#include <stdio.h>

void fun(int n)

{

static int s\_n = 2;

++s\_n;

static int s\_n2 = n;

s\_n2 += s\_n;

static int s\_n3 = s\_n2;

s\_n3 += 3;

}

int main()

{

int \*pS = (int \*)0x00424a30;

printf("%d\n" , \*pS);

fun(9);

return 0;

}

1: #include <stdio.h>

2:

3: void fun(int n)

4: {

0040D740 push ebp

0040D741 mov ebp,esp

0040D743 sub esp,40h

0040D746 push ebx

0040D747 push esi

0040D748 push edi

0040D749 lea edi,[ebp-40h]

0040D74C mov ecx,10h

0040D751 mov eax,0CCCCCCCCh

0040D756 rep stos dword ptr [edi]

5: static int s\_n = n;

0040D758 xor eax,eax

0040D75A mov al,[`fun'::`2'::$S1 (00427e3c)]

0040D75F and eax,1

0040D762 test eax,eax

0040D764 jne fun+3Eh (0040d77e)

0040D766 mov cl,byte ptr [`fun'::`2'::$S1 (00427e3c)]

0040D76C or cl,1

0040D76F mov byte ptr [`fun'::`2'::$S1 (00427e3c)],cl

0040D775 mov edx,dword ptr [ebp+8]

0040D778 mov dword ptr [`fun'::`2'::$S1+4 (00427e40)],edx

6: ++s\_n;

0040D77E mov eax,[`fun'::`2'::$S1+4 (00427e40)]

0040D783 add eax,1

0040D786 mov [`fun'::`2'::$S1+4 (00427e40)],eax

7: }

0040D78B pop edi

0040D78C pop esi

0040D78D pop ebx

0040D78E mov esp,ebp

0040D790 pop ebp

0040D791 ret

8:

9: int main()

10: {

00401060 push ebp

00401061 mov ebp,esp

00401063 sub esp,44h

00401066 push ebx

00401067 push esi

00401068 push edi

00401069 lea edi,[ebp-44h]

0040106C mov ecx,11h

00401071 mov eax,0CCCCCCCCh

00401076 rep stos dword ptr [edi]

11: int \*pS = (int \*)0x00424a30; //static变量地址

00401078 mov dword ptr [ebp-4],offset \_\_\_xt\_z+104h (00424a30)

12: printf("%d\n" , \*pS);

0040107F mov eax,dword ptr [ebp-4]

00401082 mov ecx,dword ptr [eax]

00401084 push ecx

00401085 push offset string "%d\n" (0042210c)

0040108A call printf (0040d6c0)

0040108F add esp,8

13: fun(9);

00401092 push 9

00401094 call @ILT+5(fun) (0040100a)

00401099 add esp,4

14: return 0;

0040109C xor eax,eax

15: }

0040109E pop edi

0040109F pop esi

004010A0 pop ebx

004010A1 add esp,44h

004010A4 cmp ebp,esp

004010A6 call \_\_chkesp (004010b0)

004010AB mov esp,ebp

004010AD pop ebp

004010AE ret

Static变量 常量初始化直接在编译时期完成

以变量初始化则会比较一下标识位是否已经置1。

3: void fun(int n)

4: {

0040D740 push ebp

0040D741 mov ebp,esp

0040D743 sub esp,40h

0040D746 push ebx

0040D747 push esi

0040D748 push edi

0040D749 lea edi,[ebp-40h]

0040D74C mov ecx,10h

0040D751 mov eax,0CCCCCCCCh

0040D756 rep stos dword ptr [edi]

5: static int s\_n = 2;

6: ++s\_n;

0040D758 mov eax,[\_\_\_xt\_z+104h (00424a30)]

0040D75D add eax,1

0040D760 mov [\_\_\_xt\_z+104h (00424a30)],eax

7: static int s\_n2 = n;

0040D765 xor ecx,ecx

0040D767 mov cl,byte ptr [`fun'::`2'::$S1 (00427e3c)]

0040D76D and ecx,1

0040D770 test ecx,ecx

0040D772 jne fun+4Bh (0040d78b)

0040D774 mov dl,byte ptr [`fun'::`2'::$S1 (00427e3c)]

0040D77A or dl,1

0040D77D mov byte ptr [`fun'::`2'::$S1 (00427e3c)],dl

0040D783 mov eax,dword ptr [ebp+8]

0040D786 mov [`fun'::`2'::$S1+4 (00427e40)],eax

8: s\_n2 += s\_n;

0040D78B mov ecx,dword ptr [`fun'::`2'::$S1+4 (00427e40)]

0040D791 add ecx,dword ptr [\_\_\_xt\_z+104h (00424a30)]

0040D797 mov dword ptr [`fun'::`2'::$S1+4 (00427e40)],ecx

9: }

0040D79D pop edi

0040D79E pop esi

0040D79F pop ebx

0040D7A0 mov esp,ebp

0040D7A2 pop ebp

0040D7A3 ret

3: void fun(int n)

4: {

0040D740 push ebp

0040D741 mov ebp,esp

0040D743 sub esp,40h

0040D746 push ebx

0040D747 push esi

0040D748 push edi

0040D749 lea edi,[ebp-40h]

0040D74C mov ecx,10h

0040D751 mov eax,0CCCCCCCCh

0040D756 rep stos dword ptr [edi]

5: static int s\_n = 2;

6: ++s\_n;

0040D758 mov eax,[\_\_\_xt\_z+104h (00424a30)]

0040D75D add eax,1

0040D760 mov [\_\_\_xt\_z+104h (00424a30)],eax

7: static int s\_n2 = n;

0040D765 xor ecx,ecx

0040D767 mov cl,byte ptr [`fun'::`2'::$S1 (00427e40)]

0040D76D and ecx,1

0040D770 test ecx,ecx

0040D772 jne fun+4Bh (0040d78b)

0040D774 mov dl,byte ptr [`fun'::`2'::$S1 (00427e40)]

0040D77A or dl,1

0040D77D mov byte ptr [`fun'::`2'::$S1 (00427e40)],dl

0040D783 mov eax,dword ptr [ebp+8]

0040D786 mov [`fun'::`2'::$S1+4 (00427e44)],eax

8: s\_n2 += s\_n;

0040D78B mov ecx,dword ptr [`fun'::`2'::$S1+4 (00427e44)]

0040D791 add ecx,dword ptr [\_\_\_xt\_z+104h (00424a30)]

0040D797 mov dword ptr [`fun'::`2'::$S1+4 (00427e44)],ecx

9: static int s\_n3 = s\_n2;

0040D79D xor edx,edx

0040D79F mov dl,byte ptr [`fun'::`2'::$S1 (00427e40)]

0040D7A5 and edx,2

0040D7A8 test edx,edx

0040D7AA jne fun+84h (0040d7c4)

0040D7AC mov al,[`fun'::`2'::$S1 (00427e40)]

0040D7B1 or al,2

0040D7B3 mov [`fun'::`2'::$S1 (00427e40)],al

0040D7B8 mov ecx,dword ptr [`fun'::`2'::$S1+4 (00427e44)]

0040D7BE mov dword ptr [\_\_stdbuf+8 (00427e3c)],ecx

10: s\_n3 += 3;

0040D7C4 mov edx,dword ptr [\_\_stdbuf+8 (00427e3c)]

0040D7CA add edx,3

0040D7CD mov dword ptr [\_\_stdbuf+8 (00427e3c)],edx

11: }

0040D7D3 pop edi

0040D7D4 pop esi

0040D7D5 pop ebx

0040D7D6 mov esp,ebp

0040D7D8 pop ebp

0040D7D9 ret