sizeof(CBed);
74     cout << nSizeBed << endl; // 20
75
76                                     // 向上转型
77                                     // 1 基类对象指向派生类对象
78     fur = bed;
79     fur.sit(); // normal function
80     fur.sleep(); // virtual function
```

```

81         // 这种方式不会发生虚函数调用，依然是调用各自类中的
      成员函数。对象赋值对成员对象和this指针无影响。
82
83         // 2 基类指针指向派生类对象的地址
84         CFurniture* pfur = &bed;
85         pfur->sit(); // normal function
86         pfur->sleep();// virtual function
87         // 由于基类通过指针调用成员函数，而sleep()成员函数
      同时也是虚函数，这时会发生虚调用，间接调用指针指向的对象的虚函数。
88
89         // 3 基类指针通过引用指向派生类对象
90         CFurniture &rfur = bed;
91         rfur.sit(); // normal function
92         rfur.sleep(); // virtual function
93         // 由于基类通过引用调用成员函数，而sleep()成员函数
      同时也是虚函数，发生虚调用，间接调用引用的那个对象的虚函数。
94
95         return 0;
96     }
97     /*
98     output:
99
100    CFurniture::CFurniture()
101    CFurniture::CFurniture()
102    CBed::CBed()
103    16
104    20
105    CFurniture::sit()
106    CFurniture::sleep()
107    CFurniture::sit()
108    CBed::sleep()
109    CBed::~~CBed()
110    CFurniture::~~CFurniture()
111    CFurniture::~~CFurniture()
112    请按任意键继续. . .
113    */

```

对象大小

```

1    int nSizeFur = sizeof(CFurniture);
2    cout << nSizeFur << endl; // 16
3    int nSizeBed = sizeof(CBed);
4    cout << nSizeBed << endl; // 28

```

对象内存分析

CFurniture fur;

地址	十六进制	注释
0059FD1C	34 8b 98 00	虚表指针
0059FD20	01 00 00 00
0059FD24	02 00 00 00	成员变量
0059FD28	03 00 00 00
0059FD2C	cc cc cc cc	????

CBed bed;

地址	十六进制	注释
0059FCE8	c0 8b 98 00	bed对象首地址到它自身虚表的偏移量的指针
0059FCFC	02 00 00 00	bed对象的成员变量
0059FD00	00 00 00 00
0059FD04	b8 8b 98 00	bed对象的虚表指针
0059FD08	01 00 00 00
0059FD0C	02 00 00 00
0059FD10	03 00 00 00	父类fur对象的成员变量
0059FD14	cc cc cc cc	????

内存 2

地址	十六进制	注释
0x00988BC0	00 00 00 00	暂时发现做什么用
0x00988BC4	0c 00 00 00
0x00988BC8	00 00 00 00
0x00988BCC	43 42 65 64	CBed
0x00988BD0	3a 3a 43 42	::CB
0x00988BD4	65 64 28 29	ed()
0x00988BD8	0d 0a 00 00
0x00988BDC	00 00 00 00
0x00988BE0	43 42 65 64	CBed
0x00988BE4	3a 3a 7e 43	::~C

偏移量计算方式:

$$0x0059FD04 - 0x0059FCF8 = 0C$$

0C 00 00 00

函数调用

见代码示例。