А. С. НАРКЕВИЧ
БГТУ (г. Минск, Республика Беларусь)

ПРОГРАММИРОВАНИЕ НА MASM В АРХИТЕКТУРЕ *X64*

Рассматриваются особенности программирования в архитектуре x64 для операционной системы Microsoft Windows на языке MASM. Архитектура x64 – расширение, обратно совместимое с архитектурой x86, в которой обеспечивается поддержка 16-битного и 32-битного кода приложений и операционных систем без их модификации или перекомпиляции [1].

В 64-разрядном режиме процессор предоставляет возможность 64-битной адресации и осуществляет поддержку 16-ти 64-битных регистров общего назначения и новых инструкций. 64-разрядные регистры используются при обработке чисел порядка 2^{33} (8.589.934.592) и больше. Если 32-битному процессору требуется несколько тактов для выполнения операций с целочисленными операндами, то 64-битный обрабатывает такие значения за один такт.

Регистры общего назначения в процессорах x64 именуются с префикса R. Новые регистры имеют номера от R8 до R15. Для обращения к младшим 8-, 16- и 32-битам новых регистров используются суффиксы b, w и d соответственно.

Появились новые возможности:

- использовать адресацию относительно RIP регистра, например «MOV AL, [RIP]»;
- операции с плавающей запятой выполняются с помощью 16 регистров XMM;
- для относительной адресации используется регистр RIP (всегда указывает на следующую инструкцию);
- можно обращаться к младшим байтам индексных регистров (bpl, spl, dil, sil).

Операции с 32 битными операндами обнуляют старшие 4 байта результата.

Инструкция не может ссылаться одновременно на младший байт старых регистров (ah,bh,dh,ch) и младший байт новых регистров.

В Win32 существуют следующие соглашения о вызове: stdcall, cdecl, fastcall, thiscall и так далее. В отличие от Win32 в Win64 есть только одно соглашение о вызове x86-64 fast calling conversion (соглашение о быстрой передаче параметров для x86-64). В соответствии с которым, первые четыре целочисленных аргумента (слева направо) передаются в 64-битных регистрах RCX, RDX, R8 и R9. Остальные целочисленные аргументы передаются через стек (справа налево). Для каждого аргумента, даже переданного через регистр, вызывающая функция обязана резервировать для него место в стеке, уменьшая значение регистра RSP (указателя стека). Необходимо резервировать в стеке, как минимум, 32 байта для регистров RCX, RDX, R8, R9. Если передается более четырех целочисленных параметров, в стеке нужно зарезервировать соответствующее дополнительное пространство.

Значения типа Integer возвращаются в регистре RAX Если длина возвращаемых значений больше 64 бит (для структур), то в RAX возвращается их адрес. Стек освобождает вызывающая функция. Подробности можно найти на сайте [2].

Ниже приведен пример использования 64-битной версии MASM ML64.EXE, свободно доступной в Windows Platform SDK.

```
;Conf.asm
includelib c:/masm64/lib/kernel32.lib
includelib c:/masm64/lib/user32.lib
extrn MessageBoxA : PROC    ; внешняя API – функция
extrn ExitProcess : PROC         ; внешняя API – функция
; секция данных с атрибутами по умолчанию(чтение и запись)
.data
mytit db 'Hi 64-bit!', 0
mymsg db 'Hello World!', 0
; секция кода с атрибутами по умолчанию(чтение и исполнение)
.code
Main PROC
sub    rsp, 28h          ; выравнивание стека: 8*4+8
mov r9d, 0               ; uType = MB_OK
lea r8, mytit             ; заголовок окна
lea rdx, mymsg        ; текст
mov rcx, 0               ; hWnd = HWND_DESKTOP
call MessageBoxA
mov ecx, eax           ; uExitCode = MessageBox(...)
call ExitProcess
Main ENDP
End
```

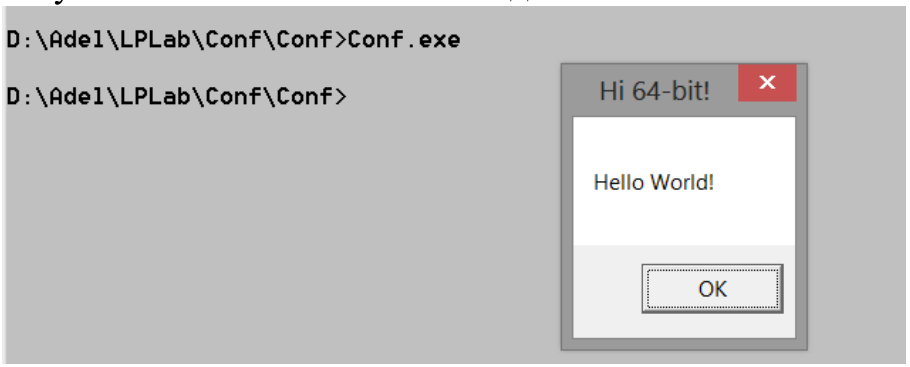
Ассемблируем: ml64.exe Conf.asm /link /subsystem:windows /entry:Main

```
D:\Adel\LPLab\Conf\Conf>c:\masm32\bin64\ml64.exe Conf.asm  /link /subsystem:windows  /entry:Main
Microsoft (R) Macro Assembler (x64) Version 14.00.24210.0
Copyright (C) Microsoft Corporation.  All rights reserved.

 Assembling: Conf.asm
Microsoft (R) Incremental Linker Version 14.00.24215.1
Copyright (C) Microsoft Corporation.  All rights reserved.

/OUT:Conf.exe
Conf.obj
/subsystem:windows
/entry:Main
```

Результат выполнения имеет вид:



Изложенный материал может быть полезен магистрантам и студентам, изучающим языки программирования и теорию компиляции.

Список использованных источников

1. MASM для x64 (ml64.exe). https://docs.microsoft.com/ru-ru/cpp/assembler/masm/masm-for-x64-ml64-exe.
2. Chris Lomont. Introduction to x64 Assembly. https://software.intel.com/en-us/articles/introduction-to-x64-assembly.