

Jan 26, 18 11:11

fact_VC_x86.txt

Page 1/4

```
1 ///////////////////////////////////////////////////////////////////
2 //      fact.c program to test a recursive call in C
3 //      M. Mizuno (c) 1995, 2004, 2005
4 //      modified for Learning Tree course 223P
5 //      NOTE: In gcc for H-8, default for char is "unsigned"
6 ///////////////////////////////////////////////////////////////////
7 /* Draw memory image of execution of the following program */
8     int ans;
9
10    void fact(int i)
11    {
12        /* what will happen if x is declared as internal static */
13        int x;
14
15        x = i - 1;
16        if ((i == 0) || (x == 0)) ans = 1;
17        else {
18            fact(x);
19            ans = (x + 1) * ans;
20        }
21    }
22
23    int main(void)
24    {
25        fact(3);
26        printf("asnsver = %d\n", ans);
27        return 0;
28    }
29
```

Jan 26, 18 11:11

fact_VC_x86.txt

Page 2/4

30

Jan 26, 18 11:11 **fact_VC_x86.txt** Page 3/4

```

31 ; Listing generated by Microsoft (R) Optimizing Compiler Version 15.00.21022.08
32
33 TITLE c:\LearningTree\223P\Programs\memory_test\Pentium\fact.c
34 .686P
35 .XMM
36 include listing.inc
37 .model flat
38
39 INCLUDELIB LIBCMT
40 INCLUDELIB OLDNAMES
41
42 _DATA SEGMENT
43 COMM _ans:DWORD
44 $SG742 DB 'answer = %d', 0aH, 00H
45 _DATA ENDS
46 PUBLIC _fact
47 ; Function compile flags: /Odtp
48 ; File c:\learningtree\223p\programs\memory_test\pentium\fact.c
49 _TEXT SEGMENT
50 _x$ = -4 ; size = 4
51 _i$ = 8 ; size = 4
52 _fact PROC
53
54 ; 11 : {
55
56 00000 55 push ebp
57 00001 8b ec mov ebp, esp
58 00003 51 push ecx
59
60 ; 12 : /* what will happen if x is declared as internal static */
61 ; 13 : int x;
62 ; 14 :
63 ; 15 : x = i - 1;
64
65 00004 8b 45 08 mov eax, DWORD PTR _i$[ebp]
66 00007 83 e8 01 sub eax, 1
67 0000a 89 45 fc mov DWORD PTR _x$[ebp], eax
68
69 ; 16 : if ((i == 0) || (x == 0)) ans = 1;
70
71 0000d 83 7d 08 00 cmp DWORD PTR _i$[ebp], 0
72 00011 74 06 je SHORT $LN2@fact
73 00013 83 7d fc 00 cmp DWORD PTR _x$[ebp], 0
74 00017 75 0c jne SHORT $LN3@fact
75 $LN2@fact:
76 00019 c7 05 00 00 00 mov DWORD PTR _ans, 1
77 00 01 00 00 00
78
79 ; 17 : else {
80
81 00023 eb 1f jmp SHORT $LN4@fact
82 $LN3@fact:
83
84 ; 18 : fact(x);
85
86 00025 8b 4d fc mov ecx, DWORD PTR _x$[ebp]
87 00028 51 push ecx
88 00029 e8 00 00 00 00 call _fact
89 0002e 83 c4 04 add esp, 4
90
91 ; 19 : ans = (x + 1) * ans;
92
93 00031 8b 55 fc mov edx, DWORD PTR _x$[ebp]
94 00034 83 c2 01 add edx, 1
95 00037 0f af 15 00 00 imul edx, DWORD PTR _ans
96 00 00
97 0003e 89 15 00 00 00 mov DWORD PTR _ans, edx
98 00
99 $LN4@fact:
100
101 ; 20 : }
102 ; 21 :
103

```

Jan 26, 18 11:11 **fact_VC_x86.txt** Page 4/4

```

104 00044 8b e5 mov esp, ebp
105 00046 5d pop ebp
106 00047 c3 ret 0
107 _fact ENDP
108 _TEXT ENDS
109 PUBLIC _main
110 EXTRN _printf:PROC
111 ; Function compile flags: /Odtp
112 _TEXT SEGMENT
113 _main PROC
114
115 ; 24 : {
116
117 00050 55 push ebp
118 00051 8b ec mov ebp, esp
119
120 ; 25 : fact(3);
121
122 00053 6a 03 push 3
123 00055 e8 00 00 00 00 call _fact
124 0005a 83 c4 04 add esp, 4
125
126 ; 26 : printf("answer = %d\n", ans);
127
128 0005d a1 00 00 00 00 mov eax, DWORD PTR _ans
129 00062 50 push eax
130 00063 68 00 00 00 00 push OFFSET $SG742
131 00068 e8 00 00 00 00 call _printf
132 0006d 83 c4 08 add esp, 8
133
134 ; 27 : return 0;
135
136 00070 33 c0 xor eax, eax
137
138 ; 28 : }
139
140 00072 5d pop ebp
141 00073 c3 ret 0
142 _main ENDP
143 _TEXT ENDS
144 END

```