Отчет по лабораторной работе №5

Архитектура компьютера

Никифоров Захар Сергеевич

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Порядок выполнения работы

2.1 Символьные и численные данные в NASM

Создаем каталог *lab06*, а в нем создаем файл *lab6-1.asm*, записываем код из *Листинг 6.1* и компилируем его.

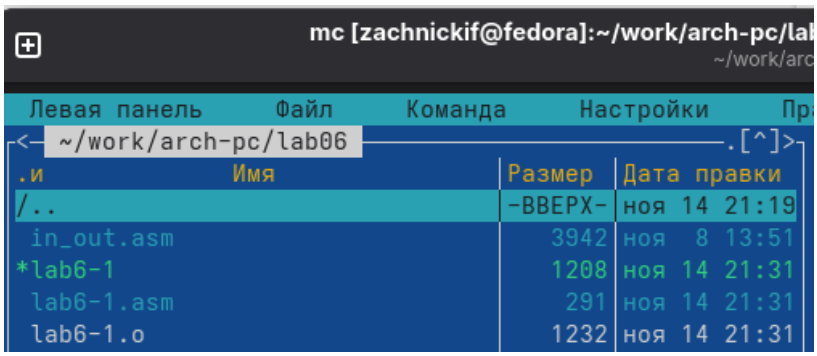

Рисунок 2.1: Скриншот терминала mc 1

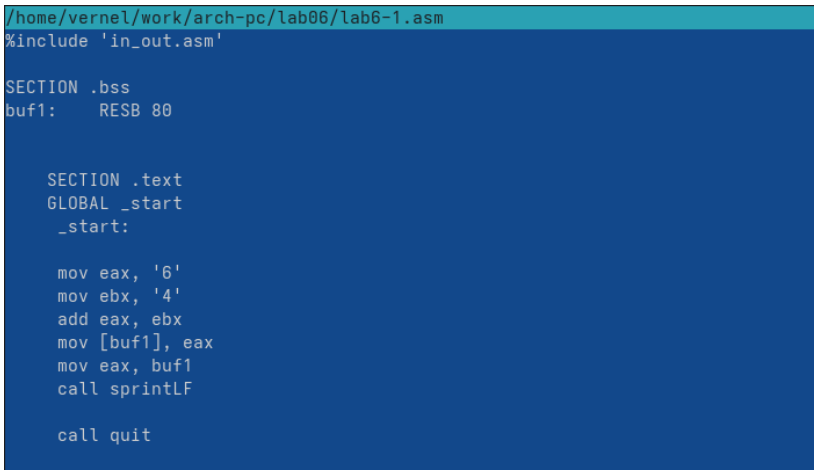

Рисунок 2.2: Скриншот терминала mc 2

```
zachnickif@fedora:~/work/arch-pc/lab06$ ./lab6-1
j
```

Рисунок 2.3: Скриншот терминала mc 3

Получаем на вывод букву *j*. Это происходит потому, что код символа 6 равен „00110110“ в двоичном представлении (или 54 в десятичном представлении), а код символа 4 – „00110100“ (52). Команда *add eax, ebx* запишет в регистр eax сумму кодов – 01101010 (106), что в свою очередь является кодом символа j.

Далее изменим код программы, убрав кавычки вокруг цифр.

```
/home/vernel/work/arch-pc/lab06/lab6-1.asm
%include 'in_out.asm'

SECTION .bss
buf1:    RESB 80


    SECTION .text
    GLOBAL _start
     _start:

     mov eax, 6
     mov ebx, 4
     add eax, ebx
     mov [buf1], eax
     mov eax, buf1
     call sprintLF

     mov ecx, buf1
     mov edx, 80
     call sread
     call quit
```

Рисунок 2.4: Скриншот терминала mc 4

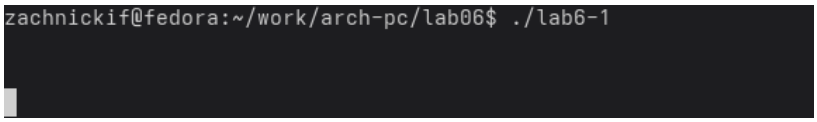

Рисунок 2.5: Скриншот терминала mc 5

На выход мы получаем символ переноса строки, который имеет код 10.

Рисунок 2.6: Скриншот терминала mc 6

Символ не выражается в качестве *n* с обратной дробью, но видно перенос строки.

Теперь создадим новый файл *lab6-2.asm* запишем в него код из *Листинг 6.2* и скомпилируем его.

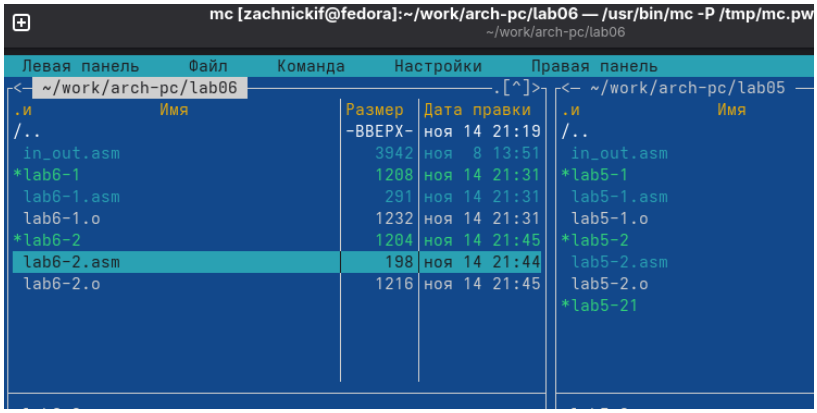

Рисунок 2.7: Скриншот терминала mc 7

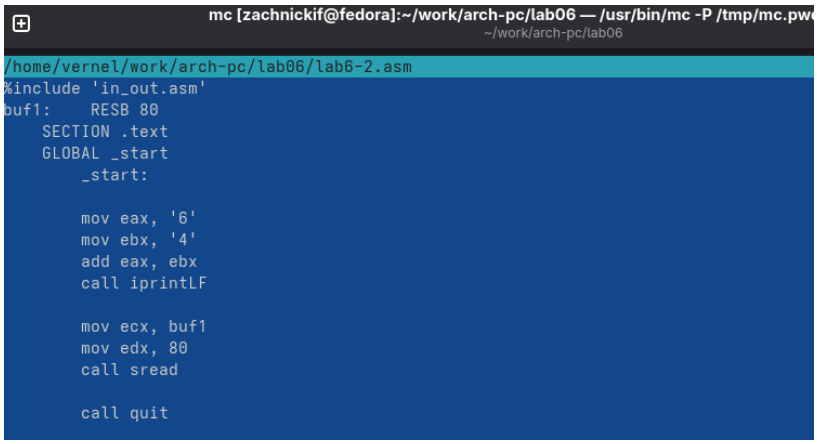

Рисунок 2.8: Скриншот терминала mc 8

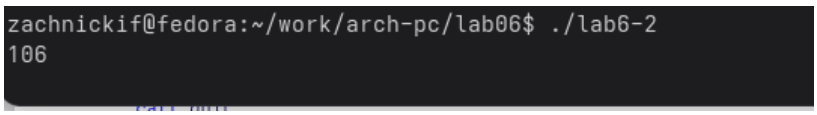

Рисунок 2.9: Скриншот терминала mc 9

В результате работы программы мы получим число 106. В данном случае, как и в первом, команда add складывает коды символов „6“ и „4“ (54+52=106). Однако, в отличии от программы из *Листинг 6.1*, функция *iprintLF* позволяет вывести число, а не символ, кодом которого является это число.

Теперь внесем изменения, убрав кавычки и заменив *iprintLF* на *iprint*.

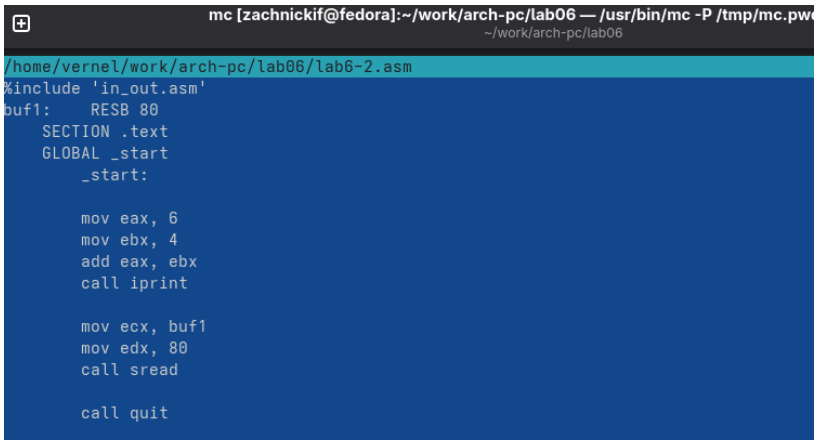

Рисунок 2.10: Скриншот терминала mc 10

Рисунок 2.11: Скриншот терминала mc 11

В результате работы программы мы получим число 10. Из отличий заметно отстутствие переноса строки.

2.2 Выполнение арифметических операций в NASM

Создадим файл *lab6-3.asm*, запишем в него код из *Листинг 6.3* и скомпилируем его.

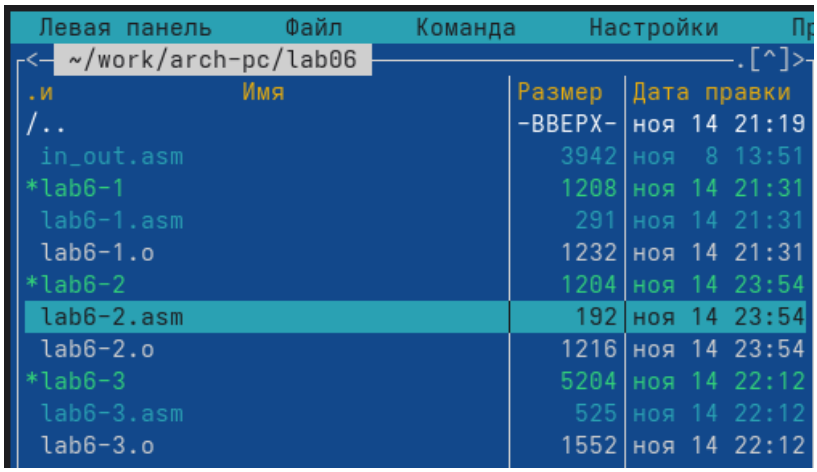

Рисунок 2.12: Скриншот терминала mc 12

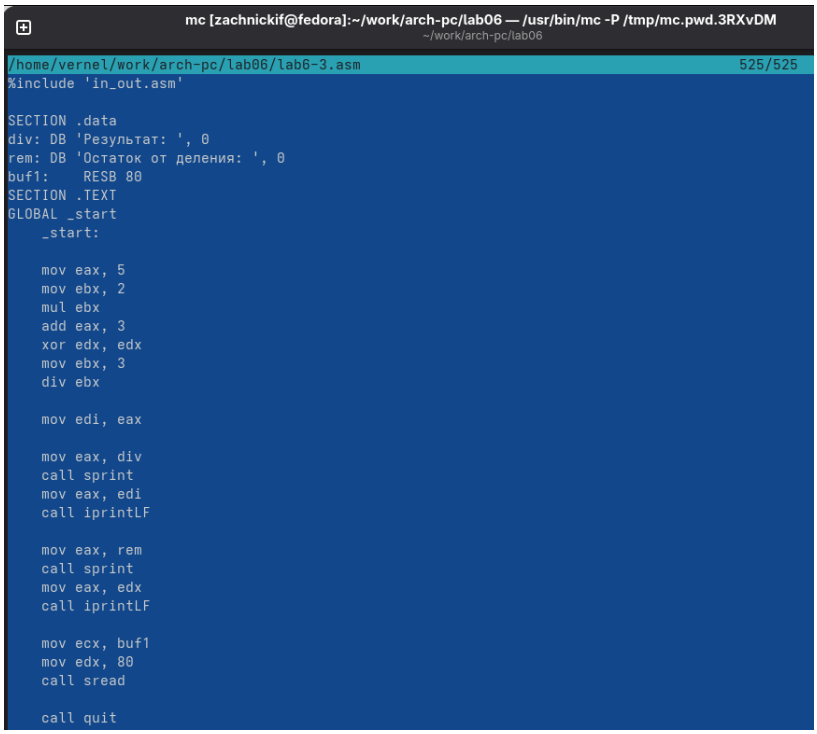

Рисунок 2.13: Скриншот терминала mc 13

Рисунок 2.14: Скриншот терминала mc 14

Результат такой, какой и должен быть. Теперь внесем изменения, изменив функцию на $f(x) = (46+2)/5$.

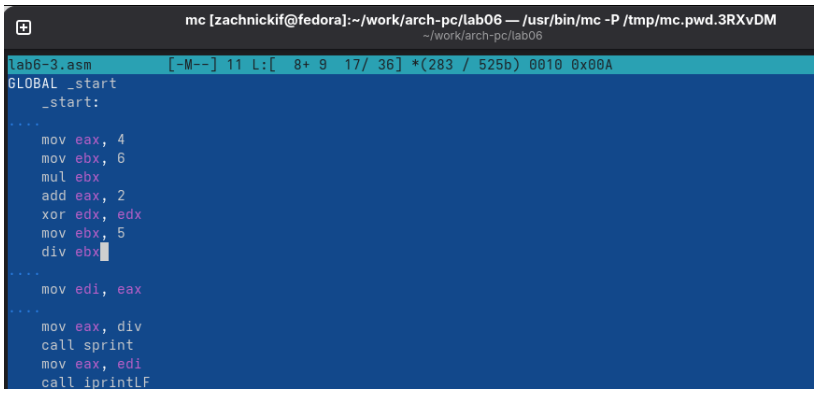

Рисунок 2.15: Скриншот терминала mc 15

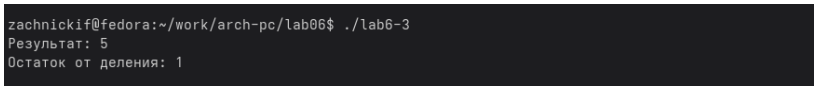

Рисунок 2.16: Скриншот терминала mc 16

Результат изменился соответственно функции.

Теперь создадим файл *variant.asm*, запишем в него код из *Листинг 6.4* и скомпилируем.

Рисунок 2.17: Скриншот терминала mc 17

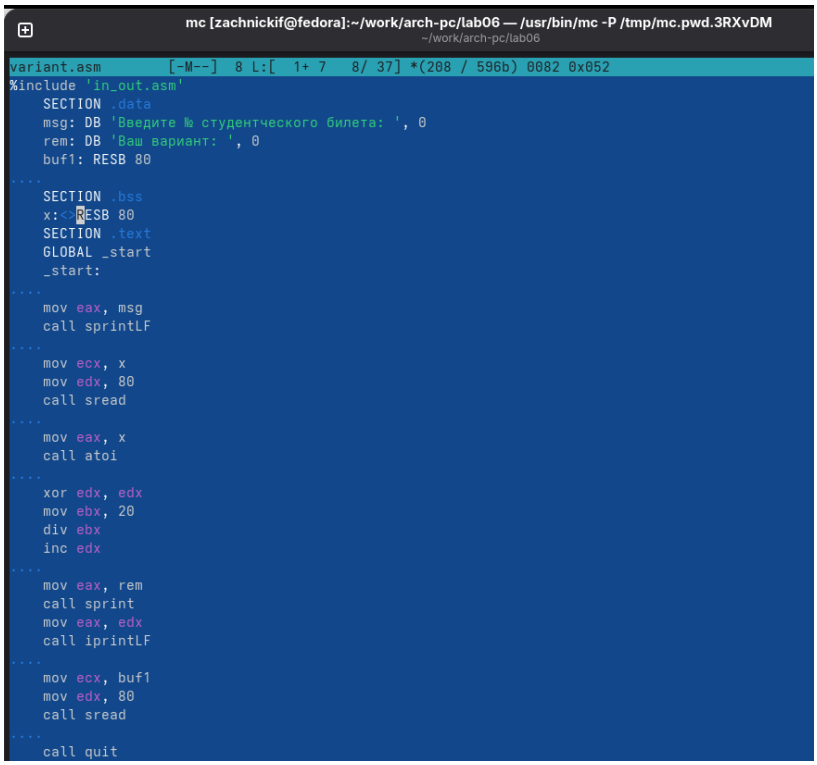

Рисунок 2.18: Скриншот терминала mc 18

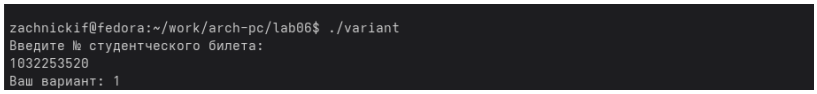

Рисунок 2.19: Скриншот терминала mc 19

В результате я получил 1. Это соответствует функции, так как мой номер студенческого билета кратен 20 и имеет остаток 0, а вариант 0 быть не может, поэтому программа превратила 0 в 1.

1. Следующие строки:

Рисунок 2.20: Скриншот терминала mc 20

2. Чтобы прочитать номер студенческого билета, как число.

3. Эта функция нужна, чтобы преобразовать строку в число.

4. Следующие строки:

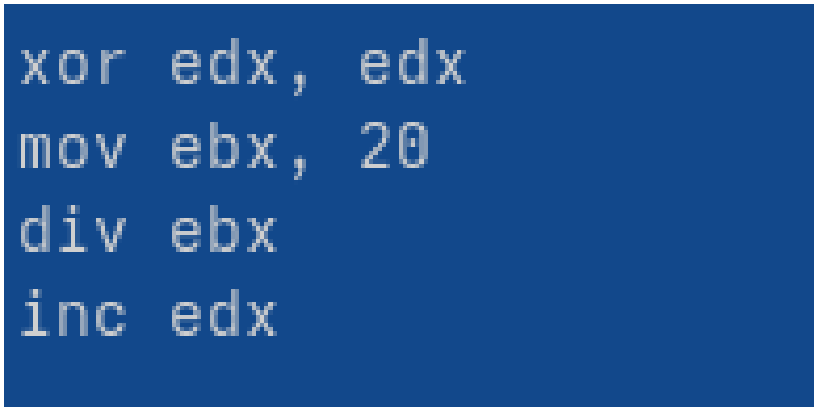

Рисунок 2.21: Скриншот терминала mc 21

5. В *EDX*, причем всегда.

6. Эта инструкция нужна, чтобы увеличить остаток на 1. В нашем случае помогает начать счёт варианта с 1, а не 0.

7. Следующие строки:

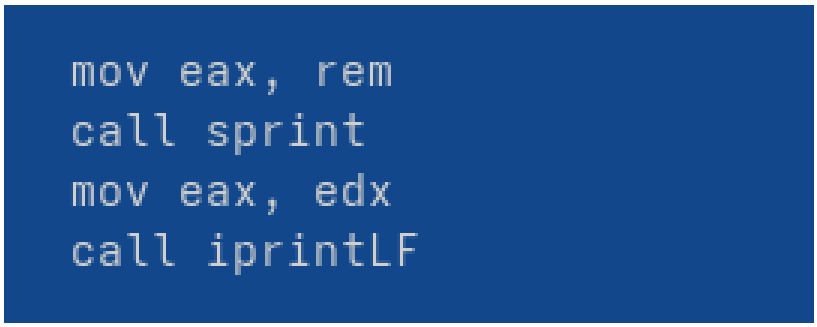

Рисунок 2.22: Скриншот терминала mc 22

3 Задание для самостоятельной работы

Создам файл *practice.asm* и, опираясь на прошлые, напишу следующий код, который будет вычислять функцию *f(x) = (10+2x)/3* из введенного x, согласно моему варианту:

Рисунок 3.1: Скриншот терминала mc 23

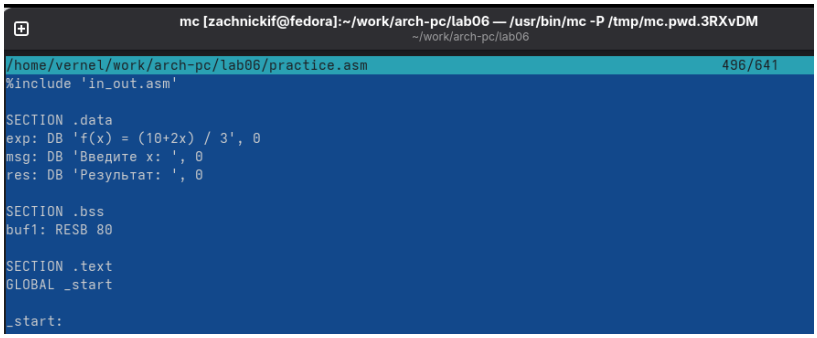

Рисунок 3.2: Скриншот терминала mc 24

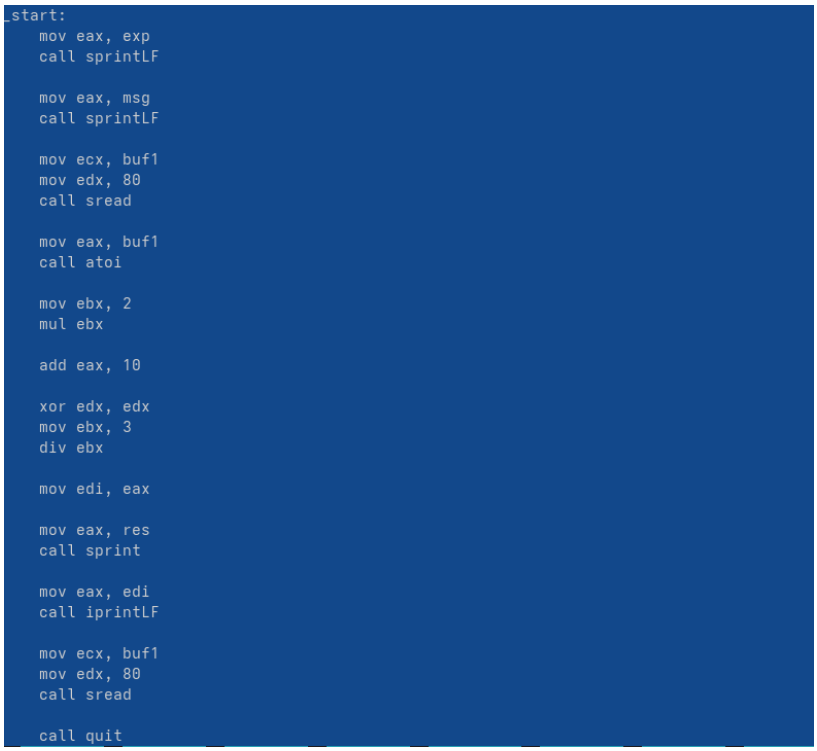

Рисунок 3.3: Скриншот терминала mc 25

Использую значения x равные 1 и 10.

Рисунок 3.4: Скриншот терминала mc 26

Рисунок 3.5: Скриншот терминала mc 27

Программа работает корректно, проверив аналитически, я пришел к выводу, что

и арифметически корректно тоже. # **Выводы**

В ходе работы были освоены арифметические конструкции языка *nasm*.