Лабораторная работа №6.

Арифметические операции в NASM.

Самарханова Полина Тимуровна

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Задание

Символьные и численные данные в NASM Выполнение арифметических операций в NASM Задания для самостоятельной работы

# 3 Теоретическое введение

Большинство инструкций на языке ассемблера требуют обработки операндов. Адрес опе- ранда 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.

# 4 Выполнение лабораторной работы

## 4.1 Символьные и численные данные в NASM

С помощью команды mkdir я создала новую директорию,в которой далее создавала файлы с программами во время всей лабораторной работы №6 (рис. ??).

Создание каталога

Создание каталога

Далее я создала файл lab6-1.asm,используя команду”touch” (рис. ??).

Создание файла

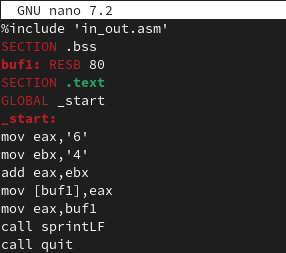
Создание файла

Скопировала в созданный каталог файл in\_out.asm,т.к он использовался и в других программах (рис. ??).

Копирование файла

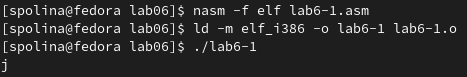
Копирование файла

После я открыла файл lab6-1.asm в nano и вставила в него программу ввода значения регистра eax (рис. ??).



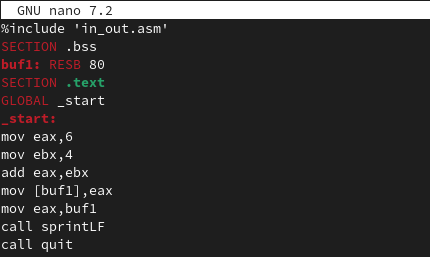
Редактирование файла

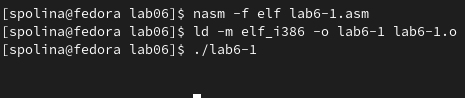
Далее я создала объектный файл и после его компановки запустила программу (рис. ??). Программа вывела символ “j”, т.к программа вывела символ, который соответствует в системе ASCII сумме двоичных символов 4 и 6.



Запуск программы

После я изменила символы ‘6’ и ‘4’ на цифры 6 и 4 (рис. ??).

 ` Далее я создала новый исполняемый файл и запустила программу (рис. ??). Программа вывела символ с кодом 10,это символ перевода строки, он не отображается при выводе на экран.



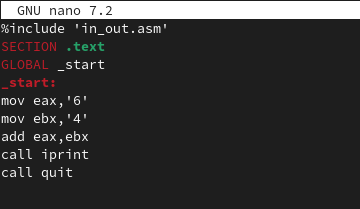
Запуск исполняемого файла

После я создала новый файл под названием lab6-2.asm и проверила его наличие(рис. ??).

Создание нового файла

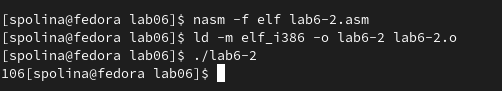
Создание нового файла

После я ввела в файл текст уже другой программы для вывода eax (рис. ??).



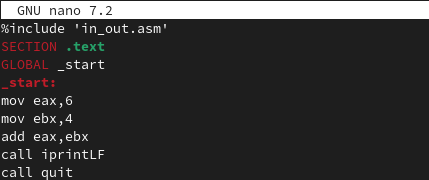
Редактирование программы

Я создала и запустила исполняемый файл lab6-2 (рис. ??). Программа стала выводить 106, т.к программа выводит именно число,не символ, хотя всё еще происходит сложение кодов символов “6” и “4”.



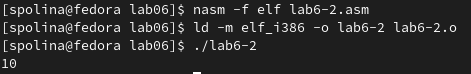
Запуск программы

Я заменила в тексте данной программы символы ‘4’ и ‘6’ на числа 6 и 4 (рис. ??).



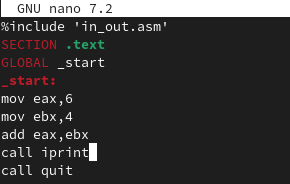
Редактирование программы

Снова создала исполняемый файл программы lab6-2 (рис. ??). Теперь программа складывала именно числа,поэтому выводом является сумма 4+6,которая равна 10.



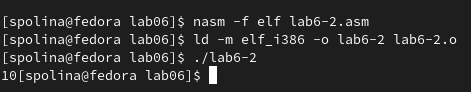
Запуск программы

После я заменила в тексте программы функцию iprintLF на iprint (рис. ??).



Редактирование программы

Создала и запустила исполняемый файл (рис. ??).



Запуск программы

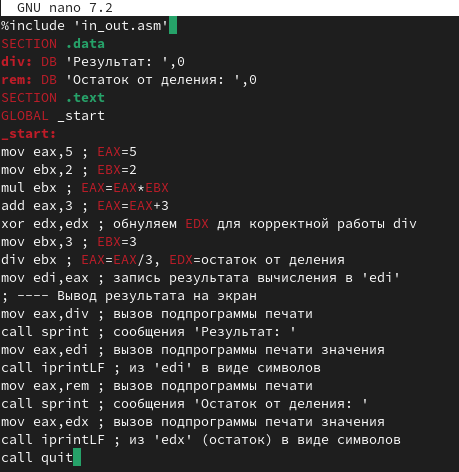
## 4.2 Выполнение арифметических операций в NASM

Я создала файл,назвала его lab6-3.asm, используя команду touch (рис. ??).

Создание файла

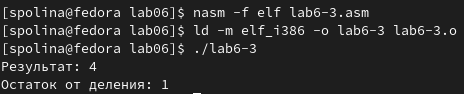
Создание файла

Далее я ввела в созданный файл текст программы для вычисления значения выражения f(x)=(5\*2+3)/3 (рис. ??).



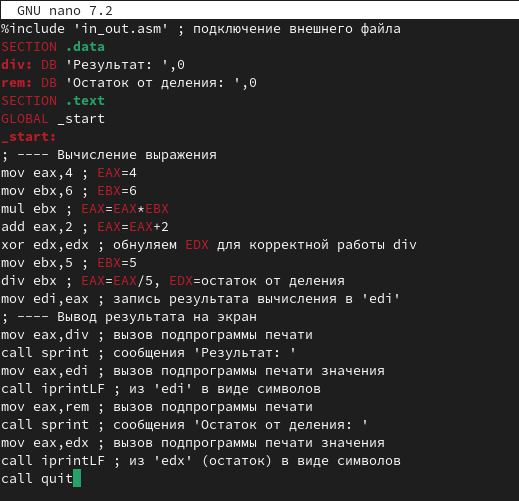
Редактирование программы

Запустила созданный исполняемый файл (рис. ??).



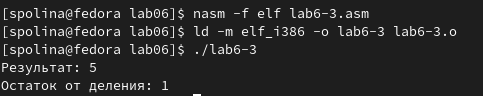
Запуск программы

Изменила программу так, чтобы она вычисляла другое выражение f(x)=(4\*6+2)/5 (рис. ??).



Редактирование программы

Далее я создала исполняемый файл и запустила программу (рис. ??). Программа посчитала выражение правильно.



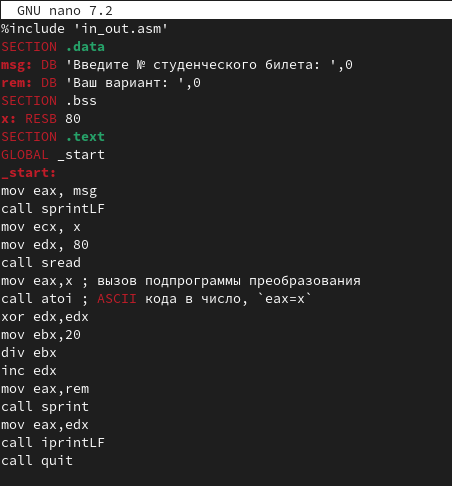
Запуск программы

После создала новый файл variant.asm (рис. ??).

Создание файла

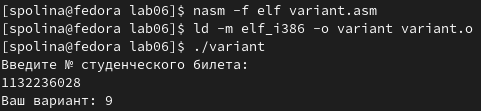
Создание файла

Ввела в файл текст программы, который вычисляет задания по номеру студенческого билета (рис. ??).



Редактирование программы

Создала исполняемый файл и запустила программу (рис. ??). Ввела номер своего студенческого



Запуск программы

### 4.2.1 Ответы на вопросы

1. За вывод сообщения “Ваш вариант” отвечают строчки кода

mov eax,rem  
call sprint

1. Инструкция mov ес, х используется, чтобы положить адрес вводимой строки х в регистр есх mov edx, 80 - запись в регистр edx длины вводимой строки call sread - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры
2. call atoi используется для вызова подпрограммы из внешнего файла, которая преобразует ascii-код символа в целое число и записывает результат в регистр eax
3. За вычисления варианта отвечают строки:

хот edx,edx ; обнуление еах для корректной работы div  
mov ebx, 20 ; ebx = 20  
div ebx; eax = eax/20, edx - остаток от деления  
inc edx; edx = edx + 1

1. При выполнении инструкции div ebx остаток от деления записывается в регистр edx
2. Инструкция inc edx увеличивает значение регистра edx на 1
3. За вывод на экран результатов вычислений отвечают строки:

mov eax,edx  
call iprintLF

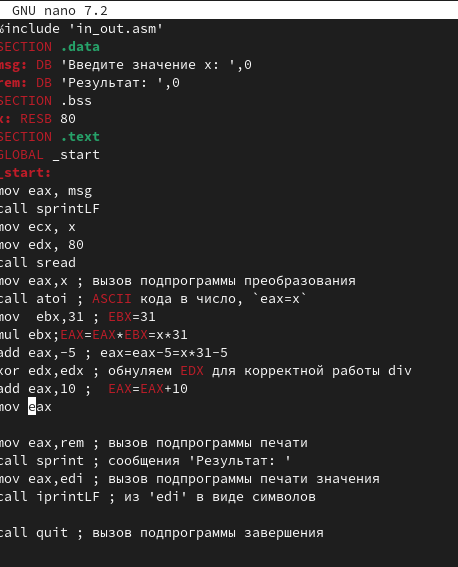
## 4.3 Задания для самостоятельной работы

Я создала файл lab6-4.asm,используя команду touch (рис. ??).

Создание файла

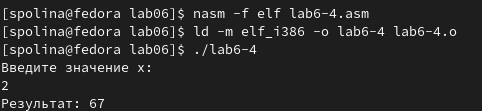
Создание файла

Открыла файл, ввела в него выражения №9 (рис. ??).



Редактирование программы

После создала исполняемый файл и запустила программу (рис. ??). Программа вывела правильный ответ.



Запуск программы

# 5 Выводы

При выполнении данной лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.

# Список литературы

1. Лабораторная работа №6
2. Таблица ASCII