Лабораторная работа номер 6.

Арифметические операции в NASM

Сорокин Кирилл

Содержание

# 1 Цель работы

Научиться писать арифметические инструкции языка ассемблер

# 2 Задание

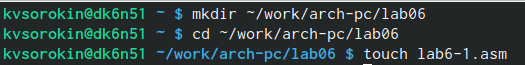
Изучить приведённый материал на практике и выполнить самостоятельную работу.

# 3 Теоретическое введение

Схема команды целочисленного сложения add (от англ. addition - добавление) выполняет сложение двух операндов и записывает результат по адресу первого операнда. Команда целочисленного вычитания sub (от англ. subtraction – вычитание) работает аналогично команде add.Довольно часто при написании программ встречается операция прибавления или вычитания единицы. Прибавление единицы называется инкрементом, а вычитание декрементом. Для этих операций существуют специальные команды: inc (от англ. increment) и dec (от англ.decrement), которые увеличивают и уменьшают на 1 свой операнд.

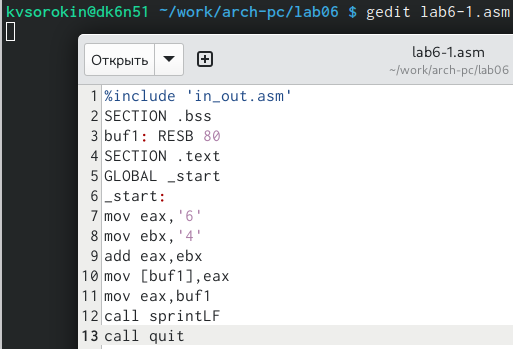
# 4 Выполнение лабораторной работы

Создадим необходимые для работы директории и файлы (рис. ??).



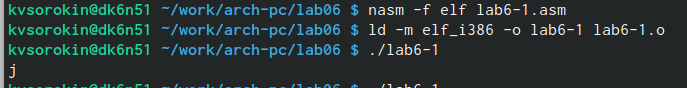
Создание файлов и директорий

Откроем файл lab6-1.asm и введём в него текст программы(рис. ??).



Текст первой программы

После компиляции файлов запустим программу и увидим, нежелаемый результат - j, вместо 10(рис. ??).



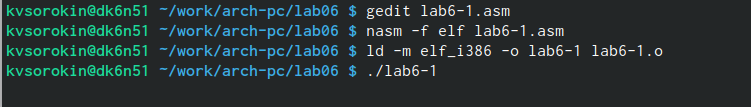
Первая попытка выполнить

Уберём ковычки у числе в тексте программы(рис. ??).

Редактирование чисел

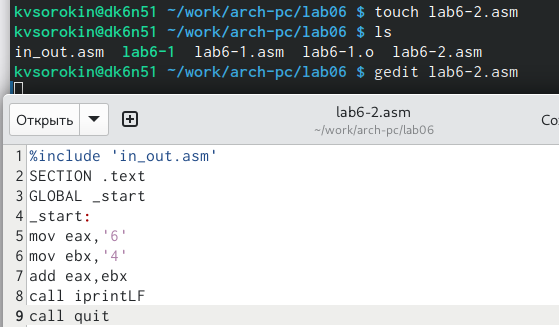
Редактирование чисел

Попробуем ещё раз выполнить программу и увидим, что вывелся символ с кодом 10, а не 10(рис. ??).



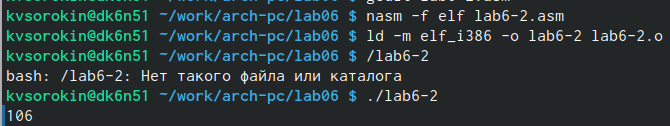
Вторая попытка выполнить

Создадим файл lab6-2.asm и введём в него более корректный код(рис. ??).



Редактирование второго файла

После выполнения опять увидим, что вместо числа 10, пишется число 106(рис. ??).



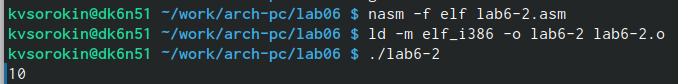
Попытка запуска второго файла

Уберём ковычки у числе в тексте программы (рис. ??).

Изменение файла 2

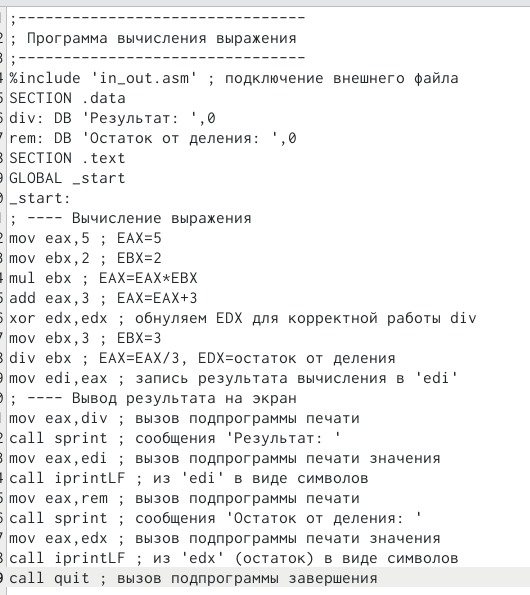
Изменение файла 2

Наконец после выполнения увидим желаемый ответ (рис. ??).



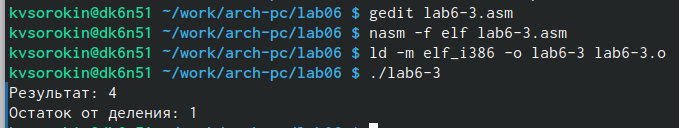
Корректная работа программы

Создадим файл lab6-3.asm и запишем в него программу, выполняющую (5 \* 2 + 3)/3 (рис. ??).



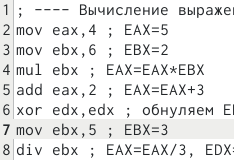
Содержимое файла lab6-3.asm

Выполним программу и убедимся в верности полученного результата (рис. ??).



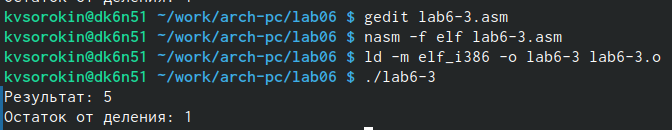
Выполнение программы вычисляющей (5 \* 2 + 3)/3

Заменим значения в программе, чтобы она считала (4\*6 + 2)/5 (рис. ??).



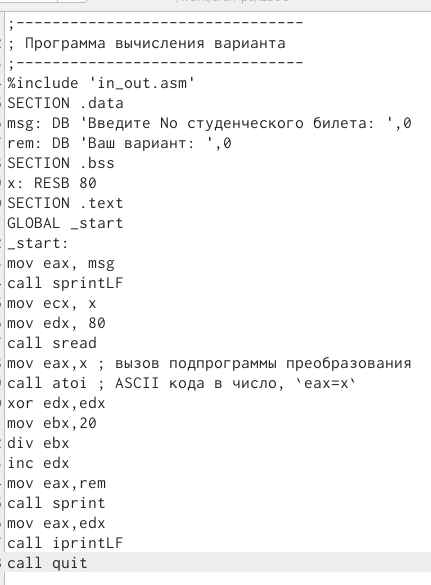
Подстановка других значений

Выполним у убедимся в верности выполнения (рис. ??).

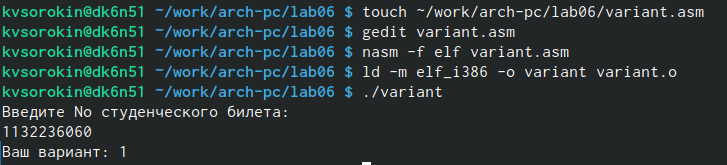


Выполнение изменённой программы

Создадим файл variant.asm и скопируем туда текст программы называющей вариант задания (рис. ??).



Выполним, и узнаем, что по нашему билету получаем первый вариант (рис. ??).



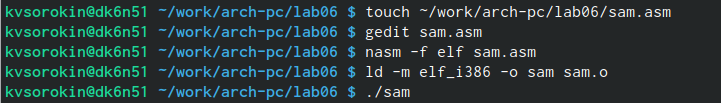
Мой вариант - 1

## 4.1 Ответы на вопросы:

1. ‘mov eax, msg’ и ‘call sprintLF’
2. Для считывания и записывания в информации в Х с максимальной длинной 80
3. Для перевода введённых данных в число
4. Строки ‘xor edx, edx’, ‘mov ebx,20’, ‘div ebx’, ‘inc edx’
5. В edx
6. Для увеличения на значения на 1
7. ‘mov eax, edx’ и ‘call iprintLF’

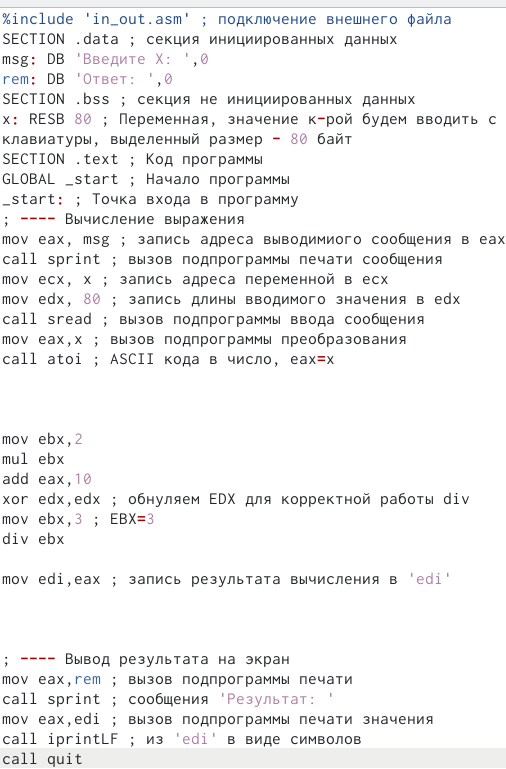
## 4.2 Самостоятельная работа

Создадим файл sam.asm для самостоятельной работы и напишем в нём программу согласно тому, что у нам первый вариант (рис. ??).



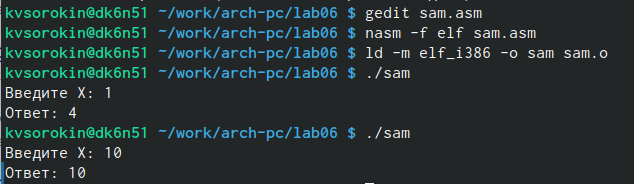
Создание файла sam.asm

Основываясь на полученных занниях напишем программу для вычисления выражения (10+2х)/3 (рис. ??).



Текст самостоятельной работы

Запустим программу и убедимся, что оба введённые значения выдают верный результат (рис. ??).



Работа самостоятельной программы

# 5 Выводы

Мы научились использовать писать простые программы для вычисления на языке ассемблера

# Список литературы

1. GDB: The GNU Project Debugger. — URL: https://www.gnu.org/software/gdb/.
2. GNU Bash Manual. — 2016. — URL: https://www.gnu.org/software/bash/manual/.
3. Midnight Commander Development Center. — 2021. — URL: https://midnightcommander.org/.
4. NASM Assembly Language Tutorials. — 2021. — URL: https://asmtutor.com/.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O’Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: http://www.amazon.com Learning-bash-Shell-Programming-Nutshell/dp/0596009658.
6. Robbins A. Bash Pocket Reference. — O’Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: https://www.nasm.us/docs.php.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. :Солон-Пресс, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: https://www.opennet.ru/docs/RUS/nasm/.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВ- Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm\_unix.
15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. - 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. -СПб. : Питер,
17. — 1120 с. — (Классика Computer Science)