Лабораторная работа No6.

Арифметические операции в NASM.

Бурыкина Софья Дмитриевна

Содержание

# 1 Цель работы

Цель даной лабораторной работы - приобретения практических навыков арифметических инструкций языка ассемблера NASM.

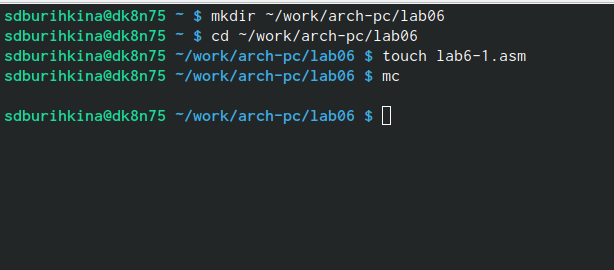
# 2 Теоретическое введение

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. Например, определим переменную intg DD 3 – это означает, что задается область памяти размером 4 байта, адрес которой обозначен меткой intg. В таком случае, команда mov eax,intg копирует из памяти по адресу intg данные в регистр eax. В свою очередь команда mov intg,eax запишет в память по адресу intg данные из регистра eax. Также рассмотрим команду mov eax,intg

Более подробно об Unix см. в [1–6].

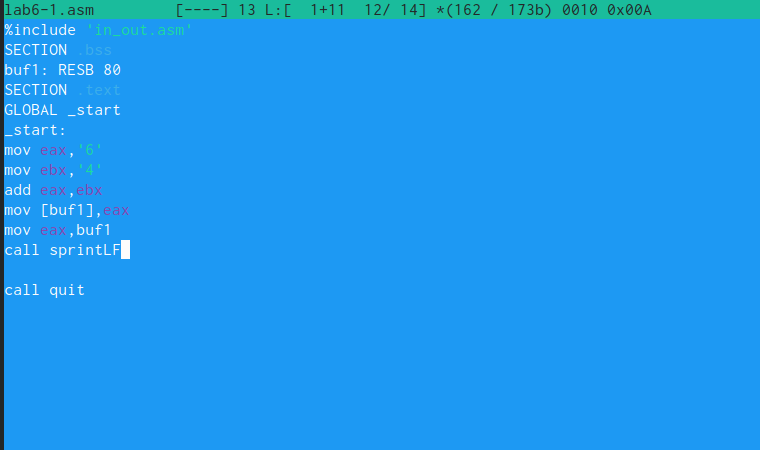
# 3 Выполнение лабораторной работы

Создала каталог для программам лабораторной работы No 6, перешла в него и создала файл lab6-1.asm (рис. ??).



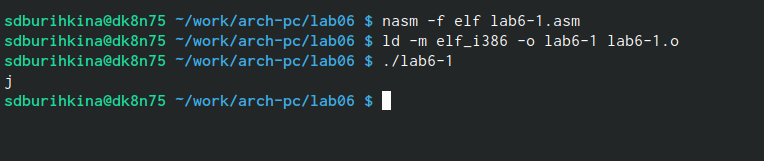
Создания файла

Создала исполняемый файл (рис. ??).



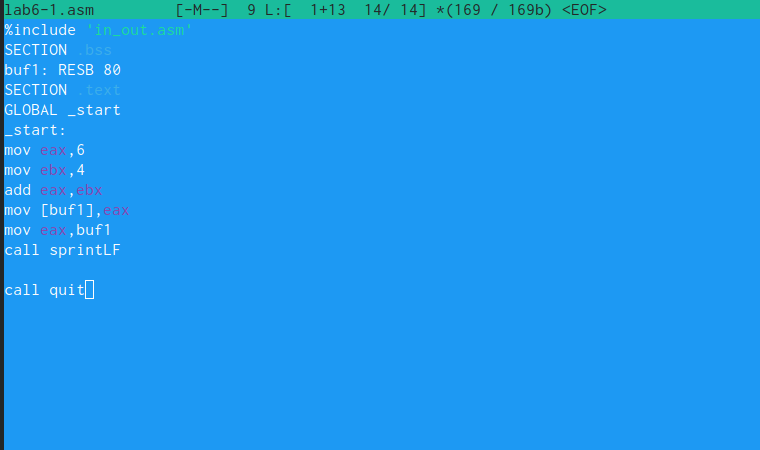
Ввела программу

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



Запуск программы

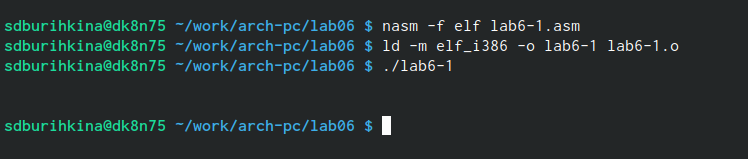
Изменила текст программы (рис. ??).



Исправила текст

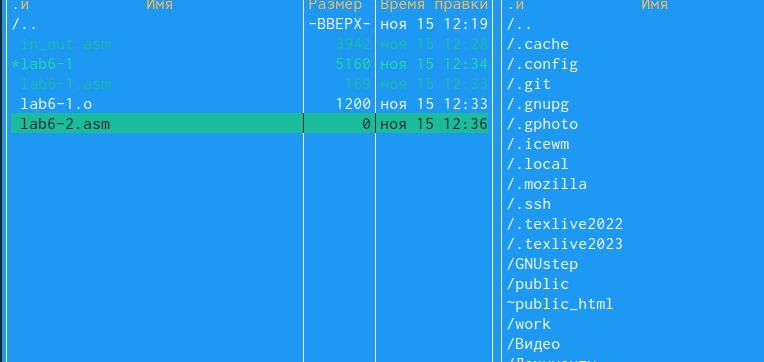
Создала исполняемый файл после изменения и запустила его (рис. ??).

Ответ на вопрс: нет не отображается ничего



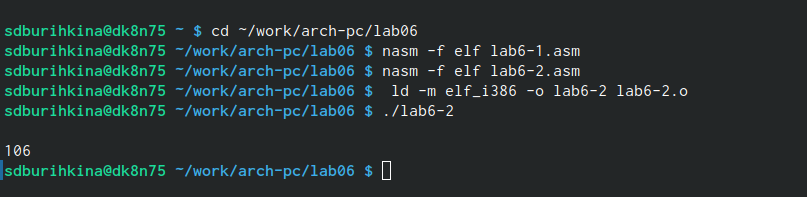
Запустила программу

Создала файл lab6-2.asm в каталоге (рис. ??).



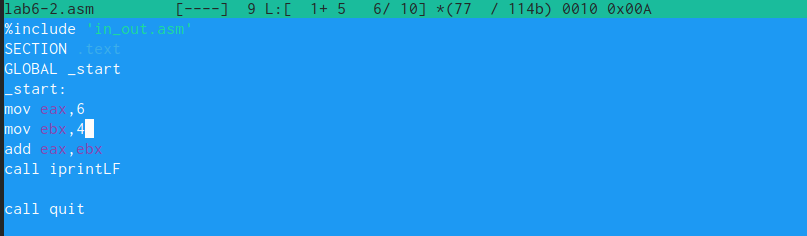
Созданный файл

Создала испольняемый файл (рис. ??).



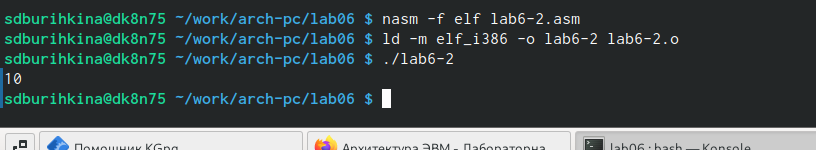
Запуск программы

Изменила содержимое файла (рис. ??).



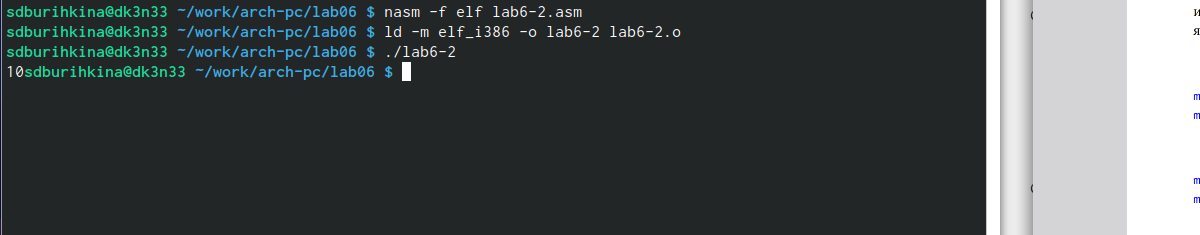
Запуск программы

Полученный пезультат 10 при исполнение программы (рис. ??).



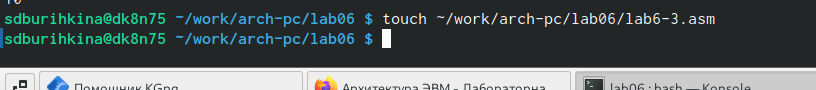
Результат программы

Заменила функцию iprintLF на iprint. Получила результат. Отличаеться выводом спроки (рис. ??).



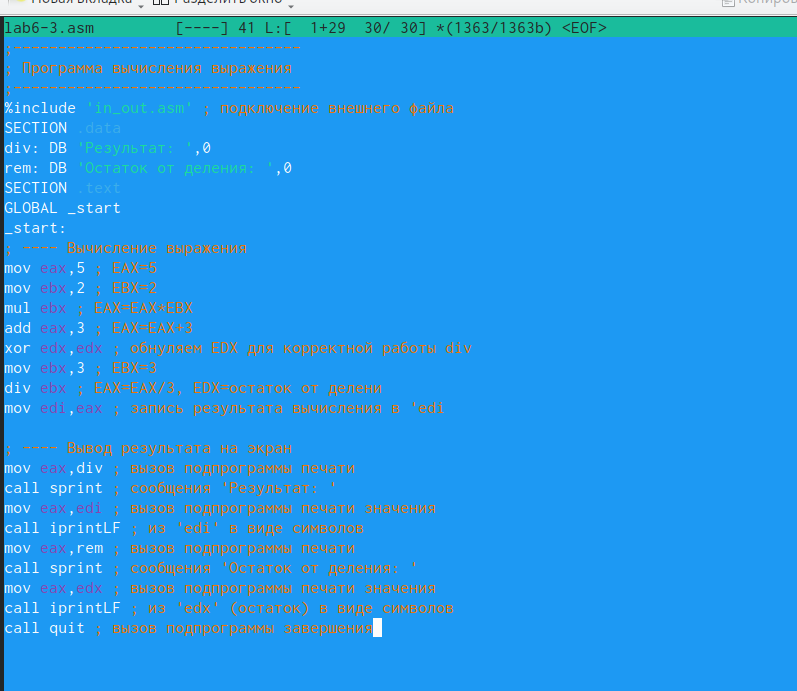
Результат программы

Создала файл lab6-3.asm в каталоге (рис. ??).



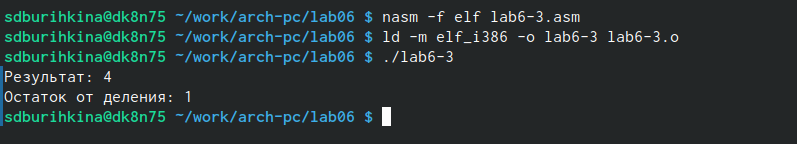
Создание файла

Ввелла тект программы (рис. ??).



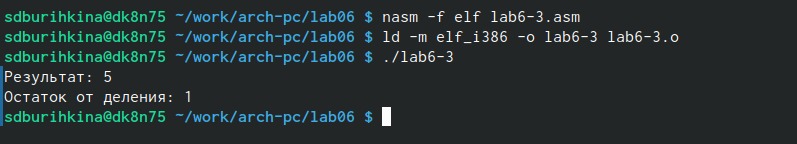
Файл с текстом

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



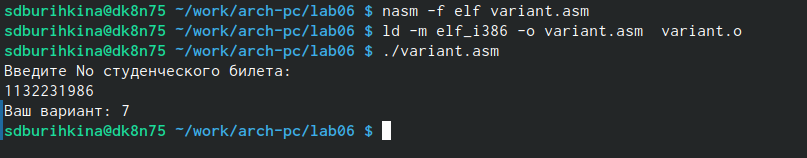
Результат работы

Изменила текст программы для вычисления выражения (рис. ??).



Результат работы

Создала файл variant.asm в каталоге (рис. ??).



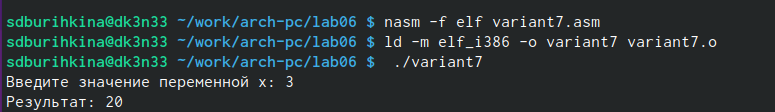
Результат работы

# 4 Ответы на вопросы

1 Ответ: sprint выводит сообщения на экран, перед вызовом sprint в регистр eax необходимо записать выводимое сообщение (mov eax,msg).Также sprintLF работает аналогично sprint, но при выводе на экран добавляет к сообщению символ перевода строки. 2 Ответ: mov ecx, x запись адреса переменной ‘EAX’ mov edx, 80 запись длины вводимого сообщения в `EBX’ call sread вызов подпрограммы ввода сообщения 3 Ответ: call - вызов, atoi – функция преобразует ascii-код символа в целое число и записыает результат в регистр eax, перед вызовом atoi в регистр eax необходимо записать число (mov eax,int). 4 Ответ: call sread вывод программы ввода сообщения. 5 Ответ: процессор поделит число, старшие биты которого хранит регистр edx, а младшие eax на значение, хранящееся в регистре ebx. 6 Ответ: для вызова прерывания с указанным номером такой как edx. 7 Ответ: call iprintLF; call quit

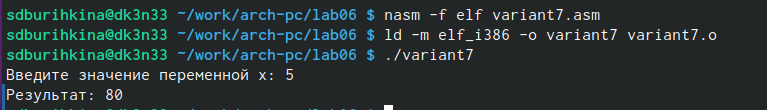
# 5 Cамостоятельная работа

Создала исполняемый файл и проверила его работу Ответ x1 (рис. ??).



Результат работы

Ответ x2 (рис. ??).



Результат работы

# 6 Выводы

Приобрела практические знания арифметических инструкций языка ассемблера NASM.

# Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016. URL: <https://www.gnu.org/software/bash/manual/>.

2. Newham C. [Learning the bash Shell: Unix Shell Programming](http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658). O’Reilly Media, 2005. 354 с.

3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.

4. Robbins A. [Bash Pocket Reference](https://www.ncbi.nlm.nih.gov/pubmed/25246403). O’Reilly Media, 2016. 156 с.

5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.

6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.