```
; Name        : tree.asm
;
;build        : aclocal && autoconf && automake --add-missing --foreign
;               mkdir build
;               cd build
;               ../configure
;               make
;
;description : an example of glib 2.0 balanced binary trees
;
;source       : https://github.com/steshaw/gtk-examples

bits 64

[list -]
    extern    g_tree_destroy
    extern    g_tree_foreach     ;g_tree_traverse is deprecated since 2.2
    extern    g_tree_height
    extern    g_tree_insert
    extern    g_tree_lookup
    extern    g_tree_new
    extern    g_tree_nnodes
    extern    g_print
    extern    strcmp
    extern    exit
[list +]

%define NNODES    4              ;nodes to show in g_tree_foreach

section .rodata
    names:
    .fred:        db    "Fred",0
    .mary:        db    "Mary",0
    .sue:         db    "Sue",0
    .john:        db    "John",0
    .shelley:     db    "Shelley",0
    .mark:        db    "Mark",0
    .renato:      db    "Renato",0
    properties:
    .loud:        db    "Loud",0
    .obnoxious:   db    "Obnoxious",0
    .drunk:       db    "Drunk",0
    .quiet:       db    "Quiet",0
    .civil:       db    "Civil",0
    .strange:     db    "Strange",0
    .mighty:      db    "Mighty",0
    messages:
    .lookup:      db    "Looking up %s => value %s",10,0
    .height:      db    "Tree height: %d",10,0
    .nodes:       db    "Tree nodes: %d",10,0
    .tree:        db    "Tree:",10,0
    .node:        db    "key: %s %s value: %s",10,0
    userdata:     db    "=>",0

section .data

    tree:         dq    0         ;start of the tree
    flag:         db    0

section .text
global _start

_start:
    ;create tree with compare function
    mov     rdi,compare
    call    g_tree_new
    mov     [tree],rax
    ;insert the key/value pairs
    mov     rdi,[tree]
    mov     rsi,names.fred
    mov     rdx,properties.loud
    call    g_tree_insert

    mov     rdi,[tree]
    mov     rsi,names.mary
    mov     rdx,properties.obnoxious
    call    g_tree_insert

    mov     rdi,[tree]
    mov     rsi,names.sue
    mov     rdx,properties.drunk
    call    g_tree_insert

    mov     rdi,[tree]
    mov     rsi,names.john
    mov     rdx,properties.quiet
    call    g_tree_insert

    mov     rdi,[tree]
    mov     rsi,names.shelley
    mov     rdx,properties.civil
    call    g_tree_insert

    mov     rdi,[tree]
    mov     rsi,names.mark
    mov     rdx,properties.strange
    call    g_tree_insert

    mov     rdi,[tree]
    mov     rsi,names.renato
    mov     rdx,properties.mighty
    call    g_tree_insert
    ;search if Fred is in the list and print result
    mov     rdi,[tree]
    mov     rsi,names.fred
    call    g_tree_lookup
    mov     rdx,rax
    mov     rdi,messages.lookup
    xor     rax,rax
    call    g_print
    ;get tree height and print result
    mov     rdi,[tree]
    call    g_tree_height
    mov     rsi,rax
    mov     rdi,messages.height
    xor     rax,rax
    call    g_print
    ;get tree nodes and print result
    mov     rdi,[tree]
    call    g_tree_nnodes
    mov     rsi,rax
    mov     rdi,messages.nodes
    xor     rax,rax
    call    g_print
    ;print nodes 0 to NNODES
    mov     rdi,messages.tree
    xor     rax,rax
    call    g_print
    mov     rdi,[tree]
    mov     rsi,display
    mov     rdx,userdata
    call    g_tree_foreach
    ;destroy our tree and the pointer
    mov     rdi,[tree]
    call    g_tree_destroy
    xor     rdi,rdi
    mov     [tree],rdi                 ;destroy pointer too
    ;exit the program
    xor     rdi,rdi
    call    exit

display:
    push    rbp
    mov     rbp,rsp
    mov     rcx,rsi
    mov     rsi,rdi
    mov     rdi,messages.node
    xor     rax,rax
    call    g_print
    xor     rax,rax
    inc     byte[flag]
    cmp     byte[flag],NNODES           ;stop after n nodes
    jl      .exit
    inc     rax                         ;return TRUE
.exit:
    mov     rsp,rbp
    pop     rbp
    ret

compare:
    ;compare strings in rdi and rsi, returning
    ;-1 when string rdi comes before string rsi
    ; 0 when both strings are equal
    ; 1 when string in rsi comes after string rdi
    call    strcmp
    ret

```