

Feb 07, 18 9:10

fact_VC_x86_64.txt

Page 1/5

```

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 #include <stdio.h>
8 /* Draw memory image of execution of the following program */
9 int ans;
10
11 void fact(int i)
12 {
13     /* what will happen if x is declared as internal static */
14     int x;
15
16     x = i - 1;
17     if ((i == 0) || (x == 0)) ans = 1;
18     else {
19         fact(x);
20         ans = (x + 1) * ans;
21     }
22 }
23
24 int main(void)
25 {
26     fact(3);
27     printf("asnsver = %d\n", ans);
28     return 0;
29 }
30

```

Feb 07, 18 9:10

fact_VC_x86_64.txt

Page 2/5

```

31 ; Listing generated by Microsoft (R) Optimizing Compiler Version 19.12.25834.0
32
33 include listing.inc
34
35 INCLUDELIB LIBCMT
36 INCLUDELIB OLDNAMES
37
38 _DATA SEGMENT
39 COMM ans:DWORD
40 _DATA ENDS
41 _DATA SEGMENT
42 $SG7080 DB 'asnsver = %d', 0aH, 00H
43 _DATA ENDS
44 PUBLIC __local_stdio_printf_options
45 PUBLIC _vfprintf_l
46 PUBLIC printf
47 PUBLIC fact
48 PUBLIC main
49 EXTRN __acrt_iob_func:PROC
50 EXTRN __stdio_common_vfprintf:PROC
51 _DATA SEGMENT
52 COMM ?_OptionsStorage@?1??__local_stdio_printf_options@@9@9:QWORD
53 ; '_local_stdio_printf_options'::'2'::_OptionsStorage@?1??__local_stdio_printf_options@@9@9:QWORD
54 _DATA ENDS
55 ; COMDAT pdata
56 pdata SEGMENT
57 $pdata$_vfprintf_l DD imagerel $LN3
58 DD imagerel $LN3+67
59 DD imagerel $unwind$_vfprintf_l
60 pdata ENDS
61 ; COMDAT pdata
62 pdata SEGMENT
63 $pdata$printf DD imagerel $LN3
64 DD imagerel $LN3+87
65 DD imagerel $unwind$printf
66 pdata ENDS
67 pdata SEGMENT
68 $pdata$fact DD imagerel $LN6
69 DD imagerel $LN6+77
70 DD imagerel $unwind$fact
71 $pdata$main DD imagerel $LN3
72 DD imagerel $LN3+39
73 DD imagerel $unwind$main
74 pdata ENDS
75 xdata SEGMENT
76 $unwind$fact DD 010801H
77 DD 06208H
78 $unwind$main DD 010401H
79 DD 04204H
80 xdata ENDS
81 ; COMDAT xdata
82 xdata SEGMENT
83 $unwind$printf DD 011801H
84 DD 06218H
85 xdata ENDS
86 ; COMDAT xdata
87 xdata SEGMENT
88 $unwind$_vfprintf_l DD 011801H
89 DD 06218H
90 xdata ENDS
91 ; Function compile flags: /Odtp
92 ; File c:\k-state\cis450\programs\memorytest\vc_x86_64\fact.c
93 _TEXT SEGMENT
94 main PROC
95 ; 25 : {
96
97 $LN3:
98 00000 48 83 ec 28 sub rsp, 40 ; 00000028H
99
100 ; 26 : fact(3);
101

```

Feb 07, 18 9:10	fact_VC_x86_64.txt	Page 3/5
102	00004 b9 03 00 00 00 mov ecx, 3	
103	00009 e8 00 00 00 00 call fact	
104		
105	; 27 : printf("asnsver = %d\n", ans);	
106		
107	0000e 8b 15 00 00 00 mov edx, DWORD PTR ans	
108	00	
109	00014 48 8d 0d 00 00 lea rcx, OFFSET FLAT:\$SG7080	
110	00 00	
111	0001b e8 00 00 00 00 call printf	
112		
113	; 28 : return 0;	
114		
115	00020 33 c0 xor eax, eax	
116		
117	; 29 : }	
118		
119	00022 48 83 c4 28 add rsp, 40 ; 00000028H	
120	00026 c3 ret 0	
121	main ENDP	
122	_TEXT ENDS	
123	; Function compile flags: /Odtp	
124	; File c:\k-state\cis450\programs\memorytest\vc_x86_64\fact.c	
125	_TEXT SEGMENT	
126	x\$ = 32	
127	i\$ = 64	
128	fact PROC	
129		
130	; 12 : {	
131		
132	\$LN6:	
133	00000 89 4c 24 08 mov DWORD PTR [rsp+8], ecx	
134	00004 48 83 ec 38 sub rsp, 56 ; 00000038H	
135		
136	; 13 : /* what will happen if x is declared as internal static */	
137	; 14 : int x;	
138	; 15 :	
139	; 16 : x = i - 1;	
140		
141	00008 8b 44 24 40 mov eax, DWORD PTR i\$[rsp]	
142	0000c ff c8 dec eax	
143	0000e 89 44 24 20 mov DWORD PTR x\$[rsp], eax	
144		
145	; 17 : if ((i == 0) (x == 0)) ans = 1;	
146		
147	00012 83 7c 24 40 00 cmp DWORD PTR i\$[rsp], 0	
148	00017 74 07 je SHORT \$LN4@fact	
149	00019 83 7c 24 20 00 cmp DWORD PTR x\$[rsp], 0	
150	0001e 75 0c jne SHORT \$LN2@fact	
151	\$LN4@fact:	
152	00020 c7 05 00 00 00 mov DWORD PTR ans, 1	
153	00 01 00 00 00	
154	0002a eb 1c jmp SHORT \$LN3@fact	
155	\$LN2@fact:	
156		
157	; 18 : else {	
158	; 19 : fact(x);	
159		
160	0002c 8b 4c 24 20 mov ecx, DWORD PTR x\$[rsp]	
161	00030 e8 00 00 00 00 call fact	
162		
163	; 20 : ans = (x + 1) * ans;	
164		
165	00035 8b 44 24 20 mov eax, DWORD PTR x\$[rsp]	
166	00039 ff c0 inc eax	
167	0003b 0f af 05 00 00 imul eax, DWORD PTR ans	
168	00 00	
169	00042 89 05 00 00 00 mov DWORD PTR ans, eax	
170	00	
171	\$LN3@fact:	
172		
173	; 21 : }	
174	; 22 : }	

Feb 07, 18 9:10	fact_VC_x86_64.txt	Page 4/5
175		
176	00048 48 83 c4 38 add rsp, 56 ; 00000038H	
177	0004c c3 ret 0	
178	fact ENDP	
179	_TEXT ENDS	
180	; Function compile flags: /Odtp	
181	; File c:\program files (x86)\windows kits\10\include\10.0.16299.0\ucrt\stdio.h	
182	; COMDAT printf	
183	_TEXT SEGMENT	
184	_Result\$ = 32	
185	_ArgList\$ = 40	
186	_Format\$ = 64	
187	printf PROC ; COMDAT	
188		
189	; 952 : {	
190		
191	\$LN3:	
192	00000 48 89 4c 24 08 mov QWORD PTR [rsp+8], rcx	
193	00005 48 89 54 24 10 mov QWORD PTR [rsp+16], rdx	
194	0000a 4c 89 44 24 18 mov QWORD PTR [rsp+24], r8	
195	0000f 4c 89 4c 24 20 mov QWORD PTR [rsp+32], r9	
196	00014 48 83 ec 38 sub rsp, 56 ; 00000038H	
197		
198	; 953 : int _Result;	
199	; 954 : va_list _ArgList;	
200	; 955 : __crt_va_start(_ArgList, _Format);	
201		
202	00018 48 8d 44 24 48 lea rax, QWORD PTR _Format\$[rsp+8]	
203	0001d 48 89 44 24 28 mov QWORD PTR _ArgList\$[rsp], rax	
204		
205	; 956 : _Result = _vfprintf_l(stdout, _Format, NULL, _ArgList);	
206		
207	00022 b9 01 00 00 00 mov ecx, 1	
208	00027 e8 00 00 00 00 call __acrt_iob_func	
209	0002c 4c 8b 4c 24 28 mov r9, QWORD PTR _ArgList\$[rsp]	
210	00031 45 33 c0 xor r8d, r8d	
211	00034 48 8b 54 24 40 mov rdx, QWORD PTR _Format\$[rsp]	
212	00039 48 8b c8 mov rcx, rax	
213	0003c e8 00 00 00 00 call _vfprintf_l	
214	00041 89 44 24 20 mov DWORD PTR _Result\$[rsp], eax	
215		
216	; 957 : __crt_va_end(_ArgList);	
217		
218	00045 48 c7 44 24 28	
219	00 00 00 00 mov QWORD PTR _ArgList\$[rsp], 0	
220		
221	; 958 : return _Result;	
222		
223	0004e 8b 44 24 20 mov eax, DWORD PTR _Result\$[rsp]	
224		
225	; 959 : }	
226		
227	00052 48 83 c4 38 add rsp, 56 ; 00000038H	
228	00056 c3 ret 0	
229	printf ENDP	
230	_TEXT ENDS	
231	; Function compile flags: /Odtp	
232	; File c:\program files (x86)\windows kits\10\include\10.0.16299.0\ucrt\stdio.h	
233	; COMDAT _vfprintf_l	
234	_TEXT SEGMENT	
235	_Stream\$ = 64	
236	_Format\$ = 72	
237	_Locale\$ = 80	
238	_ArgList\$ = 88	
239	_vfprintf_l PROC ; COMDAT	
240		
241	; 640 : {	
242		
243	\$LN3:	
244	00000 4c 89 4c 24 20 mov QWORD PTR [rsp+32], r9	
245	00005 4c 89 44 24 18 mov QWORD PTR [rsp+24], r8	
246	0000a 48 89 54 24 10 mov QWORD PTR [rsp+16], rdx	
247	0000f 48 89 4c 24 08 mov QWORD PTR [rsp+8], rcx	

Feb 07, 18 9:10

fact_VC_x86_64.txt

Page 5/5

```

248 00014 48 83 ec 38      sub     rsp, 56                ; 00000038H
249
250 ; 641 :      return __stdio_common_vfprintf(_CRT_INTERNAL_LOCAL_PRINTF_OPTIO
NS, _Stream, _Format, _Locale, _ArgList);
251
252 00018 e8 00 00 00 00     call    __local_stdio_printf_options
253 0001d 48 8b 4c 24 58     mov     rcx, QWORD PTR _ArgList$[rsp]
254 00022 48 89 4c 24 20     mov     QWORD PTR [rsp+32], rcx
255 00027 4c 8b 4c 24 50     mov     r9, QWORD PTR _Locale$[rsp]
256 0002c 4c 8b 44 24 48     mov     r8, QWORD PTR _Format$[rsp]
257 00031 48 8b 54 24 40     mov     rdx, QWORD PTR _Stream$[rsp]
258 00036 48 8b 08           mov     rcx, QWORD PTR [rax]
259 00039 e8 00 00 00 00     call    __stdio_common_vfprintf
260
261 ; 642 :      }
262
263 0003e 48 83 c4 38      add     rsp, 56                ; 00000038H
264 00042 c3              ret     0
265 __vfprintf_l ENDP
266 _TEXT    ENDS
267 ; Function compile flags: /Odtp
268 ; File c:\program files (x86)\windows kits\10\include\10.0.16299.0\ucrt\corecrt_
stdio_config.h
269 ;      COMDAT __local_stdio_printf_options
270 _TEXT    SEGMENT
271 __local_stdio_printf_options PROC                ; COMDAT
272
273 ; 86 :      static unsigned __int64 _OptionsStorage;
274 ; 87 :      return &_OptionsStorage;
275
276 00000 48 8d 05 00 00     lea     rax, OFFSET FLAT:??_OptionsStorage@?l??__local_s
277 00 00                tdio_printf_options@@9@9 ; '__local_stdio_printf_options'::'2'::_OptionsStorage
278
279 ; 88 :      }
280
281 00007 c3              ret     0
282 __local_stdio_printf_options ENDP
283 _TEXT    ENDS
284 END

```