

Feb 07, 18 9:09

var_VC_x86_64.c

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 .NET 2003 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 }

```

Feb 07, 18 9:09

var_VC_x86_64.c

Page 2/6

74

Feb 07, 18 9:09 **var_VC_x86_64.c** Page 3/6

```

75 ; Listing generated by Microsoft (R) Optimizing Compiler Version 19.12.25834.0
76
77 include listing.inc
78
79 INCLUDELIB LIBCMT
80 INCLUDELIB OLDNAMES
81
82 PUBLIC x
83 _DATA SEGMENT
84 COMM ret:BYTE
85 _DATA ENDS
86 _DATA SEGMENT
87 x DD 064H
88 sj DD 017H
89 ?y@?l??main@@9@9 DW 063H ; 'main'::'2'::y
90 _DATA ENDS
91 PUBLIC test
92 PUBLIC main
93 _BSS SEGMENT
94 si DD 01H DUP (?)
95 ?i@?l??main@@9@9 DD 01H DUP (?) ; 'main'::'2'::i
96 _BSS ENDS
97 pdata SEGMENT
98 $pdata$test DD imagerel $LN3
99 DD imagerel $LN3+137
100 DD imagerel $unwind$test
101 $pdata$main DD imagerel $LN7
102 DD imagerel $LN7+209
103 DD imagerel $unwind$main
104 pdata ENDS
105 xdata SEGMENT
106 $unwind$test DD 011801H
107 DD 02218H
108 $unwind$main DD 011201H
109 DD 0e212H
110 xdata ENDS
111 ; Function compile flags: /Odtp
112 ; File c:\k-state\cis450\programs\memorytest\vc_x86_64\var.c
113 _TEXT SEGMENT
114 uc$ = 80
115 c$ = 81
116 us$ = 84
117 s$ = 88
118 ul$ = 92
119 ui$ = 96
120 l$ = 100
121 argc$ = 128
122 argv$ = 136
123 envp$ = 144
124 main PROC
125
126 ; 23 : {
127
128 $LN7:
129 00000 4c 89 44 24 18 mov QWORD PTR [rsp+24], r8
130 00005 48 89 54 24 10 mov QWORD PTR [rsp+16], rdx
131 0000a 89 4c 24 08 mov DWORD PTR [rsp+8], ecx
132 0000e 48 83 ec 78 sub rsp, 120 ; 00000078H
133
134 ; 24 : unsigned char uc;
135 ; 25 : static short y = 99;
136 ; 26 : short s;
137 ; 27 : char c;
138 ; 28 : unsigned short us;
139 ; 29 : static int i;
140 ; 30 : unsigned int ui;
141 ; 31 : long l;
142 ; 32 : unsigned long ul;
143 ; 33 :
144 ; 34 : if (i < 0) {
145
146 00012 83 3d 00 00 00 cmp DWORD PTR ?i@?l??main@@9@9, 0
147 00 00

```

Feb 07, 18 9:09 **var_VC_x86_64.c** Page 4/6

```

148 00019 7d 19 jge SHORT $LN4@main
149
150 ; 35 : ui = us + s - c;
151
152 0001b 0f b7 44 24 54 movzx eax, WORD PTR us[rs]
153 00020 0f bf 4c 24 58 movsx ecx, WORD PTR s[rs]
154 00025 03 c1 add eax, ecx
155 00027 0f be 4c 24 51 movsx ecx, BYTE PTR c[rs]
156 0002c 2b c1 sub eax, ecx
157 0002e 89 44 24 60 mov DWORD PTR ui[rs], eax
158
159 ; 36 : }
160
161 00032 eb 16 jmp SHORT $LN5@main
162 $LN4@main:
163
164 ; 37 : else {
165 ; 38 : ul = si - sj * 2;
166
167 00034 8b 05 00 00 00 mov eax, DWORD PTR sj
168 00 00 shl eax, 1
169 0003a d1 e0 shl eax, 1
170 0003c 8b 0d 00 00 00 mov ecx, DWORD PTR si
171 00 00 sub ecx, eax
172 00042 2b c8 mov eax, ecx
173 00044 8b c1 mov DWORD PTR ul[rs], eax
174 00046 89 44 24 5c
175 $LN5@main:
176 $LN2@main:
177
178 ; 39 : }
179 ; 40 :
180 ; 41 : while (sj > 0) {
181
182 0004a 83 3d 00 00 00 cmp DWORD PTR sj, 0
183 00 00 jle SHORT $LN3@main
184 00051 7e 1e
185
186 ; 42 : uc = y - 3;
187
188 00053 0f bf 05 00 00 movsx eax, WORD PTR ?y@?l??main@@9@9
189 00 00 sub eax, 3
190 0005a 83 e8 03 mov BYTE PTR uc[rs], al
191 0005d 88 44 24 50
192
193 ; 43 : sj++;
194
195 00061 8b 05 00 00 00 mov eax, DWORD PTR sj
196 00 00 inc eax
197 00067 ff c0 mov DWORD PTR sj, eax
198 00069 89 05 00 00 00
199 00 00
200
201 ; 44 : }
202
203 0006f eb d9 jmp SHORT $LN2@main
204 $LN3@main:
205
206 ; 45 :
207 ; 46 : ret = test(ui, i, s, us, c, uc, l, ul, x, y);
208
209 00071 0f b7 05 00 00 movzx eax, WORD PTR ?y@?l??main@@9@9
210 00 00 mov WORD PTR [rsp+72], ax
211 00078 66 89 44 24 48
212 0007d 8b 05 00 00 00 mov eax, DWORD PTR x
213 00 00 mov DWORD PTR [rsp+64], eax
214 00083 89 44 24 40 mov eax, DWORD PTR ul[rs]
215 00087 8b 44 24 5c mov DWORD PTR [rsp+56], eax
216 0008b 89 44 24 38 mov eax, DWORD PTR l[rs]
217 0008f 8b 44 24 64 mov DWORD PTR [rsp+48], eax
218 00093 89 44 24 30 mov eax, BYTE PTR uc[rs]
219 00097 0f b6 44 24 50 movzx eax, BYTE PTR [rsp+40], al
220 0009c 88 44 24 28

```

Feb 07, 18 9:09	var_VC_x86_64.c	Page 5/6
221	000a0 0f b6 44 24 51	movzx eax, BYTE PTR c\$[rsp]
222	000a5 88 44 24 20	mov BYTE PTR [rsp+32], al
223	000a9 44 0f b7 4c 24	
224	54	movzx r9d, WORD PTR us\$[rsp]
225	000af 44 0f b7 44 24	
226	58	movzx r8d, WORD PTR s\$[rsp]
227	000b5 8b 15 00 00 00	
228	00	mov edx, DWORD PTR ?i@?l??main@@9@9
229	000bb 8b 4c 24 60	mov ecx, DWORD PTR ui\$[rsp]
230	000bf e8 00 00 00 00	call test
231	000c4 88 05 00 00 00	
232	00	mov BYTE PTR ret, al
233		
234	; 47 :	
235	; 48 : return 0;	
236		
237	000ca 33 c0	xor eax, eax
238		
239	; 49 : }	
240		
241	000cc 48 83 c4 78	add rsp, 120 ; 00000078H
242	000d0 c3	ret 0
243	main ENDP	
244	_TEXT ENDS	
245	; Function compile flags: /Odtp	
246	; File c:\k-state\cis450\programs\memorytest\vc_x86_64\var.c	
247	_TEXT SEGMENT	
248	c1\$ = 0	
249	c2\$ = 1	
250	il\$ = 4	
251	ui\$ = 32	
252	i\$ = 40	
253	s\$ = 48	
254	us\$ = 56	
255	c\$ = 64	
256	uc\$ = 72	
257	l\$ = 80	
258	ul\$ = 88	
259	x\$ = 96	
260	y\$ = 104	
261	test PROC	
262		
263	; 54 : {	
264		
265	\$LN3:	
266	00000 66 44 89 4c 24	mov WORD PTR [rsp+32], r9w
267	20	
268	00006 66 44 89 44 24	mov WORD PTR [rsp+24], r8w
269	18	
270	0000c 89 54 24 10	mov DWORD PTR [rsp+16], edx
271	00010 89 4c 24 08	mov DWORD PTR [rsp+8], ecx
272	00014 48 83 ec 18	sub rsp, 24
273		
274	; 55 : char c1;	
275	; 56 : int il;	
276	; 57 : char c2;	
277	; 58 :	
278	; 59 : ui = 1;	
279		
280	00018 c7 44 24 20 01	
281	00 00 00	mov DWORD PTR ui\$[rsp], 1
282		
283	; 60 : i = 2;	
284		
285	00020 c7 44 24 28 02	
286	00 00 00	mov DWORD PTR i\$[rsp], 2
287		
288	; 61 : s = 3;	
289		
290	00028 b8 03 00 00 00	mov eax, 3
291	0002d 66 89 44 24 30	mov WORD PTR s\$[rsp], ax
292		
293	; 62 : us = 4;	

Feb 07, 18 9:09	var_VC_x86_64.c	Page 6/6
294		
295	00032 b8 04 00 00 00	mov eax, 4
296	00037 66 89 44 24 38	mov WORD PTR us\$[rsp], ax
297		
298	; 63 : c = 5;	
299		
300	0003c c6 44 24 40 05	mov BYTE PTR c\$[rsp], 5
301		
302	; 64 : uc = 6;	
303		
304	00041 c6 44 24 48 06	mov BYTE PTR uc\$[rsp], 6
305		
306	; 65 : l = 7;	
307		
308	00046 c7 44 24 50 07	
309	00 00 00	mov DWORD PTR l\$[rsp], 7
310		
311	; 66 : ul = 8;	
312		
313	0004e c7 44 24 58 08	
314	00 00 00	mov DWORD PTR ul\$[rsp], 8
315		
316	; 67 : x = 9;	
317		
318	00056 c7 44 24 60 09	
319	00 00 00	mov DWORD PTR x\$[rsp], 9
320		
321	; 68 : y = 10;	
322		
323	0005e b8 0a 00 00 00	mov eax, 10
324	00063 66 89 44 24 68	mov WORD PTR y\$[rsp], ax
325		
326	; 69 : c1 = 11;	
327		
328	00068 c6 04 24 0b	mov BYTE PTR c1\$[rsp], 11
329		
330	; 70 : c2 = 12;	
331		
332	0006c c6 44 24 01 0c	mov BYTE PTR c2\$[rsp], 12
333		
334	; 71 : il = 13;	
335		
336	00071 c7 44 24 04 0d	
337	00 00 00	mov DWORD PTR il\$[rsp], 13
338		
339	; 72 : return ui * 2 + 1;	
340		
341	00079 8b 44 24 50	mov eax, DWORD PTR l\$[rsp]
342	0007d 8b 4c 24 20	mov ecx, DWORD PTR ui\$[rsp]
343	00081 8d 04 48	lea eax, DWORD PTR [rax+rcx*2]
344		
345	; 73 : }	
346		
347	00084 48 83 c4 18	add rsp, 24
348	00088 c3	ret 0
349	test ENDP	
350	_TEXT ENDS	
351	END	