

Пояснительная записка к микропроекту №1
НИУ ВШЭ Департамент программной инженерии
Программа, вычисляющая количество чисел Ферма до беззнакового машинного слова
Ловчикова Юлия Васильевна, БПИ192, вариант 14

1. Задание

Разработать программу определения количества чисел Ферма от 1 до беззнакового машинного слова.

2. Математическое обоснование и алгоритм

Число Ферма – число F_n вида $2^{2^n} + 1$, где $n \geq 0$.

$$2^{2^n} + 1 = 2^{2^{n-1} + 2^{n-1}} + 1 = (2^{2^{n-1}})^2 + 1 \quad \forall n \geq 1$$

$$F_n = (F_{n-1} - 1)^2 + 1 \quad \forall n \geq 1; F_0 = 3$$

Исходя из полученной выше рекуррентной формулы, имеем, что справедлив следующий алгоритм поиска количества чисел Ферма, меньших беззнакового машинного слова $\text{max} = 2^{16} = 65536$ (псевдокод на языке C#):

```
const int max = 65536;
int cnt = 0;
int ferma = 3;
do {
    cnt+=1;
    ferma=(int)Math.Pow((ferma-1), 2)+1;
}
while (ferma < max);
```

3. Программа

3.1. Список переменных:

cnt dd 0 – счётчик чисел, изначально равен нулю по логике цикла
ferma dd 3 – переменная, которой на каждой итерации цикла присваивается очередное значение числа из последовательности; изначально равно $F_0 = 3$
max dd 10000h – константа, равная 65536 – беззнаковому машинному слову

3.2. Метка countFerma

Является прототипом цикла while. Сначала выполняется заход в цикл, и в конце проверяется условие $\text{ferma} < \text{max}$. Если оно true, то заходим в метку снова; в противном случае, выходим из метки с помощью ret.

3.3. Формат выходных данных

Выводится константа – беззнаковое машинное слово, а также все найденные числа Ферма и их количество:

```
MaxValue=65536
Next Ferma number=3
Next Ferma number=5
Next Ferma number=17
Next Ferma number=257
Count of Ferma numbers 4
```

4. Код программы

```
format PE console
entry start
include 'win32a.inc'

section '.idata' import data readable writeable
    library kernel32, 'kernel32.dll', \
        msvcrt, 'MSVCRT.DLL'

    import kernel32, \
        ExitProcess, 'ExitProcess'

    import msvcrt, \
        printf, 'printf', \
        getch, '_getch'

section '.data' data readable writeable
    szResult db 'Count of Ferma numbers %d',10,0
    cnt dd 0
    ferma dd 3
    max dd 10000h ;65536
    szCheckMax db 'MaxValue=%d',10,0
    szCheckSteps db 'Next Ferma number=%d' ,10,0

section '.text' code readable executable
    start:
        invoke printf, szCheckMax, [max]
        add esp, 8
        call countFerma
        invoke printf, szResult, dword[cnt]

    exit:
        call [getch]
        push 0
        call [ExitProcess]

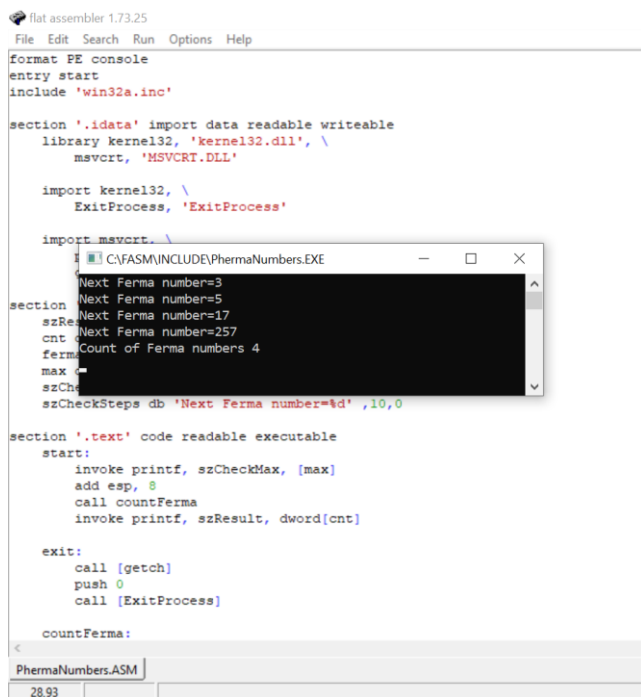
    countFerma:
        invoke printf, szCheckSteps, [ferma]
        add esp, 8

        mov eax, [cnt]
        inc eax
        mov [cnt], eax

        mov eax, [ferma]
        sub eax, 1
        mul eax
        inc eax
        mov [ferma], eax
        cmp eax, [max]
        jnl countFerma
        ret
```

5. Испытания

При запуске программы результат всегда одинаков, поскольку верхняя граница поиска является константой:



```
flat assembler 1.73.25
File Edit Search Run Options Help

format PE console
entry start
include 'win32a.inc'

section '.idata' import data readable writable
    library kernel32, 'kernel32.dll', \
        msvcrt, 'MSVCRT.DLL'

    import kernel32, \
        ExitProcess, 'ExitProcess'

    import msvcrt, \
        printf, 'printf'

section '.text' code readable executable
    start:
        invoke printf, szCheckMax, [max]
        add esp, 8
        call countFerma
        invoke printf, szResult, dword[cnt]

    exit:
        call [getch]
        push 0
        call [ExitProcess]

countFerma:
    Next Ferma number=3
    Next Ferma number=5
    Next Ferma number=17
    Next Ferma number=257
    cnt Count of Ferma numbers 4
    max =
    szCheckMax db 'Next Ferma number=%d', 10, 0

    section '.text' code readable executable
    start:
        invoke printf, szCheckMax, [max]
        add esp, 8
        call countFerma
        invoke printf, szResult, dword[cnt]

    exit:
        call [getch]
        push 0
        call [ExitProcess]

countFerma:
    Next Ferma number=3
    Next Ferma number=5
    Next Ferma number=17
    Next Ferma number=257
    cnt Count of Ferma numbers 4
    max =
    szCheckMax db 'Next Ferma number=%d', 10, 0
```

6. Репозиторий исходного кода

<https://github.com/yuvlovchikova/ComputingArchitectureHW>

7. Источники

- 1) <http://softcraft.ru/edu/comparch/tasks/mp01/> - формулировка задания
- 2) https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%BE_%D0%A4%D0%B5%D1%80%D0%BC%D0%B0#cite_note-2 – определение чисел Ферма (и так его знала, но для полноты привожу как источник)
- 3) https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D1%81%D0%BB%D0%BE%D0%B2%D0%BE#%D0%A0%D0%B0%D0%B7%D0%BC%D0%B5%D1%80_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE_%D1%81%D0%BB%D0%BE%D0%B2%D0%B0%D0%BD%D0%B0_%D1%80%D0%B0%D0%B7%D0%BB%D0%B8%D1%87%D0%BD%D1%8B%D1%85_%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0%D1%85 – определение беззнакового машинного слова
- 4) <http://osinavi.ru/index.php?param1=4¶m2=18&hidden=0&supp=1> – источник с множеством статей об ассемблере
- 5) <http://natalia.appmat.ru/c&c++/assembler.html> - о регистрах