

```

1 ; Name      : list.asm
2 ;
3 ; Build     : aclocal && autoconf && automake --add-missing --foreign
4 ;           mkdir build
5 ;           cd build
6 ;           ../configure
7 ;           make
8 ;
9 ; Description : an example of glib 2.0 single linked lists
10 ;
11 ; Source     : https://github.com/steshaw/gtk-examples
12
13 bits 64
14
15 [list -]
16     extern g_slist_append
17     extern g_slist_free
18     extern g_slist_insert
19     extern g_slist_insert_sorted
20     extern g_slist_prepend
21     extern g_slist_reverse
22     extern g_slist_sort
23     extern g_slist_nth
24     extern g_slist_find
25     extern g_print
26     extern strcmp
27     extern exit
28 [list +]
29
30 section .rodata
31     names:
32     .fred:    db    "Fred",0
33     .susie:   db    "Susie",0
34     .frank:   db    "Frank",0
35     .wilma:   db    "Wilma",0
36     .mary:    db    "Mary",0
37     .joe:     db    "Joe",0
38     .mike:    db    "Mike",0
39     .tyler:   db    "Tyler",0
40     .amy:     db    "Amy",0
41     .zoe:     db    "Zoe",0
42
43     messages:
44     .original:    db    "original list",10,"-----",10,0
45     .reversinglist: db    10,"reversing list",10,"-----",10,0
46     .reversesorting: db    10,"reverse sorted list",10,"-----",10,0
47     .sorting:     db    10,"sorted list",10,"-----",10,0
48     .notfound:    db    "not found.",10,0
49     .found:       db    "found at %d.",10,0
50     .item:        db    "%s " ;don't separate this line with the next
51     .crlf:        db    10,0
52     .lookup:      db    "looking up %s : ",0
53
54 section .data
55
56     list:         dq    0 ;start of the list
57
58 section .text
59 global _start
60
61 _start:
62     ;add Fred, Susie, Frank and Wilma to the list
63     mov     rdi,[list]
64     mov     rsi,names.fred
65     call    g_slist_append
66     mov     [list],rax
67     mov     rdi,[list]
68     mov     rsi,names.susie
69     call    g_slist_append
70     mov     [list],rax
71     mov     rdi,[list]
72     mov     rsi,names.frank
73     call    g_slist_append

```

```

74     mov     [list],rax
75     mov     rdi,[list]
76     mov     rsi,names.wilma
77     call    g_slist_append
78     mov     [list],rax
79     ;insert Joe at the front of the list
80     mov     rdi,[list]
81     mov     rsi,names.joe
82     call    g_slist_prepend
83     mov     [list],rax
84     ;insert Mary after the second position
85     mov     rdi,[list]
86     mov     rsi,names.mary
87     mov     rdx,2
88     call    g_slist_insert
89     mov     [list],rax
90     ;add names but sorted
91     mov     rdi,[list]
92     mov     rsi,names.amy
93     mov     rdx,List.Sort
94     call    g_slist_insert_sorted
95     mov     [list],rax
96     mov     rdi,[list]
97     mov     rsi,names.zoe
98     mov     rdx,List.Sort
99     call    g_slist_insert_sorted
100    mov     [list],rax
101    ;Show the list so far
102    mov     rdi,messages.original
103    xor     rax,rax
104    call    g_print
105    mov     rdi,[list]
106    call    List.Display
107    mov     rdi,messages.crlf
108    xor     rax,rax
109    call    g_print
110    ;lookup some names in the list
111    mov     rdi,[list]
112    mov     rsi,names.mike
113    call    List.Lookup
114    mov     rdi,[list]
115    mov     rsi,names.mary
116    call    List.Lookup
117    mov     rdi,[list]
118    mov     rsi,names.fred
119    call    List.Lookup
120    mov     rdi,[list]
121    mov     rsi,names.tyler
122    call    List.Lookup
123    ;reverse the list
124    mov     rdi,messages.reversinglist
125    call    g_print
126    mov     rdi,[list]
127    call    g_slist_reverse
128    mov     [list],rax
129    ;show revered list
130    mov     rdi,[list]
131    call    List.Display
132    ;sort the list alphabetically
133    mov     rdi,messages.sorting
134    call    g_print
135    mov     rdi,[list]
136    mov     rsi,List.Sort
137    call    g_slist_sort
138    mov     [list],rax
139    ;show sorted list
140    mov     rdi,[list]
141    call    List.Display
142    ;reversesort the list
143    mov     rdi,messages.reversesorting
144    call    g_print
145    mov     rdi,[list]
146    mov     rsi,List.ReverseSort

```

```

147     call     g_slist_sort
148     mov      [list],rax
149     ;show sorted list
150     mov      rdi,[list]
151     call     List.Display
152     ;free the data
153     mov      rdi,[list]
154     call     g_slist_free
155     ;exit the program
156     xor      rdi,rdi
157     call     exit
158
159 List:
160 .Lookup:
161     ;int 3
162     ;rdi has pointer to list
163     ;rsi has pointer to name which we're looking for
164     push     rbp
165     mov      rbp, rsp
166     push     rdi
167     push     rsi
168     mov      rdi,messages.lookup
169     xor      rax,rax
170     call     g_print
171     pop      rsi
172     pop      rdi
173     call     g_slist_find
174     mov      rdi,messages.notfound
175     ;rax has either zero or the item pointer
176     and      rax,rax
177     jz       .print
178     mov      rsi,rax
179     mov      rdi,messages.found
180     xor      rax,rax
181 .print:
182     call     g_print
183     mov      rsp, rbp
184     pop      rbp
185     ret
186
187 .Display:
188     push     rbp
189     mov      rbp, rsp
190     sub      rsp,0x0
191     mov      r15,rdi
192     xor      rsi,rsi                ;index = 0
193 .repeat:
194     xor      rax,rax
195     mov      rdi,r15
196     call     g_slist_nth           ;get nth item in the list
197     and      rax,rax
198     jz       .done
199     push     rdi
200     push     rsi
201     mov      rsi,[rax]
202     mov      rdi,messages.item
203     xor      rax,rax
204     call     g_print
205     pop      rsi
206     pop      rdi
207     inc      esi
208     jmp      .repeat
209 .done:
210     mov      rsp, rbp
211     pop      rbp
212     ret
213
214 .ReverseSort:
215     ;reverse sorting is fairly easy now, just swap rdi and rsi
216     xor      rdi,rsi
217     xor      rsi,rdi
218     xor      rdi,rsi
219     ;and compare after the swap

```

```
220 .Sort:
221     ;compare strings in rdi and rsi, returning
222     ;-1 when string rdi comes before string rsi
223     ; 0 when both strings are equal
224     ; 1 when string in rsi comes after string rdi
225     call    strcmp
226     ret
227
```