Отчёт по лабораторной работе №6

Арифметические операции в NASM.

Грачева Мария Валерьевна

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM

# 2 Теоретическое введение

Большинство инструкций на языке ассемблера требуют обработки операндов. Адрес операнда предоставляет место, где хранятся данные, подлежащие обработке. Это могут быть данные хранящиеся в регистре или в ячейке памяти. Далее рассмотрены все существующие способы задания адреса хранения операндов – способы адресации.

Существует три основных способа адресации:

• *Регистровая адресация* – операнды хранятся в регистрах и в команде используются имена этих регистров, например: mov ax,bx. • *Непосредственная адресация* – значение операнда задается непосредственно в ко- манде, Например: mov ax,2. • *Адресация памяти* – операнд задает адрес в памяти. В команде указывается символи- ческое обозначение ячейки памяти, над содержимым которой требуется выполнить операцию.

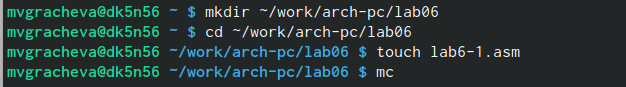
**Арифметические операции в NASM**

Сложение - add Вычитание - sub Умножение - mul Деление - div

Для выполнения лабораторных работ в файле in\_out.asm реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Это: • *iprint* – вывод на экран чисел в формате ASCII, перед вызовом iprint в регистр eax необходимо записать выводимое число (mov eax,). • *iprintLF* – работает аналогично iprint, но при выводе на экран после числа добавляет к символ перевода строки. • *atoi* – функция преобразует ascii-код символа в целое число и записает результат в регистр eax, перед вызовом atoi в регистр eax необходимо записать число (mov eax,).

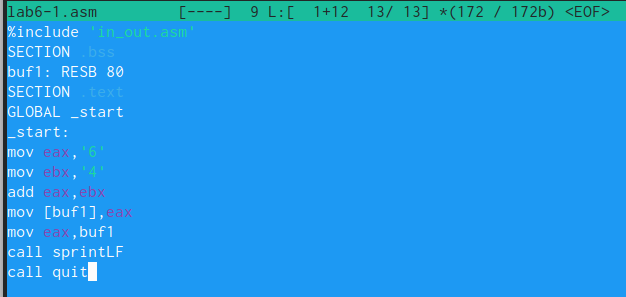
# 3 Выполнение лабораторной работы

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



Создание каталога

Ввожу текст для программы вывода значения регистра eax(рис. ??).

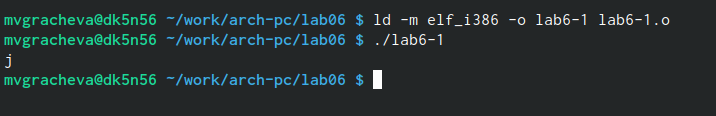


Введение текста

Создаю исполняемый файл и запускаю его (рис. ??), (рис. ??).

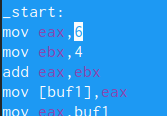
Создание исполняемого файла

Создание исполняемого файла

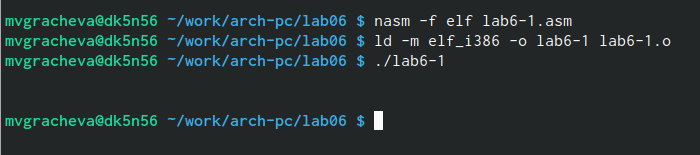


Запуск файла

Заменяю текст программы (рис. ??), должны получить символ 49-1 48-0, то есть 4948, но в итоге ничего не вывелось (рис. ??).

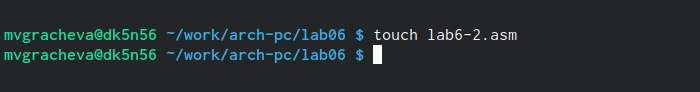


Изменение текста

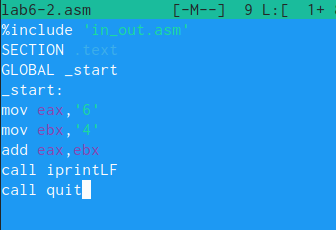


Запуск файла 2

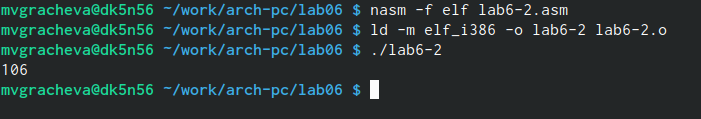
Создаю файл lab06-2 (рис. ??), ввожу в неё программу (рис. ??), создаю исполняемый файл и проверяю работу (рис. ??).



Создание файла lab06-2

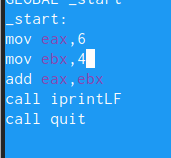


Введение текста 2



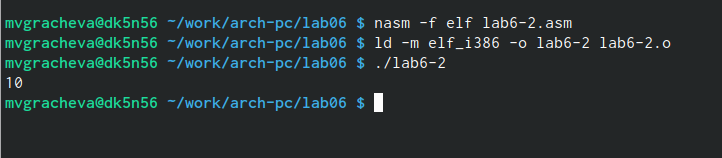
Запуск файла 3

Меняю символы на числа (рис. ??).



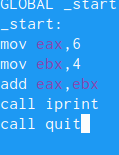
Замена символов

Создаю исполняемый файл. Получаем 10, программа работает верно! (рис. ??).

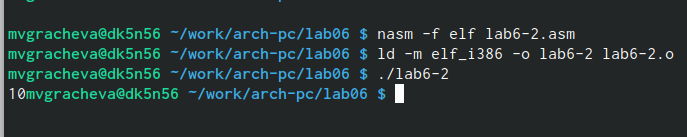


Запуск файла 4

Заменияю iprintLF на iprint(рис. ??), создаю исполянемый файл и проверяю работу. Отличие заключается в том, что нет перехода на новую строку после завершения программы (рис. ??).



Замена iprintLF на iprint

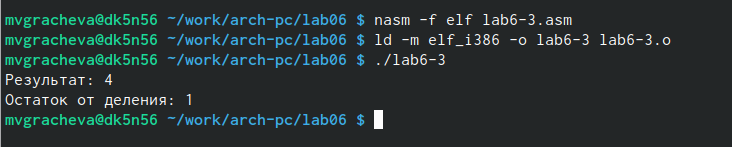


Запуск файла 5

Создаю файл lab6-3.asm (рис. ??), ввожу текст и проверяю работу (рис. ??).

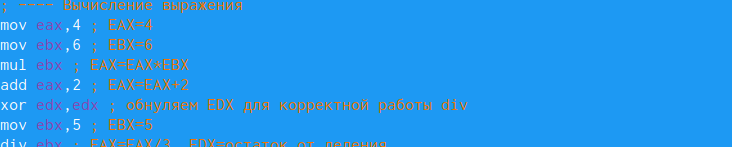
Создание файла lab06-3

Создание файла lab06-3

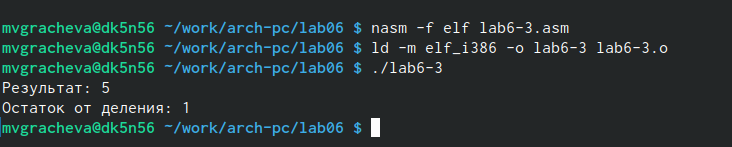


Запуск файла 6

Изменяю текст программы для вычисления выражения 𝑓(𝑥) = (4 ∗ 6 + 2)/5 (рис. ??). Создаю исполняемый файл и проверяю его работу (рис. ??).



Изменение текста 2

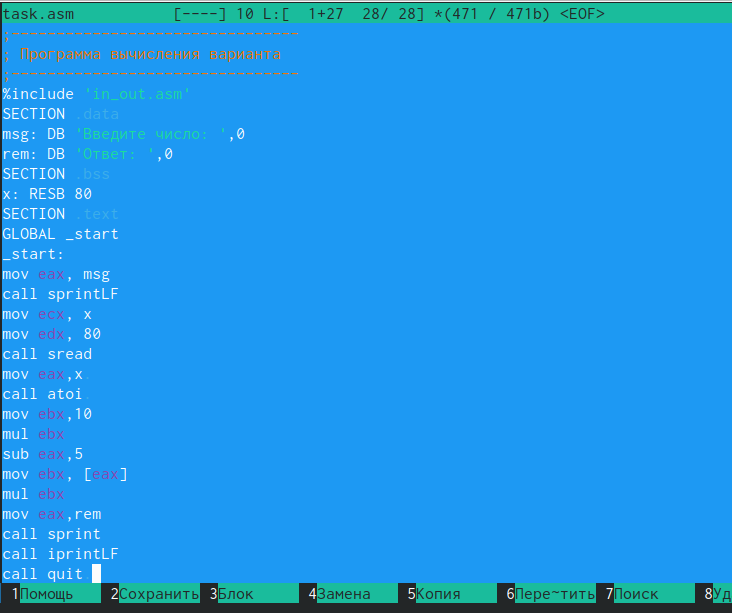


Запуск файла 7

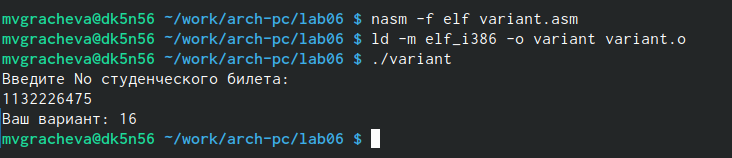
Создаю файл (рис. ??), ввожу текст программы (рис. ??), проверяю работу. Получился 16 вариант (рис. ??).

Создание файла variant.asm

Создание файла variant.asm



Программа для вычисление варианта



Получение варианта

# 4 Ответы на вопросы

1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’?

msg: DB ‘Введите No студенческого билета:’,0

и

mov eax, msg call sprintLF

1. Для чего используется следующие инструкции? mov ecx, x mov edx, 80 call sread

Для того, чтобы прочитать то значение х, которое ввёл пользователь

1. Для чего используется инструкция “call atoi”?

Для преобразования ASCII кода в число

1. Какие строки листинга 6.4 отвечают за вычисления варианта?

xor edx,edx mov ebx,20 div ebx inc edx

1. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”?

edx

1. Для чего используется инструкция “inc edx”?

Для того, чтобы занести в регистр edx остаток от деления

1. Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений?

rem: DB ‘Ваш вариант:’,0

и

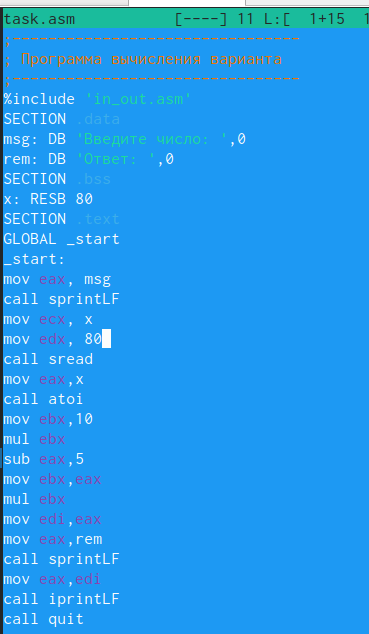
mov eax,rem call sprint mov eax,edx call iprintLF

# 5 Самостоятельная работа

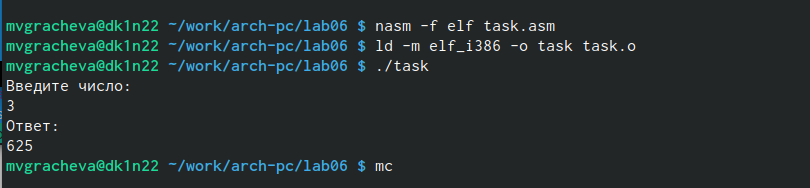
Создаю файл task.asm (рис. ??), ввожу программу (рис. ??), ввожу первое значение - 3 (рис. ??), ввожу второе значение - 1 (рис. ??).

Создание файла task.asm

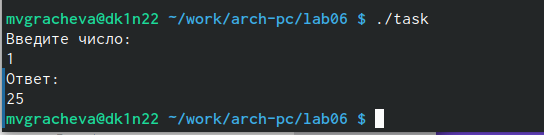
Создание файла task.asm



Написание программы



Проверка для первого значения - 3



Проверка для второго значения - 1

# 6 Выводы

Освоила арифметические инструкции языка ассемблера NASM

# Список литературы