Лабораторная работа №6

Архитектура компьютера

Мурашов Иван Вячеславович

Содержание

# 1 Цель работы

Целью данной лабораторной работы является освоение арифметических инструкций языка ассемблера NASM.

# 2 Задание

1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM
3. Выполнение заданий для самостоятельной работы

# 3 Выполнение лабораторной работы

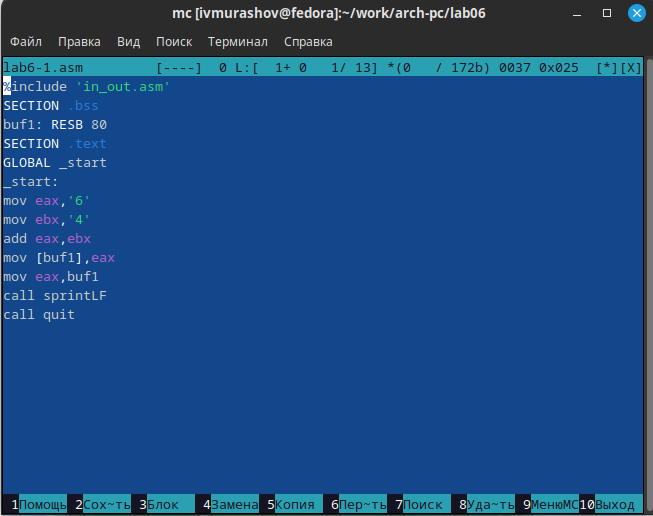
## 3.1 Символьные и численные данные в NASM

Создаю каталог для программ лабораторной работы №6, перехожу в него и создаю файл lab6-1.asm (рис. [??]).

Создание каталога и файла в нём

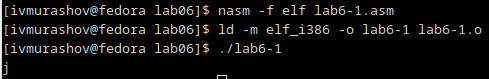
Создание каталога и файла в нём

Открываю файл lab6-1.asm для редактирования в mcedit. Ввожу в файл текст программы из листинга 6.1 (рис. [??]).



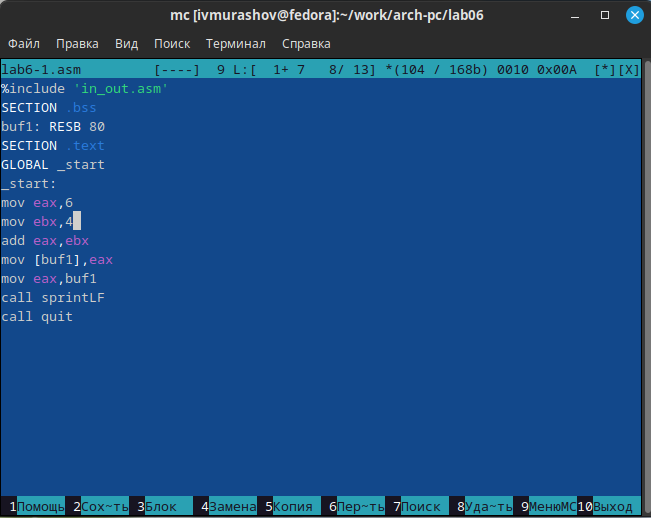
Окно Midnight Commander. Редактор mcedit

Создаю исполняемый файл и запускаю его. Вывод программы: символ j, так как программа вывела символ, соответствующий сумме двоичных кодов символов 4 и 6 по системе ASCII (рис. [??]).



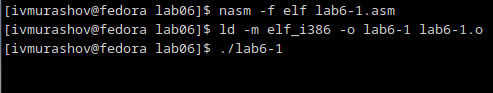
Трансляция, компоновка и запуск файлов

Изменяю текст программы, записав в регистры числа вместо символов. (рис. [??]).



Окно Midnight Commander. Редактор mcedit

Создаю исполняемый файл и запускаю его. В данном случае выводится символ с кодом 10, которое соответствует символу “LF,|n” по таблице ASCII. Символ не отображается при выводе на экран, так как является символом перевода строки (рис. [??]).



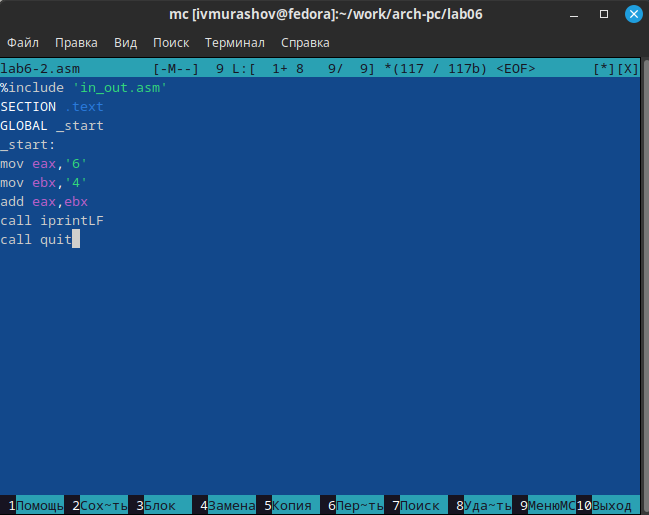
Трансляция, компоновка и запуск файлов

Создаю файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 (рис. [??]).

Создание файла

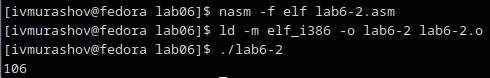
Создание файла

Ввожу в данный файл текст программы из листинга 6.2 (рис. [??]).



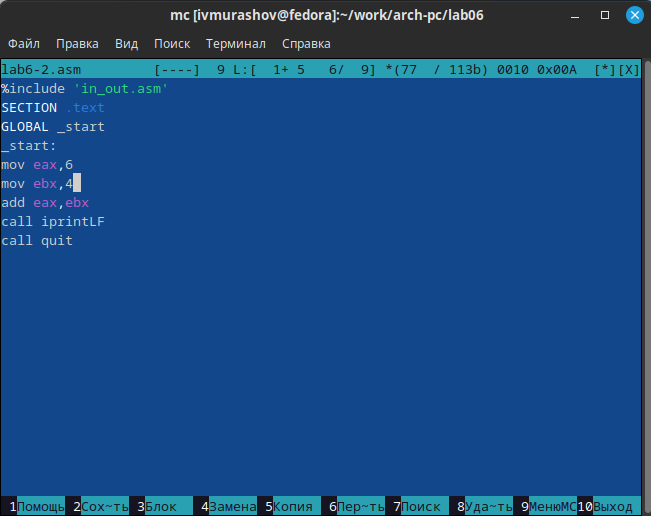
Окно Midnight Commander. Редактор mcedit

Создаю исполняемый файл и запускаю его. В результате работы программы я получаю число 106. В отличии от программы из листинга 6.1, функция iprintLF позволяет вывести число, а не символ, кодом которого является это число (рис. [??]).



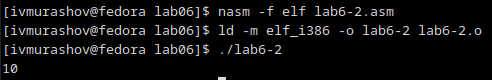
Трансляция, компоновка и запуск файлов

Изменяю текст программы, записав в регистры числа вместо символов. (рис. [??]).



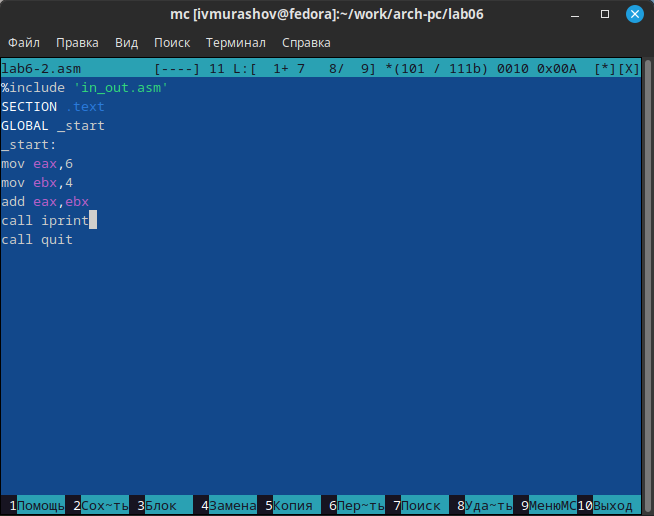
Окно Midnight Commander. Редактор mcedit

Создаю исполняемый файл и запускаю его. В результате выводится число 10, так как программа складывает не коды в системе ASCII, а сами числа (рис. [??]).



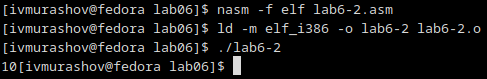
Трансляция, компоновка и запуск файлов

Заменяю функцию iprintLF на iprint (рис. [??]).



Окно Midnight Commander. Редактор mcedit

Создаю исполняемый файл и запускаю его. Вывод функций iprintLF и iprint отличается тем, что в последнем случае не происходит перехода на новую строку при выводе значения (рис. [??]).



Трансляция, компоновка и запуск файлов

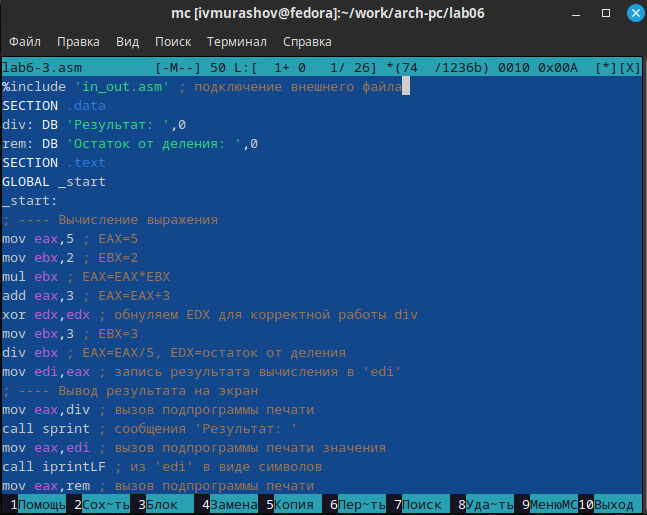
## 3.2 Выполнение арифметических операций в NASM

Создаю файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 (рис. [??]).

Создание файла

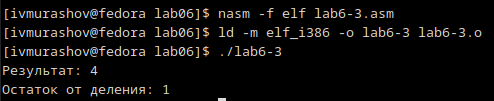
Создание файла

Ввожу в файл текст программы из листинга 6.3 (рис. [??]).



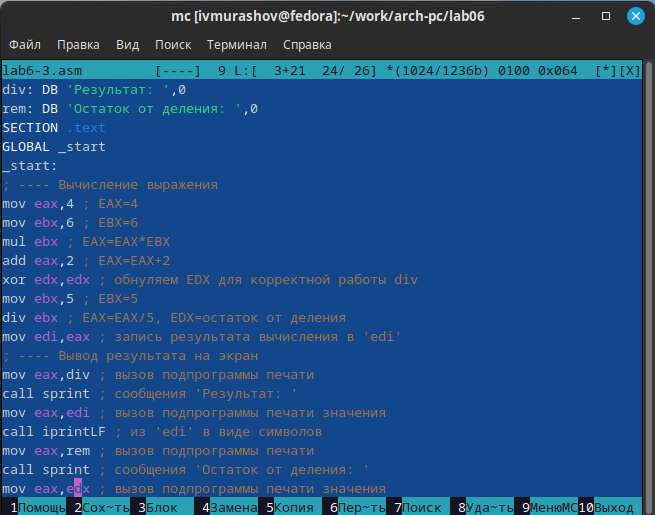
Окно Midnight Commander. Редактор mcedit

Создаю исполняемый файл и запускаю его (рис. [??]).



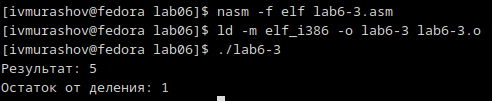
Трансляция, компоновка и запуск файлов

Изменяю текст программы для вычисления выражения f(x) = (4 \* 6 + 2)/5 (рис. [??]).



Окно Midnight Commander. Редактор mcedit

Создаю исполняемый файл и запускаю его. При самостоятельном расчёте результат совпал с ранее полученным выводом (рис. [??]).



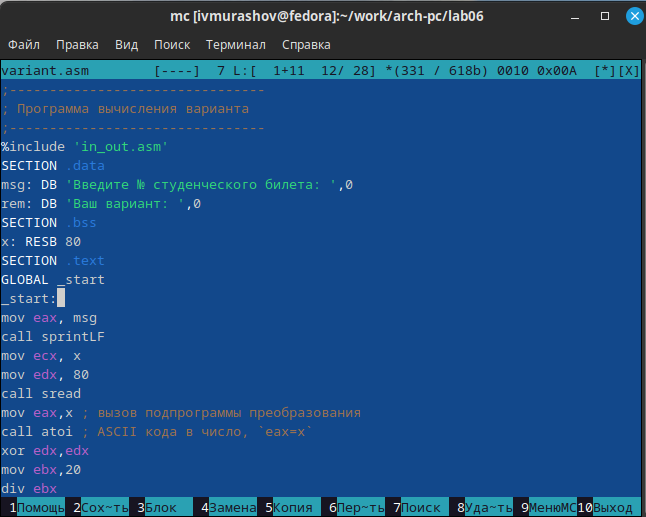
Трансляция, компоновка и запуск файлов

Создаю файл variant.asm в каталоге ~/work/arch-pc/lab06 (рис. [??]).

Создание файла

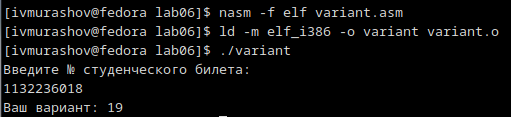
Создание файла

Ввожу в файл текст программы из листинга 6.4 (рис. [??]).



Окно Midnight Commander. Редактор mcedit

Создаю исполняемый файл и запускаю его. Проверяю результат работы программы, вычислив номер своего варианта аналитически. Результаты совпали (рис. [??]).



Трансляция, компоновка и запуск файлов

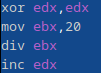
### 3.2.1 Ответы на вопросы по программе

1. За вывод на экран сообщения ‘Ваш вариант:’ отвечают следующие строки:

Окно Midnight Commander. Редактор mcedit

Окно Midnight Commander. Редактор mcedit

1. Данные инструкции используются для того, чтобы поместить адрес вводимой строки в регистр ecx и записать в регистр edx длину вводимой строки (80 bytes). Затем мы вызываем подпрограмму из внешнего файла in\_out.asm для ввода значения x с клавиатуры.
2. Инструкция ‘call atoi’ используется для перевода ASCII-код символа в целое число.
3. Зы вычисление варианта отвечают следующие строки листинга 6.4:



Окно Midnight Commander. Редактор mcedit

1. Остаток от деления при выполнении инструкции ‘div ebx’ записывается в регистр edx.
2. Инструкция ‘inc edx’ используется для прибавления 1 к значению, записанному в регистре edx.
3. За вывод на экран результата вычислений отвечают следующие строки листинга 6.4:

Окно Midnight Commander. Редактор mcedit

Окно Midnight Commander. Редактор mcedit

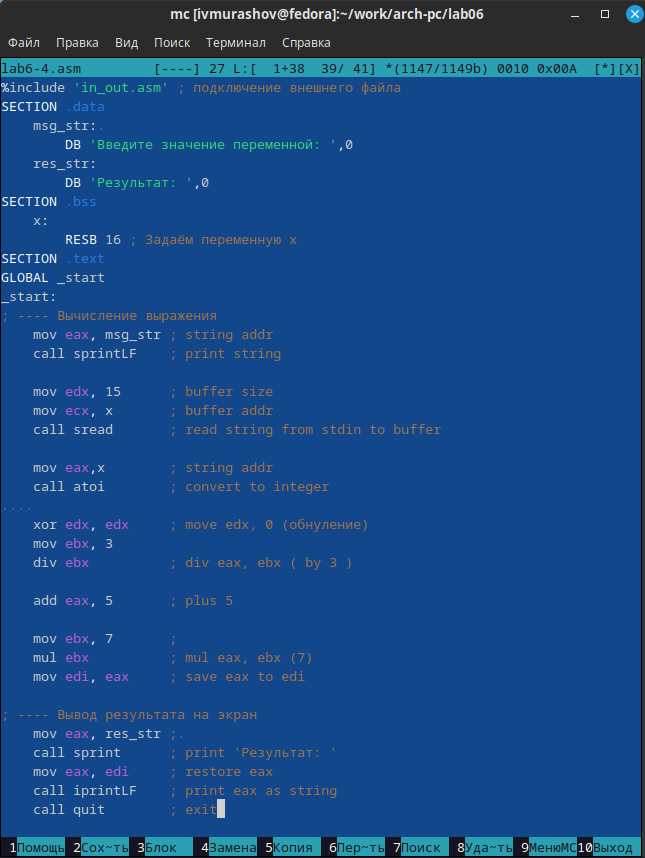
## 3.3 Выполнение заданий для самостоятельной работы

Создаю файл lab6-4.asm в каталоге ~/work/arch-pc/lab06 (рис. [??]).

Создание файла

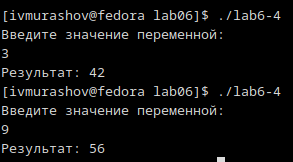
Создание файла

Открываю созданный файл с помощью редактора mc для редактирования и ввожу в него текст программы для вычисления выражения (((1/3)*x + 5)*  7) (19 вариант) (рис. [??]).



Окно Midnight Commander. Редактор mcedit

Создаю исполняемый файл и запускаю его. Проверяю корректность работы программы, вычислив получаемые значения вручную. Результаты совпали (рис. [??]).



Трансляция, компоновка и запуск файлов

**Листинг 4. Программа для вычисления значения выражения (((1/3)*x + 5)*  7)**

%include 'in\_out.asm' ; подключение внешнего файла  
SECTION .data  
 msg\_str:   
 DB 'Введите значение переменной: ',0  
 res\_str:  
 DB 'Результат: ',0  
SECTION .bss  
 x:  
 RESB 16 ; Задаём переменную x  
SECTION .text  
GLOBAL \_start  
\_start:  
; ---- Вычисление выражения  
 mov eax, msg\_str ; string addr  
 call sprintLF ; print string  
  
 mov edx, 15 ; buffer size  
 mov ecx, x ; buffer addr  
 call sread ; read string from stdin to buffer  
  
 mov eax,x ; string addr  
 call atoi ; convert to integer  
   
 xor edx, edx ; move edx, 0 (обнуление)  
 mov ebx, 3  
 div ebx ; div eax, ebx ( by 3 )  
  
 add eax, 5 ; plus 5  
  
 mov ebx, 7 ;  
 mul ebx ; mul eax, ebx (7)  
 mov edi, eax ; save eax to edi  
  
; ---- Вывод результата на экран  
 mov eax, res\_str ;   
 call sprint ; print 'Результат: '  
 mov eax, edi ; restore eax  
 call iprintLF ; print eax as string  
 call quit ; exit

# 4 Выводы

В ходе выполнения данной лабораторной работы я освоил арифметические инструкции языка ассемблера NASM.