Отчет пол лабораторной работе №6

Простейший вариант

Михальский Кирилл Алексеевич

Содержание

Список иллюстраций

Список таблиц

# Цель работы

Освоение арифметических инструкций языка ассемблера NASM

# Задание

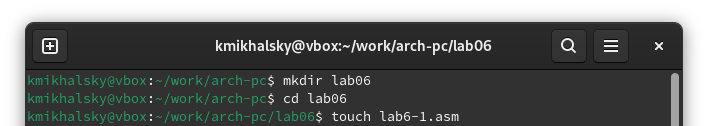
1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM

# Теоретическое введение

Большинство инструкций на языке ассемблератребуютобработки операндов.Адрес опе ранда предоставляетместо,где хранятся данные,подлежащие обработке.Это могутбыть данные хранящиеся в регистре или в ячейке памяти.Далее рассмотрены все существующие способы задания адреса хранения операндов–способы адресации. Существуеттри основных способа адресации: \* Регистровая адресация–операндыхранятся врегистрах 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–операндзадаетадресвпамяти.Вкомандеуказывается символи ческое обозначение ячейки памяти,над содержимым которойтребуется выполнить операцию.

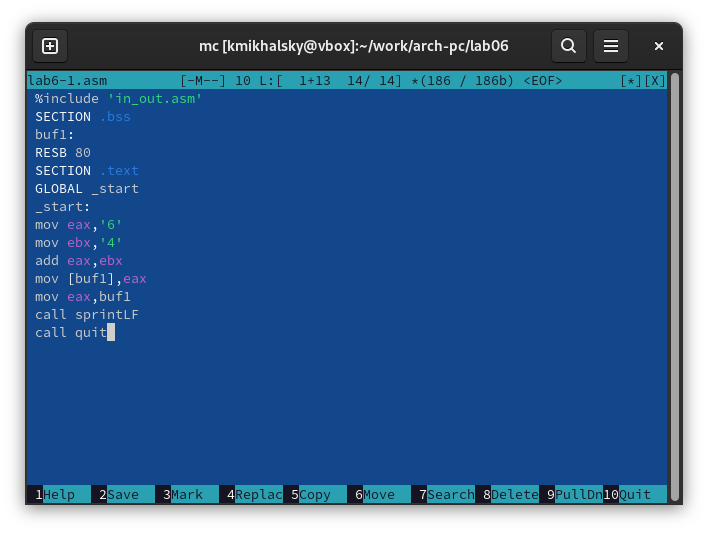
# Выполнение лабораторной работы

1. Создал каталог и файл lab6-1.asm.

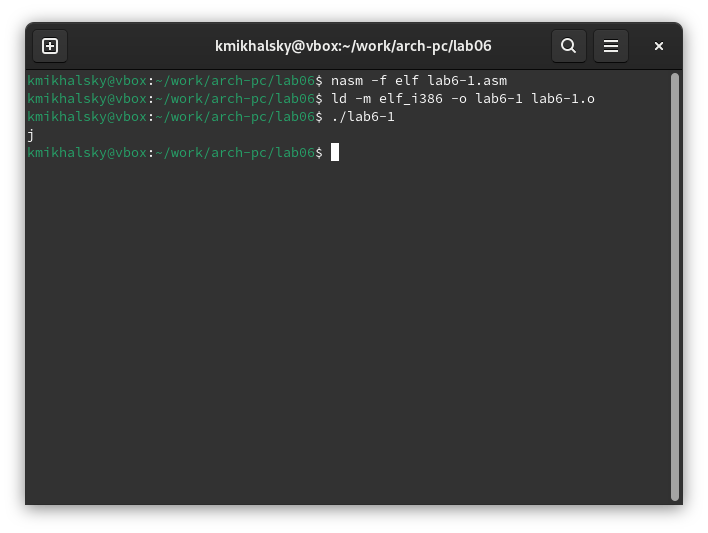


Создание

1. Вставил текст программы и запустил.

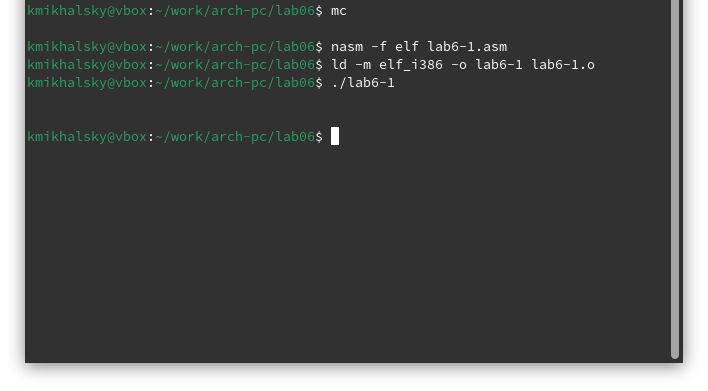


Изменение



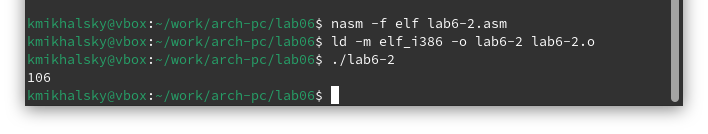
Компиляция

1. Внес необзодимые изменения и запустил. Код соответствует пустому символу перевода строки, который и отображался на экране.



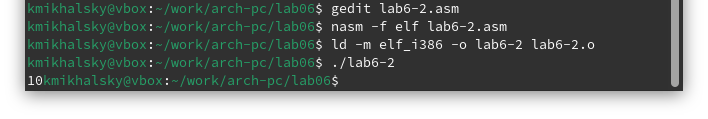
Запуск

1. Создал файл lab6-2.asm, вставил текст программы и запустил исполняемый файл.

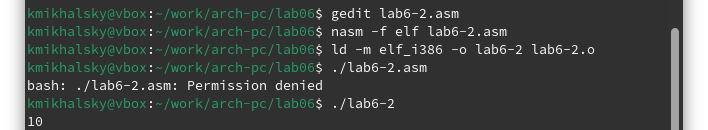


Создание

1. Внес изменения в файл lab6-2.asm. 6 + 4 = 10. Заменил функцию iprintLF на iprint и запустил исполняемый файл. Не отобразился перевод строки.

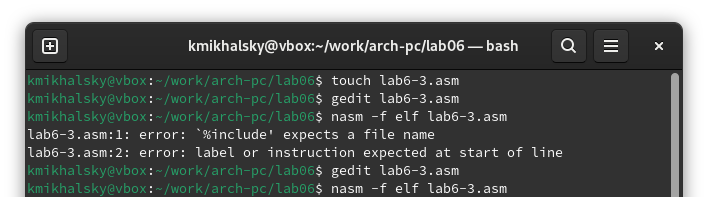


Изменения

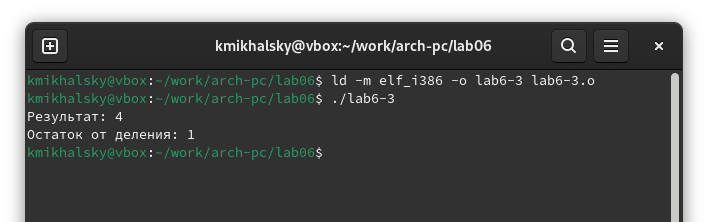


Изменения

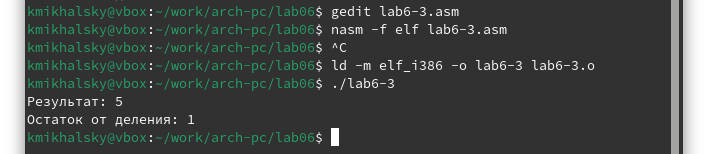
1. Создал файл lab6-3.asm. Ввел текст программы. Исправил текст программы.



Создарние и ввод

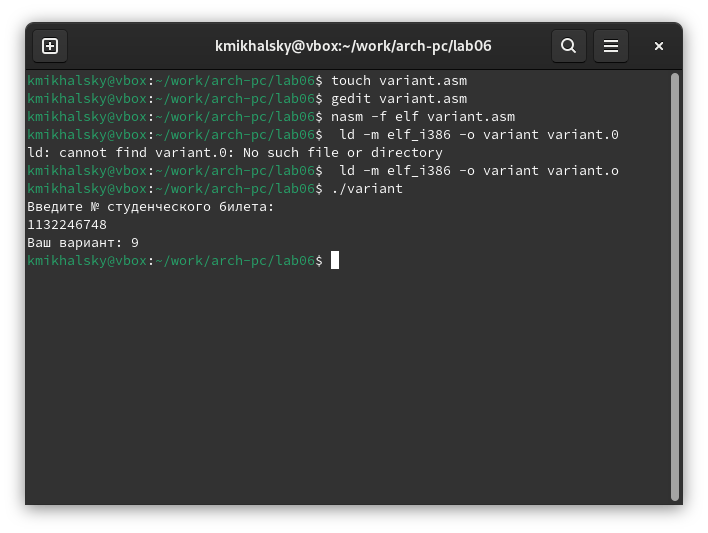


Запуск



Изменение и запуск

1. Создал файл variant.asm, ввел текст и запустил.



Работа с файлом

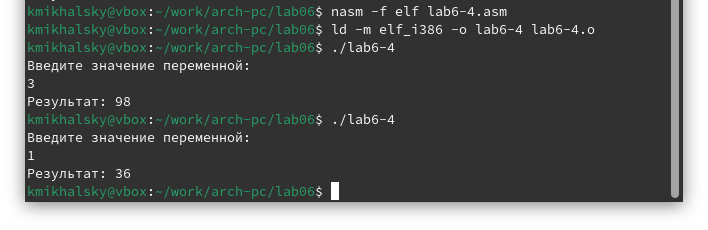
1. mov eax,rem call sprint
2. Инструкция mov ecx, x используется, чтобы положить адрес вводимой строки x в регистр ecx mov edx, 80 - запись в регистр edx длины вводимой строки call sread - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры
3. call atoi используется для вызова подпрограммы из внешнего файла, которая преобразует ascii-код символа в целое число и записывает результат в регистр eax
4. xor edx,edx ; обнуление edx для корректной работы div mov ebx,20 ; ebx = 20 div ebx ; eax = eax/20, edx - остаток от деления inc edx ; edx = edx + 1
5. При выполнении инструкции div ebx остаток от деления записывается в регистр edx
6. Инструкция inc edx увеличивает значение регистра edx на 1
7. mov eax,edx call iprintLF

# Выполнение задания для самостоятельной работы.

1. Копировал файл lab6-4, исправил текст программы для работы по условию, проверил правильность используя переменные из задачи. Вариант 9



Работа с файлом



Работа с файлом

# Выводы

Я освоил арифметические конструкции языка ассемблера.

# Список литературы