

Jan 26, 18 11:01

var_VC_x86.txt

Page 1/6

```

1 ///////////////////////////////////////////////////////////////////
2 //      var.c program to test memory allocation in C
3 //      M. Mizuno (c) 1995, 2004, 2005
4 //      modified for Learning Tree course 223P
5 //
6 // to compile var.c on Pentium,
7 // 1. execute Visual Studio .NET2003 (~2017) x86 Native Tools Command Prompt
8 // 2. go to the directory which contains var.c
9 // 3. issue cl /Od /Facs var.c
10 //      /Od: disable optimization
11 //      /Facs: generate a listing file with source code and machine code
12 ///////////////////////////////////////////////////////////////////
13 long test(unsigned int ui, int i, short s, unsigned short us,
14           char c, unsigned char uc, long l, unsigned long ul,
15           int x, short y);
16
17 char ret;
18 int x=100;
19 static int si;
20 static int sj = 23;
21
22 int main(int argc, char **argv, char **envp)
23 {
24     unsigned char uc;
25     static short y = 99;
26     short s;
27     char c;
28     unsigned short us;
29     static int i;
30     unsigned int ui;
31     long l;
32     unsigned long ul;
33
34     if (i < 0) {
35         ui = us + s - c;
36     }
37     else {
38         ul = si - sj * 2;
39     }
40
41     while (sj > 0) {
42         uc = y - 3;
43         sj++;
44     }
45
46     ret = test(ui, i, s, us, c, uc, l, ul, x, y);
47
48     return 0;
49 }
50
51 long test(unsigned int ui, int i, short s, unsigned short us,
52           char c, unsigned char uc, long l, unsigned long ul,
53           int x, short y)
54 {
55     char c1;
56     int i1;
57     char c2;
58
59     ui = 1;
60     i = 2;
61     s = 3;
62     us = 4;
63     c = 5;
64     uc = 6;
65     l = 7;
66     ul = 8;
67     x = 9;
68     y = 10;
69     c1 = 11;
70     c2 = 12;
71     i1 = 13;
72     return ui * 2 + 1;
73 }

```

Jan 26, 18 11:01

var_VC_x86.txt

Page 2/6

74

Jan 26, 18 11:01 **var_VC_x86.txt** Page 3/6

```

75 ; Listing generated by Microsoft (R) Optimizing Compiler Version 14.00.50727.42
76
77 TITLE C:\LearningTree\223P\Programs\memory_test\Pentium\var.c
78 .686P
79 .XMM
80 include listing.inc
81 .model flat
82
83 INCLUDELIB LIBCMT
84 INCLUDELIB OLDNAMES
85
86 PUBLIC _x
87 _DATA SEGMENT
88 COMM _ret:BYTE
89 _x DD 064H
90 _sj DD 017H
91 ?y@?l??main@@9@9 DW 063H ; 'main'::'2'::y
92 _DATA ENDS
93 PUBLIC _test
94 PUBLIC _main
95 _BSS SEGMENT
96 _si DD 01H DUP (?)
97 ?i@?l??main@@9@9 DD 01H DUP (?) ; 'main'::'2'::i
98 ; Function compile flags: /Odtp
99 ; File c:\learningtree\223p\programs\memory_test\pentium\var.c
100 _BSS ENDS
101 _TEXT SEGMENT
102 _c$ = -25 ; size = 1
103 _us$ = -24 ; size = 2
104 _l$ = -20 ; size = 4
105 _ul$ = -16 ; size = 4
106 _uc$ = -9 ; size = 1
107 _ui$ = -8 ; size = 4
108 _s$ = -4 ; size = 2
109 _argc$ = 8 ; size = 4
110 _argv$ = 12 ; size = 4
111 _envp$ = 16 ; size = 4
112 _main PROC
113
114 ; 23 : {
115
116 00000 55 push ebp
117 00001 8b ec mov ebp, esp
118 00003 83 ec 1c sub esp, 28 ; 0000001cH
119
120 ; 24 : unsigned char uc;
121 ; 25 : static short y = 99;
122 ; 26 : short s;
123 ; 27 : char c;
124 ; 28 : unsigned short us;
125 ; 29 : static int i;
126 ; 30 : unsigned int ui;
127 ; 31 : long l;
128 ; 32 : unsigned long ul;
129 ; 33 :
130 ; 34 : if (i < 0) {
131
132 00006 83 3d 00 00 00 cmp DWORD PTR ?i@?l??main@@9@9, 0
133 00 00 jge SHORT $LN4@main
134 0000d 7d 15
135
136 ; 35 : ui = us + s - c;
137
138 0000f 0f b7 45 e8 movzx eax, WORD PTR _us$[ebp]
139 00013 0f bf 4d fc movsx ecx, WORD PTR _s$[ebp]
140 00017 03 c1 add eax, ecx
141 00019 0f be 55 e7 movsx edx, BYTE PTR _c$[ebp]
142 0001d 2b c2 sub eax, edx
143 0001f 89 45 f8 mov DWORD PTR _ui$[ebp], eax
144
145 ; 36 : }
146 ; 37 : else {
147

```

Jan 26, 18 11:01 **var_VC_x86.txt** Page 4/6

```

148 00022 eb 12 jmp SHORT $LN2@main
149 $LN4@main:
150
151 ; 38 : ul = si - sj * 2;
152
153 00024 a1 00 00 00 00 mov eax, DWORD PTR _sj
154 00029 d1 e0 shl eax, 1
155 0002b 8b 0d 00 00 00 mov ecx, DWORD PTR _si
156 00 00
157 00031 2b c8 sub ecx, eax
158 00033 89 4d f0 mov DWORD PTR _ul$[ebp], ecx
159 $LN2@main:
160
161 ; 39 : }
162 ; 40 :
163 ; 41 : while (sj > 0) {
164
165 00036 83 3d 00 00 00 cmp DWORD PTR _sj, 0
166 00 00
167 0003d 7e 1c jle SHORT $LN1@main
168
169 ; 42 : uc = y - 3;
170
171 0003f 0f bf 15 00 00 movsx edx, WORD PTR ?y@?l??main@@9@9
172 00 00
173 00046 83 ea 03 sub edx, 3
174 00049 88 55 f7 mov BYTE PTR _uc$[ebp], dl
175
176 ; 43 : sj++;
177
178 0004c a1 00 00 00 00 mov eax, DWORD PTR _sj
179 00051 83 c0 01 add eax, 1
180 00054 a3 00 00 00 00 mov DWORD PTR _sj, eax
181
182 ; 44 : }
183
184 00059 eb db jmp SHORT $LN2@main
185 $LN1@main:
186
187 ; 45 :
188 ; 46 : ret = test(ui, i, s, us, c, uc, l, ul, x, y);
189
190 0005b 0f bf 0d 00 00 movsx ecx, WORD PTR ?y@?l??main@@9@9
191 00 00 push ecx
192 00062 51
193 00063 8b 15 00 00 00 mov edx, DWORD PTR _x
194 00 00 push edx
195 00069 52 push eax, DWORD PTR _ul$[ebp]
196 0006a 8b 45 f0 mov eax, DWORD PTR _l$[ebp]
197 0006d 50 push ecx, DWORD PTR _l$[ebp]
198 0006e 8b 4d ec mov ecx, DWORD PTR _l$[ebp]
199 00071 51 push ecx
200 00072 0f b6 55 f7 movzx edx, BYTE PTR _uc$[ebp]
201 00076 52 push edx
202 00077 0f be 45 e7 movsx eax, BYTE PTR _c$[ebp]
203 0007b 50 push eax
204 0007c 0f b7 4d e8 movzx ecx, WORD PTR _us$[ebp]
205 00080 51 push ecx
206 00081 0f bf 55 fc movsx edx, WORD PTR _s$[ebp]
207 00085 52 push edx
208 00086 a1 00 00 00 00 mov eax, DWORD PTR ?i@?l??main@@9@9
209 0008b 50 push eax
210 0008c 8b 4d f8 mov ecx, DWORD PTR _ui$[ebp]
211 0008f 51 push ecx
212 00090 e8 00 00 00 00 call _test
213 00095 83 c4 28 add esp, 40 ; 00000028H
214 00098 a2 00 00 00 00 mov BYTE PTR _ret, al
215
216 ; 47 :
217 ; 48 : return 0;
218
219 0009d 33 c0 xor eax, eax
220

```

Jan 26, 18 11:01	var_VC_x86.txt	Page 5/6
221	; 49 : }	
222		
223	0009f 8b e5	mov esp, ebp
224	000a1 5d	pop ebp
225	000a2 c3	ret 0
226	_main ENDP	
227	; Function compile flags: /Odtp	
228	_c2\$ = -6	; size = 1
229	_c1\$ = -5	; size = 1
230	_i1\$ = -4	; size = 4
231	_ui\$ = 8	; size = 4
232	_i\$ = 12	; size = 4
233	_s\$ = 16	; size = 2
234	_us\$ = 20	; size = 2
235	_c\$ = 24	; size = 1
236	_uc\$ = 28	; size = 1
237	_l\$ = 32	; size = 4
238	_ul\$ = 36	; size = 4
239	_x\$ = 40	; size = 4
240	_y\$ = 44	; size = 2
241	_test PROC	
242		
243	; 54 : {	
244		
245	000b0 55	push ebp
246	000b1 8b ec	mov ebp, esp
247	000b3 83 ec 08	sub esp, 8
248		
249	; 55 : char c1;	
250	; 56 : int i1;	
251	; 57 : char c2;	
252	; 58 :	
253	; 59 : ui = 1;	
254		
255	000b6 c7 45 08 01 00	
256	00 00	mov DWORD PTR _ui\$[ebp], 1
257		
258	; 60 : i = 2;	
259		
260	000bd c7 45 0c 02 00	
261	00 00	mov DWORD PTR _i\$[ebp], 2
262		
263	; 61 : s = 3;	
264		
265	000c4 66 c7 45 10 03	
266	00	mov WORD PTR _s\$[ebp], 3
267		
268	; 62 : us = 4;	
269		
270	000ca 66 c7 45 14 04	
271	00	mov WORD PTR _us\$[ebp], 4
272		
273	; 63 : c = 5;	
274		
275	000d0 c6 45 18 05	mov BYTE PTR _c\$[ebp], 5
276		
277	; 64 : uc = 6;	
278		
279	000d4 c6 45 1c 06	mov BYTE PTR _uc\$[ebp], 6
280		
281	; 65 : l = 7;	
282		
283	000d8 c7 45 20 07 00	
284	00 00	mov DWORD PTR _l\$[ebp], 7
285		
286	; 66 : ul = 8;	
287		
288	000df c7 45 24 08 00	
289	00 00	mov DWORD PTR _ul\$[ebp], 8
290		
291	; 67 : x = 9;	
292		
293	000e6 c7 45 28 09 00	

Jan 26, 18 11:01	var_VC_x86.txt	Page 6/6
294	00 00	mov DWORD PTR _x\$[ebp], 9
295		
296	; 68 : y = 10;	
297		
298	000ed 66 c7 45 2c 0a	
299	00	mov WORD PTR _y\$[ebp], 10 ; 0000000aH
300		
301	; 69 : c1 = 11;	
302		
303	000f3 c6 45 fb 0b	mov BYTE PTR _c1\$[ebp], 11 ; 0000000bH
304		
305	; 70 : c2 = 12;	
306		
307	000f7 c6 45 fa 0c	mov BYTE PTR _c2\$[ebp], 12 ; 0000000cH
308		
309	; 71 : i1 = 13;	
310		
311	000fb c7 45 fc 0d 00	
312	00 00	mov DWORD PTR _i1\$[ebp], 13 ; 0000000dH
313		
314	; 72 : return ui * 2 + 1;	
315		
316	00102 8b 45 08	mov eax, DWORD PTR _ui\$[ebp]
317	00105 8b 4d 20	mov ecx, DWORD PTR _l\$[ebp]
318	00108 8d 04 41	lea eax, DWORD PTR [ecx+eax*2]
319		
320	; 73 : }	
321		
322	0010b 8b e5	mov esp, ebp
323	0010d 5d	pop ebp
324	0010e c3	ret 0
325	_test ENDP	
326	_TEXT ENDS	
327	END	