```
; Name        : mnemonic.asm
;
; Build       : aclocal && autoconf && automake --add-missing --foreign
;               mkdir build
;               cd build
;               ../configure
;               make
;
; Description : mnemonic demo. once the window is on-screen, press ALT-B
;               the program must run in a terminal otherwise it seems dead
;
; Source      : http://zetcode.com/gui/gtk2/firstprograms/

bits 64

[list -]
    extern    gtk_init
    extern    gtk_window_new
    extern    gtk_window_set_title
    extern    gtk_window_set_default_size
    extern    gtk_container_set_border_width
    extern    gtk_button_new_with_mnemonic
    extern    gtk_widget_set_tooltip_text
    extern    gtk_alignment_new
    extern    gtk_container_add
    extern    gtk_widget_show_all
    extern    gtk_main_quit
    extern    g_signal_connect_data
    extern    gtk_main
    extern    exit
    extern    g_printf
[list +]

section .rodata
    szTitle:     db  "Mnemonic - press ALT-B",0
    szDestroy:   db  "destroy",0
    szClicked:   db  "clicked",0
    szCaption:   db  "_Button",0
    szMessage:   db  "Button clicked",10,0

section .data
    window:      dq  0
    button:      dq  0
    halign:      dq  0

section .text

global _start
_start:
    ;init gtk
    xor     rsi,rsi                         ;argv
    xor     rdi,rdi                         ;argc
    call    gtk_init
    ;the main window
    xor     rdi,rdi                         ;GTK_WINDOW_TOPLEVEL = 0 in RDI
    call    gtk_window_new
    mov     qword[window],rax
    ;set title
    mov     rdi,qword[window]
    mov     rsi,szTitle
    call    gtk_window_set_title
    ;set size
    mov     rdi,qword[window]
    mov     rsi,300                         ;width
    mov     rdx,200                         ;height
    call    gtk_window_set_default_size
    ;set border width
    mov     rdi,qword[window]
    mov     rsi,15                          ;borderwidth
    call    gtk_container_set_border_width
    ;create a button
    mov     rdi,szCaption
    call    gtk_button_new_with_mnemonic
    mov     qword[button],rax
    ;horizontal alignment
    xor     rdi,rdi
    xor     rsi,rsi
    xor     rdx,rdx
    call    gtk_alignment_new
    mov     qword[halign],rax
    ;add the button to the halign container
    mov     rdi,qword[halign]
    mov     rsi,qword[button]
    call    gtk_container_add
    ;add the halign container to the window
    mov     rdi,qword[window]
    mov     rsi,qword[halign]
    call    gtk_container_add
    ;show window
    mov     rdi,qword[window]
    call    gtk_widget_show_all
    ;connect print_msg to the button clicked signal
    xor     r9d,r9d                         ;combination of GConnectFlags
    xor     r8d,r8d                         ;a GClosureNotify for data
    mov     rcx,qword[window]               ;pointer to window instance in RCX
    mov     rdx,print_msg                   ;pointer to the handler
    mov     rsi,szClicked                   ;pointer to the signal
    mov     rdi,qword[button]               ;pointer to window instance in RDI
    call    g_signal_connect_data           ;the value in RAX is the handler, but we don't store it now
    ;connect destroy signal to the window
    xor     r9d,r9d                         ;combination of GConnectFlags
    xor     r8d,r8d                         ;a GClosureNotify for data
    mov     rcx,qword[window]               ;pointer to window instance in RCX
    mov     rdx,gtk_main_quit               ;pointer to the handler
    mov     rsi,szDestroy                   ;pointer to the signal
    mov     rdi,qword[window]               ;pointer to window instance in RDI
    call    g_signal_connect_data           ;the value in RAX is the handler, but we don't store it now
    ;go into application main loop
    call    gtk_main
    ;exit program
    xor     rdi, rdi
    call    exit
    ;event handler for signal button.clicked
print_msg:
    mov     rdi,szMessage
    xor     rax,rax
    call    g_printf
    ret
```