01.11.2020

Архитектура вычислительных систем.

Микропроект №1.

НИУ ВШЭ, ФКН ПИ

Албогачиев Алим

БПИ196

1.Текст задания

Разработать программу, определяющей максимальное значение параметра числа линейной рекуррентной последовательности tn=tn-1+tn-2+tn-3 при n≥3 ("числа трибоначчи") со стартовой тройкой чисел [0,0,1], которое не выходит за пределы беззнакового двойного машинного слова.

2.Список источников

1. http://natalia.appmat.ru/c&c++/assembler.html
2. https://www.cyberforum.ru/fasm/thread1240599.html

Оба сайта предоставляют справочные материалы о языке FASM

3.Код программы

1. Секция ***data*** и секци ***code*** с вызовами определенных далее функций:

```
format PE console
entry start

include 'win32a.inc'

;--------------------------------------------------------------------------
;Вариант 1.
;Албогачиев Алим Ахмедович, БПИ196


section '.data' data readable writable


        strScanInt   db '%d', 0
        strOutput db 'Max integer number of Tribonacci sequence before double world oveflow:', 10, 0


        num1          dd 0
        num2          dd 0
        num3          dd 1
        curr          dd 1
        tmpStack     dd ?


;--------------------------------------------------------------------------
section '.code' code readable executable
start:
;вызов функции, вычисляющей необходимую последовательность.
        call Tribonacci
;вызов функции, выводящей полученное число.
        call Output
finish:
;вызов фукнции принимающей char для продления работы программы
        call [getch]

        push 0
        call [ExitProcess]
```

2. Определение фукнций ***out*** (вывод данных) и ***Tribonacci*** (вычисление последовательности:

```
;-----------------------------------------------------------------------------
Output:
        ;пушим в стек строку для вывода
        push strOutput
        call [printf]
        ;сохраняем текущее значение стека, чтобы в последствии вернуться к нему
        mov [tmpStack], esp
        ;выводим найденное число
        mov eax, [num2]
        push eax
        push strScanInt
        call[printf]
        mov esp, [tmpStack]
        ret
;-----------------------------------------------------------------------------
Tribonacci:
        xor edx, edx
TribonacciLoop:
        ;перенос значения первого числа в eax
        mov eax, [num1]
        ;прибаление к нему двух других
        add eax, [num2]
        add eax, [num3]
        ;перенос в первую ячейку второй
        mov edx, [num2]
        mov [num1], edx
        ;перенос во вторую ячейку третьей
        mov edx,[num3]
        mov [num2], edx
        ;перенос в третью ячейку нового значения
        mov [num3], eax
        ;проверка переполнения
        jo endTribonacciLoop
        jmp TribonacciLoop
endTribonacciLoop:
ret
```

Все вычисления проходят в регистре ***eax***. Хранение исходной и в последствии текущей тройки чисел осуществляется в переменных ***num1, num2, num3***.

В регистр eax переносится значение ***num1***, затем прибавляются значения ***num2*** и ***num3***. После все элементы тройки сдвигаются вниз, а новый элемент занимает место в переменной ***num3***. Цикл выполняется до того момента, как в флаг ***OF*** не покажет переполнение. В таком случае, максимальное число до переполнения будет лежать в переменной ***num2***.

3. Секция ***idata***:

```
;--------------------------------------------------------------

section '.idata' import data readable
    library kernel, 'kernel32.dll',\
            msvcrt, 'msvcrt.dll',\
            user32,'USER32.DLL'

include 'api\user32.inc'
include 'api\kernel32.inc'
    import kernel,\
           ExitProcess, 'ExitProcess',\
           HeapCreate,'HeapCreate',\
           HeapAlloc,'HeapAlloc'
  include 'api\kernel32.inc'
    import msvcrt,\
           printf, 'printf',\
           scanf, 'scanf',\
           getch, '_getch'
```

5. Скриншоты