#include <stdio.h>

class A

{

public:

int m\_nA;

A(int a = 9)

{

m\_nA = a;

}

virtual ~A()

{}

};

int main()

{

A \*pA = new A(7);

delete []pA; 不报错 对内存正常 直接调用scala析构

return 0;

}

003B14E0 push 3 0x11

003B14E2 mov ecx,dword ptr [ebp-0ECh]

003B14E8 mov edx,dword ptr [ecx]

003B14EA mov ecx,dword ptr [ebp-0ECh]

003B14F0 mov eax,dword ptr [edx]

003B14F2 call eax

A::`vector deleting destructor':

003B11A9 jmp A::`scalar deleting destructor' (3B1610h)

003B1630 mov dword ptr [ebp-8],ecx

003B1633 mov ecx,dword ptr [this]

003B1636 call A::~A (3B1113h)

003B163B mov eax,dword ptr [ebp+8]

003B163E and eax,1 判断是否是0 否则不调用delete函数

003B1641 je A::`scalar deleting destructor'+3Fh (3B164Fh)

003B1643 mov eax,dword ptr [this]

003B1646 push eax

00841647 call operator delete (841096h)

0084164C add esp,4

0084164F mov eax,dword ptr [this]

008E14A8 mov eax,dword ptr [ebp-14h]

008E14AB mov dword ptr [ebp-0E0h],eax

008E14B1 mov ecx,dword ptr [ebp-0E0h]

008E14B7 mov dword ptr [ebp-0ECh],ecx

008E14BD mov edx,dword ptr [ebp-0ECh]

008E14C3 mov dword ptr [ebp-0F8h],edx

008E14C9 cmp dword ptr [ebp-0F8h],0

008E14D0 je main+11Dh (8E152Dh)

008E14D2 mov eax,dword ptr [ebp-0F8h]

008E14D8 cmp dword ptr [eax-4],0

008E14DC je main+0F3h (8E1503h)

008E14DE mov esi,esp

008E14E0 push 3

008E14E2 mov ecx,dword ptr [ebp-0ECh]

008E14E8 mov edx,dword ptr [ecx]

008E14EA mov ecx,dword ptr [ebp-0ECh]

008E14F0 mov eax,dword ptr [edx]

008E14F2 call eax

008E14F4 cmp esi,esp

008E14F6 call @ILT+335(\_\_RTC\_CheckEsp) (8E1154h)

008E14FB mov dword ptr [ebp-118h],eax

008E1501 jmp main+10Fh (8E151Fh) 此处跳转

008E1503 mov ecx,dword ptr [ebp-0F8h]

008E1509 sub ecx,4

008E150C push ecx

008E150D call operator delete (8E1096h) 跳过调用

008E1512 add esp,4

008E1515 mov dword ptr [ebp-118h],0

008E151F mov edx,dword ptr [ebp-118h]

008E1525 mov dword ptr [ebp-11Ch],edx

008E152B jmp main+127h (8E1537h)

008E152D mov dword ptr [ebp-11Ch],0

#include <stdio.h>

class A

{

public:

int m\_nA;

A(int a = 9)

{

m\_nA = a;

}

virtual ~A()

{}

};

int main()

{

A \*pA = new A(7);

A \*pAA = new A[5];

delete []pA;

delete pAA;

return 0;

}

Delete []pA

A::`vector deleting destructor':

00C61E00 mov dword ptr [ebp-8],ecx

00C61E03 mov eax,dword ptr [ebp+8]

00C61E06 and eax,2 高位为2调用vector

00C61E09 je A::`vector deleting destructor'+61h (0C61E41h)

00C61E0B push offset A::~A (0C61113h)

00C61E10 mov eax,dword ptr [this]

00C61E13 mov ecx,dword ptr [eax-4]

00C61E16 push ecx 此处为fdfdfdfd越界标志 出错

00C61E17 push 8

00C61E19 mov edx,dword ptr [this]

00C61E1C push edx

00C61E1D call `eh vector destructor iterator' (0C611F4h)

00C61E22 mov eax,dword ptr [ebp+8]

00C61E25 and eax,1

00C61E28 je A::`vector deleting destructor'+59h (0C61E39h)

00C61E2A mov eax,dword ptr [this]

00C61E2D sub eax,4

00C61E30 push eax

00C61E31 call operator delete (0C61096h)

00C61E36 add esp,4

00C61E39 mov eax,dword ptr [this]

00C61E3C sub eax,4

00C61E3F jmp A::`vector deleting destructor'+80h (0C61E60h)

00C61E41 mov ecx,dword ptr [this]

00C61E44 call A::~A (0C61113h)

00C61E49 mov eax,dword ptr [ebp+8]

00C61E4C and eax,1

00C61E4F je A::`vector deleting destructor'+7Dh (0C61E5Dh)

00C61E51 mov eax,dword ptr [this]

00C61E54 push eax

00C61E55 call operator delete (0C61096h)

00C61E5A add esp,4

00C61E5D mov eax,dword ptr [this]

00C61E60 pop edi

00C61E61 pop esi

00C61E62 pop ebx

00C61E63 add esp,0CCh

00C61E69 cmp ebp,esp

00C61E6B call @ILT+335(\_\_RTC\_CheckEsp) (0C61154h)

00C61E70 mov esp,ebp

00C61E72 pop ebp

00C61E73 ret 4