Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет «Высшая школа экономики»

Факультет компьютерных наук

Департамент программной инженерии

Отчёт к домашнему заданию
по дисциплине
«Архитектура вычислительных систем»

Работу выполнил:
студент группы БПИ191 Я. Д. Карпунькин

Москва 2020

Постановка задачи

Разработать программу, определяющую число ‘0’ и ‘1’ в битовом представлении заданного двойного машинного слова.

Алгоритм решения

Входными данными для программы является одно целое неотрицательное число в диапазоне от 0 до $2^{32} - 1$, вводимое через консоль. Организовано считывание данного числа с помощью scanf, а также сообщение с помощью пользователю для корректного ввода.

Далее с помощью вспомогательной переменной мы последовательно проверяем каждый бит из двоичного представления введённого числа, и в случае обнаружения 0 или 1 увеличиваем соответствующий счётчик на 1. Сложность алгоритма линейная. После проверки каждого бита в консоль выводится ответ – количество найденных 0 и 1.

Использованные данные

```
digit       db "%d", 0
string0     db 256 DUP(?)
string1     db 256 DUP(?)
newStr      db '', 10, 13, 0

number          dd ?
tmpStack        dd ?
num0            dd 0
num1            dd 0
indic           dd 0
```

Digit – необходимая для формата переменная
String0/1 – переменные для записи итогового ответа.
NewStr – переменная для перехода на новую строчку при выводе.
Number – переменная, хранящая введённое число.
TmpStack – переменная для хранения значения регистра esp.
Num0/1 – переменные для хранения количества встретившихся 0 / 1.
Indic – переменная для правильного подсчёта значащих нулей.

Описание кода

Секция с кодом состоит из вызова 3 процедур из метки start и завершения работы программы в метке finish.

```
section '.code' code readable executable
start:
        call Begin
        call MainPart
        call Output
```

```
finish:
        call [getch]
        invoke ExitProcess, 0
```

В 1 процедуре происходит считывание машинного слова. Мы не можем осуществить проверку введенного числа на принадлежность диапазону, так как иначе произойдет переполнение и она будет работать некорректно.

```
Begin:
        mov [tmpStack], esp
        invoke printf, "Input number from 0 to 2^32 - 1 to find out, how many 0/1 in its bit represention: "
        cinvoke scanf, digit, number
        mov esp,[tmpStack]
        ret
```

Далее идет главная часть программы. Там мы идём по всем битам числа с помощью поразрядной конъюнкции значения регистра edx и введённого числа. В случае первой встречи 1 мы изменяем индикатор indic, свидетельствующий о том, что все нули, которые мы встретим далее будут являться значащими. Далее просто увеличиваем переменные, отвечающие за число встретившихся 0/1 с помощью команды inc.

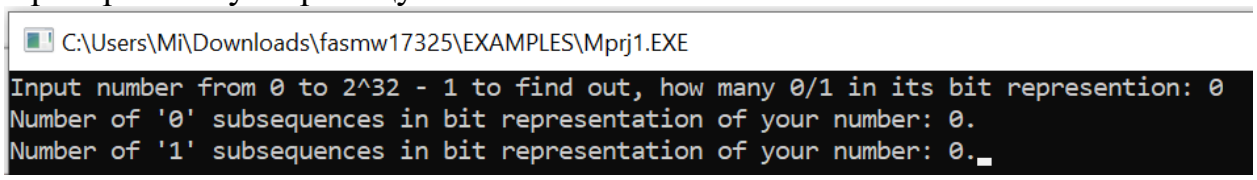
```
MainPart:
        mov [tmpStack], esp
        mov edx, 10000000000000000000000000000000b
        jmp body

                body:
                        cmp edx, 0
                        je over

                        mov eax, [number]
                        and eax, edx
                        cmp eax, 0
                        je do0
                        jne do1

                        do0:
                                cmp [indic], 0
                                je fin
                                inc [num0]
                                jmp fin

                        do1:
                                inc [num1]
                                cmp [indic], 0
                                je ChangeIndicator
                                jmp fin

                        ChangeIndicator:
                                mov [indic], 1
                                jmp fin
                        fin:
                                shr edx, 1
                                jmp body

                over:
                        mov esp, [tmpStack]
                        ret
```

В процедуре output выводим найденное количество вхождений 0/1 в битовое представление заданного двойного машинного слова.

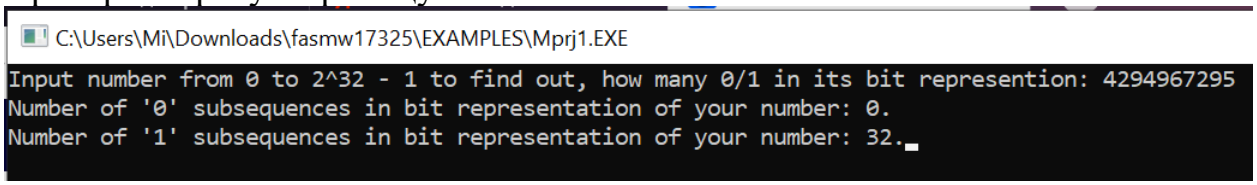
```
Output:
        invoke sprintf, string0, "Number of '0' subsequences in bit representation of your number: %d.",[num0]
        invoke sprintf, string1, "Number of '1' subsequences in bit representation of your number: %d.",[num1]
        cinvoke printf, string0
        cinvoke printf, newStr
        cinvoke printf, string1
```

Примеры работы программы

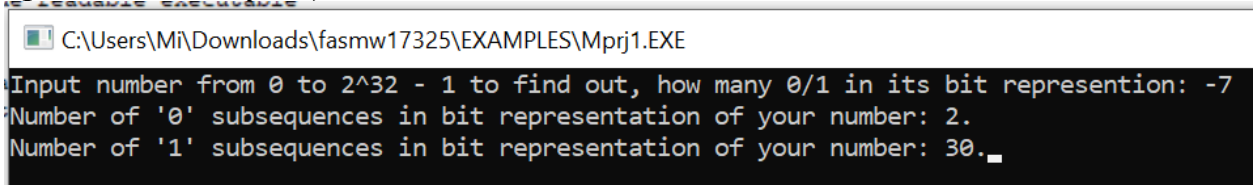
1) Проверим левую границу:



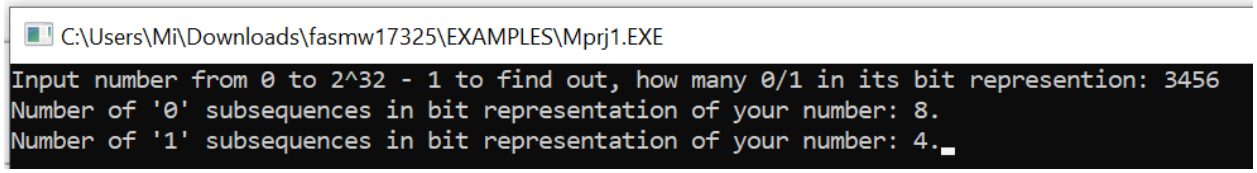
2) Проверим правую границу:



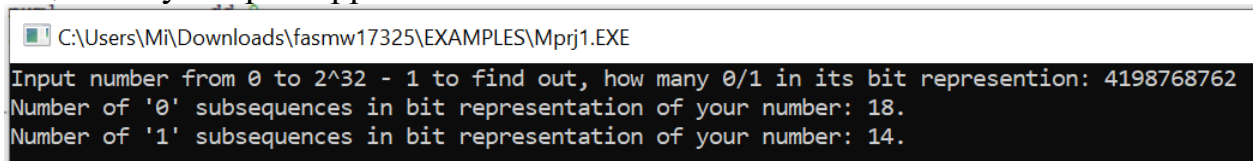
3) Теперь проверим что будет со значениями вне диапазона (всё должно браться по модулю):



4) Запустим при обычном значении:



5) Ещё 1 запуск при корректном большом значении:



Использованные источники

1) https://wiki2.org/ru/Машинное_слово
2) https://www.frolov-lib.ru/books/bsp/v02/ch12_4.htm
3) https://drive.google.com/file/d/1cYZ68FZQJwEwntH5b9n8LPpLmHcvX7pN/view?usp=sharing