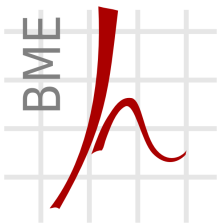
Objektum modell – Többszörös származtatás

Kódvisszafejtés.



Híradástechnikai Tanszék

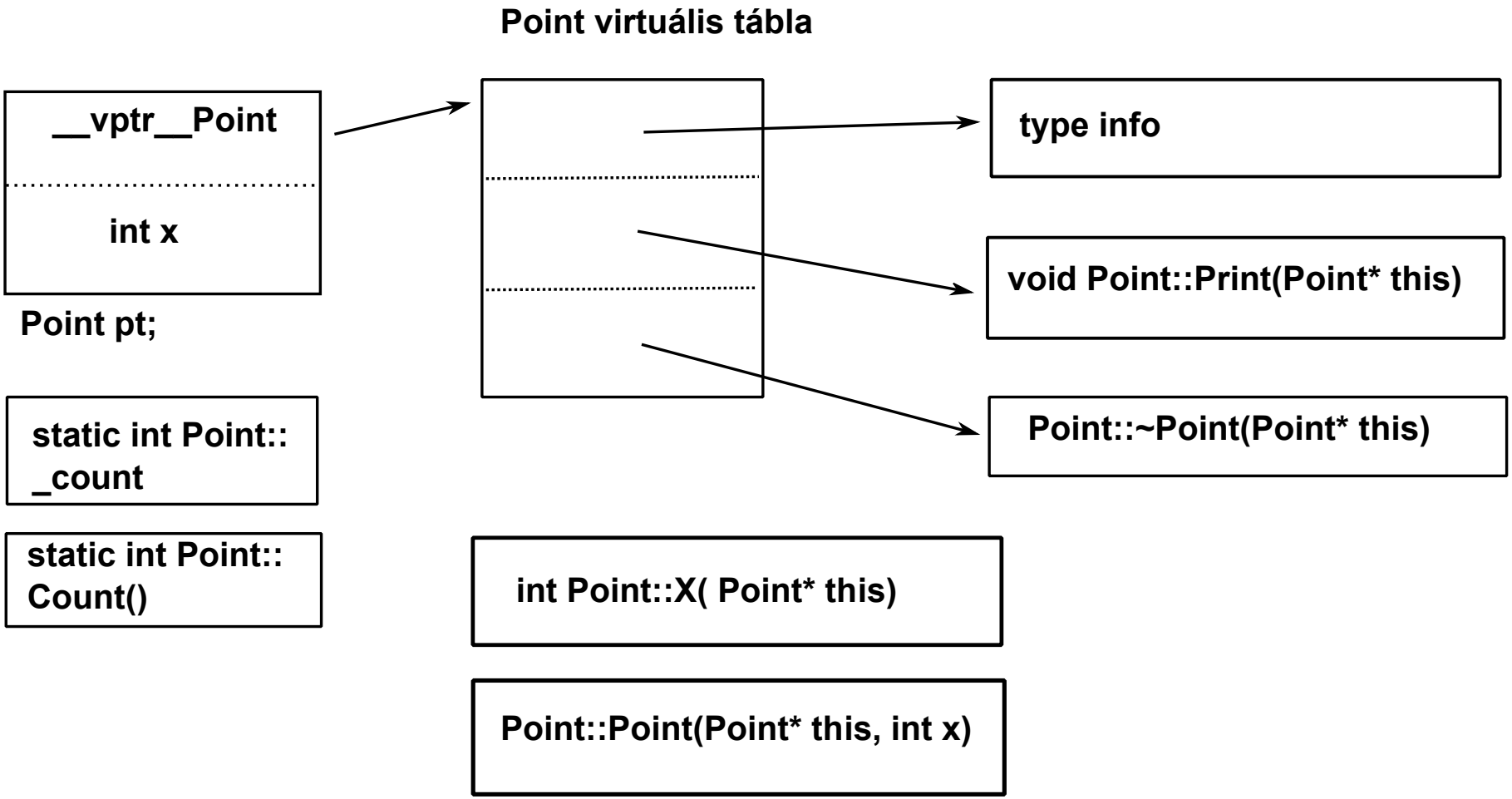
Izsó Tamás

2012. november 22.

Section 1

Objektum modell

Lehetséges C++ objektum modell



Virtuális függvény származtatás esetén

```
class PointXY : public Point {
public :
   PointXY( int x, int y );
   virtual void Print() const;
protected:
   int y;
};

PointXY::PointXY( int x, int y ) : Point(x), y(y) {}
void PointXY::Print() const {
    Point::Print(); printf("␣%d␣", y); }

int Point::_count = 0;

int main() {
   PointXY p1(10,20);
   Point& p = p1;
   p.Print();
   return 0;
}
```

Konstruktor hívás (GCC fordítóval)

```
000001C7        mov     [esp+80h+var_78], 14h
000001CF        mov     [esp+80h+var_7C], 0Ah
000001D7        lea     eax, [esp+80h+var_20]
000001DB        mov     [esp+80h+var_80], eax
000001DE        call    __ZN7PointXYC1Eii ; PointXY::PointXY(int,int)
```

A GCC a this pointert tagfüggvény első paraméterének a helyén, és nem az **ecx** regiszterben adja át.

Virtuális tagfüggvény meghívása

```
00000074 _main    proc near
00000074      push   ebp
00000075      mov    ebp, esp
00000077      sub    esp, 3
0000007A      and    esp, 0FFFFFFF8h
0000007D      add    esp, 4
00000080      sub    esp, 20h
00000083      lea    eax,  dword ptr [ebp-18h]
00000086      mov    [esp], 0Ah
0000008D      mov    dword ptr [esp+4], 14h
00000095      mov    ecx, eax
00000097      call   ??0PointXY@@QAE@HH@Z ; PointXY::PointXY(int,int)
0000009C      mov     dword ptr[ebp-0Ch], eax
0000009F      lea    eax,  dword ptr [ebp-18h]
000000A2      mov     dword ptr [ebp-8h], eax
000000A5      mov    eax,  dword ptr [ebp-8h]
000000A8      mov     dword ptr [ebp-4h], eax
000000AB      mov    eax,  dword ptr [ebp-4h]
000000AE      mov    edx, 4
000000B3      add    edx, [eax]
000000B5      mov    eax, [edx]
000000B7      mov    edx,  dword ptr [ebp-4h]
000000BA      mov    ecx, edx
000000BC      call   eax
000000BE      lea    eax,  dword ptr [ebp-18h]
000000C1      mov    ecx, eax
000000C3      call   ??1PointXY@@UAE@XZ ; PointXY::~PointXY(void)
000000C8      mov    eax, 0
000000CD      leave
000000CE      retn
000000CE _main        endp
```

Virtuális tagfüggvény meghívása

```
00000074 _main    proc near
00000074          push   ebp
00000075          mov    ebp, esp
00000077          sub    esp, 3
0000007A          and    esp, 0FFFFFFF8h
0000007D          add    esp, 4
00000080          sub    esp, 20h
00000083          lea    eax, dword ptr [ebp-18h]
00000086          mov    [esp]
0000008D          mov                4], 14h
00000095          mov
00000097          call   ??0PointXY@@QAE@HH@Z ; PointXY::PointXY(int,int)
0000009C          mov    dword ptr[ebp-0Ch], eax
0000009F          lea    eax, dword ptr [ebp-18h]
000000A2          mov    dword ptr [ebp-8h], eax
000000A5          mov    eax, dword ptr [ebp-8h]
000000A8          mov    dword ptr [ebp-4h], eax
000000AB          mov    eax, dword ptr [ebp-4h]
000000AE          mov    edx, 4
000000B3          add    edx, [eax]
000000B5          mov    eax, [edx]
000000B7          mov    edx, dword ptr [ebp-4h]
000000BA          mov    ecx, edx
000000BC          call   eax
000000BE          lea    eax, dword ptr [ebp-18h]
000000C1          mov    ecx, eax
000000C3          call   ??1PointXY@@UAE@XZ ; PointXY::~PointXY(void)
000000C8          mov    eax, 0
000000CD          leave
000000CE          retn
000000CE _main    endp
```

Referencia beállítása.

Virtuális tagfüggvény meghívása

```
00000074 _main    proc near
00000074          push   ebp
00000075          mov    ebp, esp
00000077          sub    esp, 3
0000007A          and    esp, 0FFFFFFF8h
0000007D          add    esp, 4
00000080          sub    esp, 20h
00000083          lea    eax, dword ptr [ebp-18h]
00000086          mov
0000008D          m                +4], 14h
00000095          m
00000097          c              @HH@Z ; PointXY::PointXY(int,int)
0000009C          mov            bp-0Ch], eax
0000009F          lea    ea      dword ptr [ebp-18h]
000000A2          mov     dwo    d ptr [ebp-8h], eax
000000A5          mov    eax, dword ptr [ebp-8h]
000000A8          mov     dword ptr [ebp-4h], eax
000000AB          mov    eax, dword ptr [ebp-4h]
000000AE          mov    edx, 4
000000B3          add    edx, [eax]
000000B5          mov    eax, [edx]
000000B7          mov    edx, dword ptr [ebp-4h]
000000BA          mov    ecx, edx
000000BC          call   eax
000000BE          lea    eax, dword ptr [ebp-18h]
000000C1          mov    ecx, eax
000000C3          call   ??1PointXY@@UAE@XZ ; PointXY::~PointXY(void)
000000C8          mov    eax, 0
000000CD          leave
000000CE          retn
000000CE _main            endp
```

Objektum címének elmentése egy temporális változóba.

Virtuális tagfüggvény meghívása

```
00000074 _main    proc near
00000074          push   ebp
00000075          mov    ebp, esp
00000077          sub    esp, 3
0000007A          and    esp, 0FFFFFFF8h
0000007D          add    esp, 4
00000080          sub    esp, 20h
00000083          lea    eax, dword ptr [ebp-18h]
00000086          mov    [esp], 0Ah
0000008D          mov    [esp+4], 14h
00000095          m
00000097          c                @HH@Z ; PointXY::PointXY(int,int)
0000009C          n                -0Ch], eax
0000009F          le               tr [ebp-18h]
000000A2          mov              [ebp-8h], eax
000000A5          mov    eax       dword ptr [ebp-8h]
000000A8          mov     dwo      d ptr [ebp-4h], eax
000000AB          mov    eax, dword ptr [ebp-4h]
000000AE          mov    edx, 4
000000B3          add    edx, [eax]
000000B5          mov    eax, [edx]
000000B7          mov    edx, dword ptr [ebp-4h]
000000BA          mov    ecx, edx
000000BC          call   eax
000000BE          lea    eax, dword ptr [ebp-18h]
000000C1          mov    ecx, eax
000000C3          call   ??1PointXY@@UAE@XZ ; PointXY::~PointXY(void)
000000C8          mov    eax, 0
000000CD          leave
000000CE          retn
000000CE _main    endp
```

Virtuális függvénytábla második értékének betöltése az **eax**-be.

Virtuális tagfüggvény meghívása

```
00000074 _main    proc near
00000074          push   ebp
00000075          mov    ebp, esp
00000077          sub    esp, 3
0000007A          and    esp, 0FFFFFFF8h
0000007D          add    esp, 4
00000080          sub    esp, 20h
00000083          lea    eax,  dword ptr [ebp-18h]
00000086          mov    [esp], 0Ah
0000008D          mov    dword ptr [esp+4], 14h
00000095          mov    ecx, eax
00000097          call   ??0PointXY@@QAE@HH@Z ; PointXY::PointXY(int,int)
0000009C          mov     dword ptr[ebp-0Ch], eax
0000009F          lea    eax,  dword ptr [ebp-18h]
000000A2          mov     dword ptr [ebp-8h], eax
000000A5          mov    eax,  dword ptr [ebp-8h]
000000A8          mov     dword ptr [ebp-4h], eax
000000AB          mo                 tr [ebp-4h]
000000AE          m
000000B3          add    edx
000000B5          mov    eax, [edx]
000000B7          mov    edx,  dword ptr [ebp-4h]
000000BA          mov    ecx, edx
000000BC          call   eax
000000BE          lea    eax,  dword ptr [ebp-18h]
000000C1          mov    ecx, eax
000000C3          call   ??1PointXY@@UAE@XZ ; PointXY::~PointXY(void)
000000C8          mov    eax, 0
000000CD          leave
000000CE          retn
000000CE _main            endp
```

this pointer betöltése az **ecx**-be.

Virtuális tagfüggvény meghívása

```
00000074 _main    proc near
00000074          push   ebp
00000075          mov    ebp, esp
00000077          sub    esp, 3
0000007A          and    esp, 0FFFFFFF8h
0000007D          add    esp, 4
00000080          sub    esp, 20h
00000083          lea    eax, dword ptr [ebp-18h]
00000086          mov    [esp], 0Ah
0000008D          mov    dword ptr [esp+4], 14h
00000095          mov    ecx, eax
00000097          call   ??0PointXY@@QAE@HH@Z ; PointXY::PointXY(int,int)
0000009C          mov    dword ptr[ebp-0Ch], eax
0000009F          lea    eax, dword ptr [ebp-18h]
000000A2          mov    dword ptr [ebp-8h], eax
000000A5          mov    eax, dword ptr [ebp-8h]
000000A8          mov    dword ptr [ebp-4h], eax
000000AB          mov    eax, dword ptr [ebp-4h]
000000AE          mov    edx, 4
000000B3          add    eax, 1
000000B5          mov    edx, eax
000000B7          mov    eax, dword ptr [ebp-4h]
000000BA          mov    ecx, edx
000000BC          call   eax
000000BE          lea    eax, dword ptr [ebp-18h]
000000C1          mov    ecx, eax
000000C3          call   ??1PointXY@@UAE@XZ ; PointXY::~PointXY(void)
000000C8          mov    eax, 0
000000CD          leave
000000CE          retn
000000CE _main    endp
```

Virtuális tagfüggvény hívása.

Section 2

Többszörös származtatás

Többszörös származtatás

```
#include <stdio.h>

class A {
  public:
     A( int a ) : a(a) { }
      virtual void f(int i) { printf("A␣%d\n", a); }
      virtual void printA() { printf("A::a␣%d␣%d\n", a); }
  protected:
     int a;
};

class B {
  public :
     B( int b ) : b(b) { }
      virtual void f(int i) { printf("B␣%d\n", i); }
      virtual void printB() { printf("B::b␣%d\n", b); }
  protected:
     int b;
} ;

class C : public A, public B {
  public :
     C( int a, int b, int c ) : A(a), B(b), c(c) { }
     void f(int i) { printf("C␣%d\n",i); };
      virtual void printC() { printf("C::c␣%d\n", c); }
  protected:
     int c;
} ;
```

Virtuális függvénytáblák

```
00000288 ; const A::`vftable'
00000288 ??_7A@@6B@       dd offset ?f@A@@UAEXH@Z ; A::f(int)
0000028C                  dd offset ?printA@A@@UAEXXZ ; A::printA(void)
00000290                  dd 0

000002F8 ; const B::`vftable'
000002F8 ??_7B@@6B@       dd offset ?f@B@@UAEXH@Z ; B::f(int)
000002FC                  dd offset ?printB@B@@UAEXXZ ; B::printB(void)
00000300                  dd 0

00000368 ; const C::`vftable'{for `A'}
00000368 ??_7C@@6BA@@@    dd offset ?f@C@@UAEXH@Z ;  C::f(int)
0000036C                  dd offset ?printA@A@@UAEXXZ ; A::printA(void)
00000370                  dd offset ?printC@C@@UAEXXZ ; C::printC(void)

00000404 ; const C::`vftable'{for `B'}
00000404 ??_7C@@6BB@@@    dd offset ?f@C@@W7AEXH@Z ;[thunk]:C::f`adjustor{8}' (int)
00000408                  dd offset ?printB@B@@UAEXXZ ; B::printB(void)
0000040C                  dd 0
```

C::C(int, int, int) konstruktor

```
00000120 ; public: __thiscall C::C(int, int, int)
00000120         public ??0C@@QAE@HHH@Z
00000120 ??0C@@QAE@HHH@Z proc near
00000120         push    ebp
00000121         mov     ebp, esp
00000123         sub     esp, 10h
00000126         mov     [ebp - 0Ch], ecx
00000129         mov     eax, [ebp - 0Ch]
0000012C         mov     edx, [ebp+8] ; a paraméter
0000012F         mov     [esp], edx
00000132         mov     ecx, eax
00000134         call    ??0A@@QAE@H@Z   ; A::A(int)
00000139         mov     [ebp-8], eax
0000013C         push    eax
0000013D         mov     eax, [ebp-0Ch]
00000140         lea     eax, [eax+8]  ; this pointer növelése
00000143         mov     edx, [ebp+0Ch] ; b paraméter
00000146         mov     [esp], edx
00000149         mov     ecx, eax
0000014B         call    ??0B@@QAE@H@Z   ; B::B(int)
00000150         mov     [ebp-4], eax
00000153         mov     eax, [ebp-0Ch]
00000156         mov     dword ptr [eax], offset ??_7C@@6BA@@@ ; const C::`vftable'{for `A'}
0000015C         mov     eax, [ebp-0Ch]
0000015F         mov     dword ptr [eax+8], offset ??_7C@@6BB@@@ ; const C::`vftable'{for `B'}
00000166         mov     eax, [ebp-0Ch]
00000169         mov     edx, [ebp+10h]  ; c paraméter
0000016C         mov     [eax+10h], edx
0000016F         mov     eax, [ebp-0Ch]
00000172         leave
00000173         retn    0Ch
00000173 ??0C@@QAE@HHH@Z endp
```

C::C(int, int, int) konstruktor

```
00000120 ; public: __thiscall C::C(int, int, int)
00000120         public: [illegible]@@QAE@HHH@Z
00000120 ??0C@
00000120
00000121         mov     [illegible], esp
00000123         sub     esp, 10h
00000126         mov     [ebp - 0Ch], ecx
00000129         mov     eax, [ebp - 0Ch]
0000012C         mov     edx, [ebp+8] ; a paraméter
0000012F         mov     [esp], edx
00000132         mov     ecx, eax
00000134         call    ??0A@@QAE@H@Z   ; A::A(int)
00000139         mov     [ebp-8], eax
0000013C         push    eax
0000013D         mov     eax, [ebp-0Ch]
00000140         lea     eax, [eax+8]  ; this pointer növelése
00000143         mov     edx, [ebp+0Ch] ; b paraméter
00000146         mov     [esp], edx
00000149         mov     ecx, eax
0000014B         call    ??0B@@QAE@H@Z   ; B::B(int)
00000150         mov     [ebp-4], eax
00000153         mov     eax, [ebp-0Ch]
00000156         mov     dword ptr [eax], offset ??_7C@@6BA@@@ ; const C::`vftable'{for `A'}
0000015C         mov     eax, [ebp-0Ch]
0000015F         mov     dword ptr [eax+8], offset ??_7C@@6BB@@@ ; const C::`vftable'{for `B'}
00000166         mov     eax, [ebp-0Ch]
00000169         mov     edx, [ebp+10h]  ; c paraméter
0000016C         mov     [eax+10h], edx
0000016F         mov     eax, [ebp-0Ch]
00000172         leave
00000173         retn    0Ch
00000173 ??0C@@QAE@HHH@Z endp
```

this pointer elmentése egy lokális változóba.

C::C(int, int, int) konstruktor

A::A(int) konstruktor hívása. Vegyük észre, hogy az átadott this pointer megegyezik a C this pointerével.

```
00000120 ; pu                    , int , int )
00000120                         Z
00000120 ??0C
00000120
00000121
00000123          sub            , 10h
00000126          mov            ebp - 0Ch], ecx
00000129          mov            eax, [ebp - 0Ch]
0000012C          mov            edx, [ebp+8] ; a paraméter
0000012F          mov            [esp], edx
00000132          mov            ecx, eax
00000134          call           ??0A@@QAE@H@Z   ; A::A(int)
00000139          mov            [ebp-8], eax
0000013C          push           eax
0000013D          mov            eax, [ebp-0Ch]
00000140          lea            eax, [eax+8]  ; this pointer növelése
00000143          mov            edx, [ebp+0Ch] ; b paraméter
00000146          mov            [esp], edx
00000149          mov            ecx, eax
0000014B          call           ??0B@@QAE@H@Z   ; B::B(int)
00000150          mov            [ebp-4], eax
00000153          mov            eax, [ebp-0Ch]
00000156          mov            dword ptr [eax], offset ??_7C@@6BA@@@ ; const C::`vftable'{for `A'}
0000015C          mov            eax, [ebp-0Ch]
0000015F          mov            dword ptr [eax+8], offset ??_7C@@6BB@@@ ; const C::`vftable'{for `B'}
00000166          mov            eax, [ebp-0Ch]
00000169          mov            edx, [ebp+10h]  ; c paraméter
0000016C          mov            [eax+10h], edx
0000016F          mov            eax, [ebp-0Ch]
00000172          leave
00000173          retn           0Ch
00000173 ??0C@@QAE@HHH@Z endp
```

C::C(int, int, int) konstruktor

```
00000120 ; public: __thiscall C::C(int, int, int)
00000120         public ??0C@@QAE@HHH@Z
00000120 ??0C@@QAE@HHH@Z proc near
00000120         push    ebp
00000121         m
00000123
00000126                                 ecx
00000129                                 Ch]
0000012C                                 ; a paraméter
0000012F
00000132
00000134                                 @H@Z    ; A::A(int)
00000139         mov                     p-8], eax
0000013C         push                    x
0000013D         mov     eax, [ebp-0Ch]
00000140         lea     eax, [eax+8]    ; this pointer növelése
00000143         mov     edx, [ebp+0Ch]  ; b paraméter
00000146         mov     [esp], edx
00000149         mov     ecx, eax
0000014B         call    ??0B@@QAE@H@Z   ; B::B(int)
00000150         mov     [ebp-4], eax
00000153         mov     eax, [ebp-0Ch]
00000156         mov     dword ptr [eax], offset ??_7C@@6BA@@@ ; const C::`vftable'{for `A'}
0000015C         mov     eax, [ebp-0Ch]
0000015F         mov     dword ptr [eax+8], offset ??_7C@@6BB@@@ ; const C::`vftable'{for `B'}
00000166         mov     eax, [ebp-0Ch]
00000169         mov     edx, [ebp+10h]  ; c paraméter
0000016C         mov     [eax+10h], edx
0000016F         mov     eax, [ebp-0Ch]
00000172         leave
00000173         retn    0Ch
00000173 ??0C@@QAE@HHH@Z endp
```

B::B(int) konstruktor hívása. Vegyük észre, hogy az átadott this pointer 8 byte-tal odébb mutat, mint a C this pointere.

C::C(int, int, int) konstruktor

```
00000120 ; public: __thiscall C::C(int, int, int)
00000120         public ??0C@@QAE@HHH@Z
00000120 ??0C@@QAE@HHH@Z proc near
00000120         push    ebp
00000121         mov     ebp, esp
00000123         sub     esp, 10h
00000126         mov     [ebp - 0Ch], ecx
00000129         mov     eax, [ebp - 0Ch]
0000012C         mov     edx, [ebp+8] ; a paraméter
0000012F         mov     [esp], edx
00000132         mov     [esp+4], eax
00000134         call    ??0A@@QAE@H@Z   ; A::A(int)
00000139         mov     eax, ecx
0000013C         mov     [ebp-4], eax
0000013D         mov     eax, [ebp-4]
00000140         add     eax, 8          ; this pointer növelése
00000143         mov     edx, [ebp+0Ch]  ; b paraméter
00000146         mov     [esp+4], edx
00000149         mov     eax, [ebp-4]
0000014B         call    ??0B@@QAE@H@Z   ; B::B(int)
00000150         mov     eax, [ebp-4], eax
00000153         mov     eax, [ebp-0Ch]
00000156         mov     dword ptr [eax], offset ??_7C@@6BA@@@ ; const C::`vftable'{for `A'}
0000015C         mov     eax, [ebp-0Ch]
0000015F         mov     dword ptr [eax+8], offset ??_7C@@6BB@@@ ; const C::`vftable'{for `B'}
00000166         mov     eax, [ebp-0Ch]
00000169         mov     edx, [ebp+10h]  ; c paraméter
0000016C         mov     [eax+10h], edx
0000016F         mov     eax, [ebp-0Ch]
00000172         leave
00000173         retn    0Ch
00000173 ??0C@@QAE@HHH@Z endp
```

Virtuális függvénytáblára mutató pointer beállítása. Itt van az A és a C osztály virtuális függvénytáblája, ezért a két osztály között meg van osztva.

C::C(int, int, int) konstruktor

```
00000120 ; public: __thiscall C::C(int, int, int)
00000120         public ??0C@@QAE@HHH@Z
00000120 ??0C@@QAE@HHH@Z proc near
00000120         push    ebp
00000121         mov     ebp, esp
00000123         sub     esp, 10h
00000126         mov     [ebp - 0Ch], ecx
00000129         mov     eax, [ebp - 0Ch]
0000012C         mov     edx, [ebp+8] ; a paraméter
0000012F         mov     [esp], edx
00000132         mov     ecx, eax
00000134         call    ??0A@@QAE@H@Z   ; A::A(int)
00000139         mov     [ebp-8], eax
0000013C         pu
0000013D                                 Ch]
00000140                                 ; this pointer növelése
00000143                                 ; b paraméter
00000146
00000149
0000014B                                 Z   ; B::B(int)
00000150                                 eax
00000153         mov     a, [ebp-0Ch]
00000156         mov     dword ptr [eax], offset ??_7C@@6BA@@@ ; const C::`vftable'{for `A'}
0000015C         mov     eax, [ebp-0Ch]
0000015F         mov     dword ptr [eax+8], offset ??_7C@@6BB@@@ ; const C::`vftable'{for `B'}
00000166         mov     eax, [ebp-0Ch]
00000169         mov     edx, [ebp+10h]  ; c paraméter
0000016C         mov     [eax+10h], edx
0000016F         mov     eax, [ebp-0Ch]
00000172         leave
00000173         retn    0Ch
00000173 ??0C@@QAE@HHH@Z endp
```

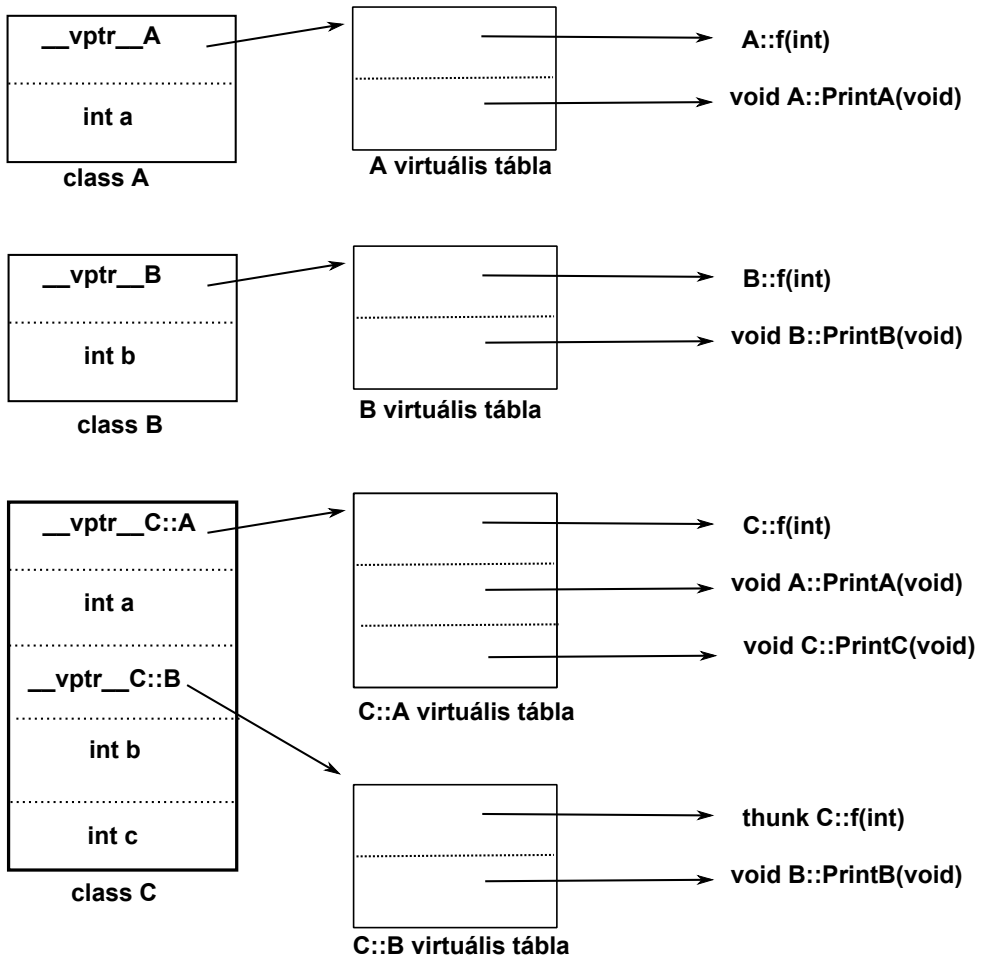
A C-ben lévő B alap-osztály virtuális függ-vénytáblára mutató pointer beállítása. A helye az objektum elejétől 8 byte-ra van.

C::C(int, int, int) konstruktor

```
00000120 ; public: __thiscall C::C(int, int, int)
00000120         public ??0C@@QAE@HHH@Z
00000120 ??0C@@QAE@HHH@Z proc near
00000120         push    ebp
00000121         mov     ebp, esp
00000123         sub     esp, 10h
00000126         mov     [ebp - 0Ch], ecx
00000129         mov     eax, [ebp - 0Ch]
0000012C         mov     edx, [ebp+8] ; a paraméter
0000012F         mov     [esp], edx
00000132         mov     ecx, eax
00000134         call    ??0A@@QAE@H@Z   ; A::A(int)
00000139         mov     [ebp-8], eax
0000013C         push    eax
0000013D         mov     eax, [ebp-0Ch]
00000140         lea     eax, [eax+8]  ; this pointer növelése
00000143         mov     edx, [ebp+0Ch] ; b paraméter
00000146         mov     [esp], edx
00000149
0000014B                                 ; B::B(int)
00000150
00000153
00000156                 [eax], offset ??_7C@@6BA@@@ ; const C::`vftable'{for `A'}
0000015C         mov     , [ebp-0Ch]
0000015F         mov     word ptr [eax+8], offset ??_7C@@6BB@@@ ; const C::`vftable'{for `B'}
00000166         mov     eax, [ebp-0Ch]
00000169         mov     edx, [ebp+10h]  ; c paraméter
0000016C         mov     [eax+10h], edx
0000016F         mov     eax, [ebp-0Ch]
00000172         leave
00000173         retn    0Ch
00000173 ??0C@@QAE@HHH@Z endp
```

A C osztály tagváltozójának a beállítása. A helye az objektum elejétől 16 byte-ra van.

A, B és C osztályok rekonstruálása



Többszörös származtatás

```
int main(int argc, char* argv[] )
{
  C c(16,32,64);
  B b(16);
  B* pb;
  C* pc=&c;  // Hol van a c obj. a memóriában?
  if( argc > 2 )
  {
     pb=&c;  // Hova fog mutatni a pointer?
  }
  else
  {
     pb = &b; // pb a b-re mutat
  }
  pb->f(20); // ?::f(int)
  return 0;
}
```

C::C(int, int, int) konstruktor

```
000001C8 _main          proc near
000001C8        push    ebp
000001C9        mov     ebp, esp
000001CB        sub     esp, 3
000001CE        and     esp, 0FFFFFFF8h
000001D1        add     esp, 4
000001D4        sub     esp, 3Ch
000001D7        lea     eax, [ebp - 30h] ; C::c címe
000001DA        mov     [esp], 10h
000001E1        mov     [esp+4], 20h ; ' '
000001E9        mov     dword ptr [esp+8], 40h ; '@'
000001F1        mov     ecx, eax
000001F3        call    ??0C@@QAE@HHH@Z ; C::C(int,int,int)
000001F8        mov     [ebp - 1Ch], eax
000001FB        push    eax
000001FC        lea     eax, [ebp - 18h] ; B::b címe
000001FF        mov     dword ptr [esp], 10h
00000206        mov     ecx, eax
00000208        call    ??0B@@QAE@H@Z   ; B::B(int)
0000020D        mov     [ebp - 10h], eax
```

C::C(int, int, int) konstruktor

```
000001C8 _main          proc near
000001C8        push    ebp
000001C9        mov     ebp, esp
000001CB
000001CE                                    8h
000001D1        add     sp, 4
000001D4        sub     sp, 3Ch
000001D7        lea     eax, [ebp - 30h] ; C::c címe
000001DA        mov     [esp], 10h
000001E1        mov     [esp+4], 20h ; ' '
000001E9        mov     dword ptr [esp+8], 40h ; '@'
000001F1        mov     ecx, eax
000001F3        call    ??0C@@QAE@HHH@Z ; C::C(int,int,int)
000001F8        mov     [ebp - 1Ch], eax
000001FB        push    eax
000001FC        lea     eax, [ebp - 18h] ; B::b címe
000001FF        mov     dword ptr [esp], 10h
00000206        mov     ecx, eax
00000208        call    ??0B@@QAE@H@Z   ; B::B(int)
0000020D        mov     [ebp - 10h], eax
```

C c objektum konstruktorának a hívása.

C::C(int, int, int) konstruktor

```
000001C8 _main          proc near
000001C8        push    ebp
000001C9        mov     ebp, esp
000001CB        sub     esp, 3
000001CE        and     esp, 0FFFFFFF8h
000001D1        add     esp, 4
000001D4        sub     esp, 3Ch
000001D7        lea     eax, [ebp - 30h] ; C::c címe
000001DA        mov     [esp], 10h
000001E1        mov     [esp+4], 20h ; ' '
000001E9        mov     dword ptr [esp+8], 40h ; '@'
000001F1
000001F3                                @Z ; C::C(int,int,int)
000001F8        mov     [ebp - 1Ch], eax
000001FB        push    eax
000001FC        lea     eax, [ebp - 18h] ; B::b címe
000001FF        mov     dword ptr [esp], 10h
00000206        mov     ecx, eax
00000208        call    ??0B@@QAE@H@Z   ; B::B(int)
0000020D        mov     [ebp - 10h], eax
```

B b objektum konstruktorának a hívása.

C::C(int, int, int) konstruktor

```
00000210          lea     eax, [ebp - 30h]
00000213          mov     [ebp-0CH], eax
00000216          mov     eax, [ebp+8]    ; argc
00000219          cmp     eax, 2
0000021C          jle     short loc_226
0000021E          lea     eax, [ebp-28h]  ; C::B !
00000221          mov     [ebp-8H], eax
00000224          jmp     short loc_22C
00000226 loc_226:
00000226          lea     eax, [ebp-18h]   ; B::b
00000229          mov     [ebp-8h], eax
0000022C loc_22C:
0000022C          mov     eax, [ebp-8]    ; eax <- pb
0000022F          mov     [ebp-4], eax    ; tmp <- eax
00000232          mov     eax, [ebp-4]    ; eax <- tmp
00000235          mov     eax, [eax]      ; eax <- vptr
00000237          mov     eax, [eax]      ; eax <- &f(int)
00000239          push    eax             ; ???
0000023A          mov     edx, [ebp-4]    ; edx <- this pointer
0000023D          mov     dword ptr [esp], 14h ; paraméter (20)
00000244          mov     ecx, edx        ; ecx <- this pointer
00000246          call    eax             ; call f(int)
00000248          mov     eax, 0
0000024D          leave
0000024E          retn
0000024E _main            endp
```

C::C(int, int, int) konstruktor

```
00000210          lea     eax, [ebp - 30h]
00000213          mov     [ebp-0CH], eax
00000216          mov     eax, [ebp+8]    ; argc
00000219          cmp     [ebp+0Ch], 1
0000021C          jle     short loc_226
0000021E          lea     eax, [ebp-28h]  ; C::B !
00000221          mov     [ebp-8H], eax
00000224          jmp     short loc_22C
00000226  loc_226:
00000226          lea     eax, [ebp-18h]   ; B::b
00000229          mov     [ebp-8h], eax
0000022C  loc_22C:
0000022C          mov     eax, [ebp-8]    ; eax <- pb
0000022F          mov     [ebp-4], eax    ; tmp <- eax
00000232          mov     eax, [ebp-4]    ; eax <- tmp
00000235          mov     eax, [eax]      ; eax <- vptr
00000237          mov     eax, [eax]      ; eax <- &f(int)
00000239          push    eax             ; ???
0000023A          mov     edx, [ebp-4]    ; edx <- this pointer
0000023D          mov     dword ptr [esp], 14h ; paraméter (20)
00000244          mov     ecx, edx        ; ecx <- this pointer
00000246          call    eax             ; call f(int)
00000248          mov     eax, 0
0000024D          leave
0000024E          retn
0000024E  _main           endp
```

C* pc=&c;

C::C(int, int, int) konstruktor

```
00000210        lea     eax, [ebp - 30h]
00000213        mov     [ebp-0CH], eax
00000216                                    ; argc
00000219        cmp     [illegible], 2
0000021C        jle     short loc_226
0000021E        lea     eax, [ebp-28h]  ; C::B !
00000221        mov     [ebp-8H], eax
00000224        jmp     short loc_22C
00000226 loc_226:
00000226        lea     eax, [ebp-18h]   ; B::b
00000229        mov     [ebp-8h], eax
0000022C loc_22C:
0000022C        mov     eax, [ebp-8]    ; eax <- pb
0000022F        mov     [ebp-4], eax    ; tmp <- eax
00000232        mov     eax, [ebp-4]    ; eax <- tmp
00000235        mov     eax, [eax]      ; eax <- vptr
00000237        mov     eax, [eax]      ; eax <- &f(int)
00000239        push    eax             ; ???
0000023A        mov     edx, [ebp-4]    ; edx <- this pointer
0000023D        mov     dword ptr [esp], 14h ; paraméter (20)
00000244        mov     ecx, edx        ; ecx <- this pointer
00000246        call    eax             ; call f(int)
00000248        mov     eax, 0
0000024D        leave
0000024E        retn
0000024E _main          endp
```

pb c-re mutat.

C::C(int, int, int) konstruktor

```
00000210          lea     eax, [ebp - 30h]
00000213          mov     [ebp-0CH], eax
00000216          mov     eax, [ebp+8]    ; argc
00000219          cmp     eax, 2
0000021C          jle     short loc_226
0000021E          lea     eax, [ebp-28h]  ; C::B !
00000221
00000224          jmp     short loc_22C
00000226  loc_226:
00000226          lea     eax, [ebp-18h]  ; B::b
00000229          mov     [ebp-8h], eax
0000022C  loc_22C:
0000022C          mov     eax, [ebp-8]    ; eax <- pb
0000022F          mov     [ebp-4], eax    ; tmp <- eax
00000232          mov     eax, [ebp-4]    ; eax <- tmp
00000235          mov     eax, [eax]      ; eax <- vptr
00000237          mov     eax, [eax]      ; eax <- &f(int)
00000239          push    eax             ; ???
0000023A          mov     edx, [ebp-4]    ; edx <- this pointer
0000023D          mov     dword ptr [esp], 14h ; paraméter (20)
00000244          mov     ecx, edx        ; ecx <- this pointer
00000246          call    eax             ; call f(int)
00000248          mov     eax, 0
0000024D          leave
0000024E          retn
0000024E  _main           endp
```

pb b-re mutat.

C::C(int, int, int) konstruktor

```
00000210        lea     eax, [ebp - 30h]
00000213        mov     [ebp-0CH], eax
00000216        mov     eax, [ebp+8]    ; argc
00000219        cmp     eax, 2
0000021C        jle     short loc_226
0000021E        lea     eax, [ebp-28h]  ; C::B !
00000221        mov     [ebp-8H], eax
00000224        jmp     short loc_22C
00000226 loc
00000226                                ]   ; B::b
00000229        mov                 ], eax
0000022C loc_22C:
0000022C        mov     eax, [ebp-8]    ; eax <- pb
0000022F        mov     [ebp-4], eax    ; tmp <- eax
00000232        mov     eax, [ebp-4]    ; eax <- tmp
00000235        mov     eax, [eax]      ; eax <- vptr
00000237        mov     eax, [eax]      ; eax <- &f(int)
00000239        push    eax             ; ???
0000023A        mov     edx, [ebp-4]    ; edx <- this pointer
0000023D        mov     dword ptr [esp], 14h ; paraméter (20)
00000244        mov     ecx, edx        ; ecx <- this pointer
00000246        call    eax             ; call f(int)
00000248        mov     eax, 0
0000024D        leave
0000024E        retn
0000024E _main          endp
```

Virtuális függvény meghívása.

Probléma

- Az pb pointer egy B* típusú adatra mutat.
- Ha a b-re mutat, akkor
  - a b objektum első byte-jára mutat;
  - a B::f(int) virtuális függvényt hívja meg a virtuális függvénytábla segítségével;
  - az átadott this pointer a b objektum elejére mutat.
- Ha a c-re mutat, akkor
  - a c objektumba lévő C::B részre, azaz a c közepébe mutat;
  - a C::f(int) virtuális függvényt kellene meghívni;
  - az átadott this pointer a c objektum belsejébe, és nem az elejére mutat, de a C::f(int) függvénynek az objektum elejére mutató pointert kell átadni.

Probléma megoldása 1.

A this pointert módosítani kell, de honnan tudjuk, hogy mennyivel?

- A virtuális függvénytáblába írjuk be, hogy az alapobjektum milyen messze van a származtatott objektum elejétől.
- Bjarne Stroustrup ezt a módszert használja a CFRONT fordítóhoz.

Pszeudó kód

```
(*pBase2->vptr[0].faddr)( pBase2 + pBase2->vptr[0].offset )
```

Probléma megoldása 2.

Előbb egy kis töredék programot meghívunk, amely korrigálja a this pointert. Ezt a programot thunk-nek hívják, állítólag azért mert hasonló módszert Donald Knuth vezetett be az Algol programozási nyelvbe, amit thunk-nak hívott, és a thunk visszafele olvasva a nevét adja :) .

1. A this pointer kiigazítás
2. ugrás a virtuális függvény kódjára

Pszeudó kód

```
pbase2_f_thunk:
  this -= sizeof( base1 );
  goto C::f;
```

thunk C::f‘adjustor8

```
00000250 ; [thunk]:public: virtual void __thiscall C::f‘adjustor{8}’ (int)
00000250                 public ?f@C@@W7AEXH@Z
00000250 ?f@C@@W7AEXH@Z  proc near   ; DATA XREF: .data1:const C::‘vftable’{for ‘B’}
00000250 var_4           = dword ptr -4
00000250         push    ebp
00000251         mov     ebp, esp
00000253         push    esi
00000254         mov     [ebp+var_4], ecx
00000257         mov     eax, 0FFFFFFF8h
0000025C         add     eax, [ebp+var_4]
0000025F         mov     [ebp+var_4], eax
00000262         mov     eax, [ebp+var_4]
00000265         mov     ecx, eax
00000267         leave
00000268         jmp     ?f@C@@UAEXH@Z   ; C::f(int)
00000268 ?f@C@@W7AEXH@Z  endp
```

Optimalizált kód

```
00000000 _main            proc near
00000000        push    ebp
00000001        mov     ebp, esp
00000003        and     esp, 0FFFFFF80h
00000006        sub     esp, 80h
0000000C        push    3
0000000E        call    ___intel_new_proc_init
00000013        add     esp, 4
00000016        mov     eax, 10h
0000001B        stmxcsr [esp+1Ch]
00000020        mov     [esp+4], eax
00000024        lea     ecx, [esp+8]     ; C::B címe
00000028        cmp     [ebp+8], 2
0000002C        lea     edx, [esp+14h]
00000030        mov     [esp+0Ch], 20h ; ' '
00000038        jg      short loc_3C
0000003A        mov     ecx, edx
0000003C loc_3C:
0000003C        mov     [esp], offset ??_7C@@6BA@@@ ; const C::`vftable'{for `A'}
00000043        mov     [esp+8], offset ??_7C@@6BB@@@ ; const C::`vftable'{for `B'}
0000004B        mov     [esp+10h], 40h ; '@'
00000053        mov     [esp+14h], offset ??_7B@@6B@ ; const B::`vftable'
0000005B        mov     [esp+18h], eax
0000005F        mov     eax, [ecx]
00000061        or      [esp+1Ch], 8000h
00000069        ldmxcsr [esp+1Ch]
0000006E        push    14h
00000070        call    dword ptr [eax]
00000072        xor     eax, eax
00000074        mov     esp, ebp
00000076        pop     ebp
00000077        retn
```

Startup kód (MSVC)

main() program futása előtt szükséges inicializálások

- alacsony szintű fájlkezelés inicializálása
- heap inicializálás
- multi-thread initializálás
- command line argumentum átvétele
- enviroment változók átvétele
- C adatszerkezetek inicializálása
- atexit() – kilépés elötti a paraméterként átadott függvény-pointer által mutatott eljárás meghívódik.
- globális C++ objektumok konstruktorainak meghívása
- main() program hívása (három paraméter kerül a stack-re)
- cexit() – destruktorok és egyéb adatlezáró rutinok meghívása

Minimál alkalmazás

```
int add() {
    int i=2,j=3,k;
    k=i+j;
    return k;
}
```

```
cl -c dummy.c
link /Entry:add /Subsystem:console dummy.obj
```

```
CPU Disasm
00401000    55              PUSH EBP
00401001    8BEC            MOV EBP,ESP
00401003    83EC 0C         SUB ESP,0C
00401006    C745 FC 02000   MOV DWORD PTR SS:[LOCAL.1],2
0040100D    C745 F4 03000   MOV DWORD PTR SS:[LOCAL.3],3
00401014    8B45 FC         MOV EAX,DWORD PTR SS:[LOCAL.1]
00401017    0345 F4         ADD EAX,DWORD PTR SS:[LOCAL.3]
0040101A    8945 F8         MOV DWORD PTR SS:[LOCAL.2],EAX
0040101D    33C0            XOR EAX,EAX
0040101F    8BE5            MOV ESP,EBP
00401021    5D              POP EBP
00401022    C3              RETN
```