Отчет по лабораторной работе №6

Архитектура компьютеров

Ясиновская Дарья Олеговна

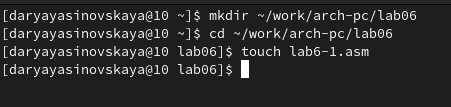
Содержание

# Цель работы

Целью лабораторной работы является освоение арифметических инструкций языка ассемблера NASM

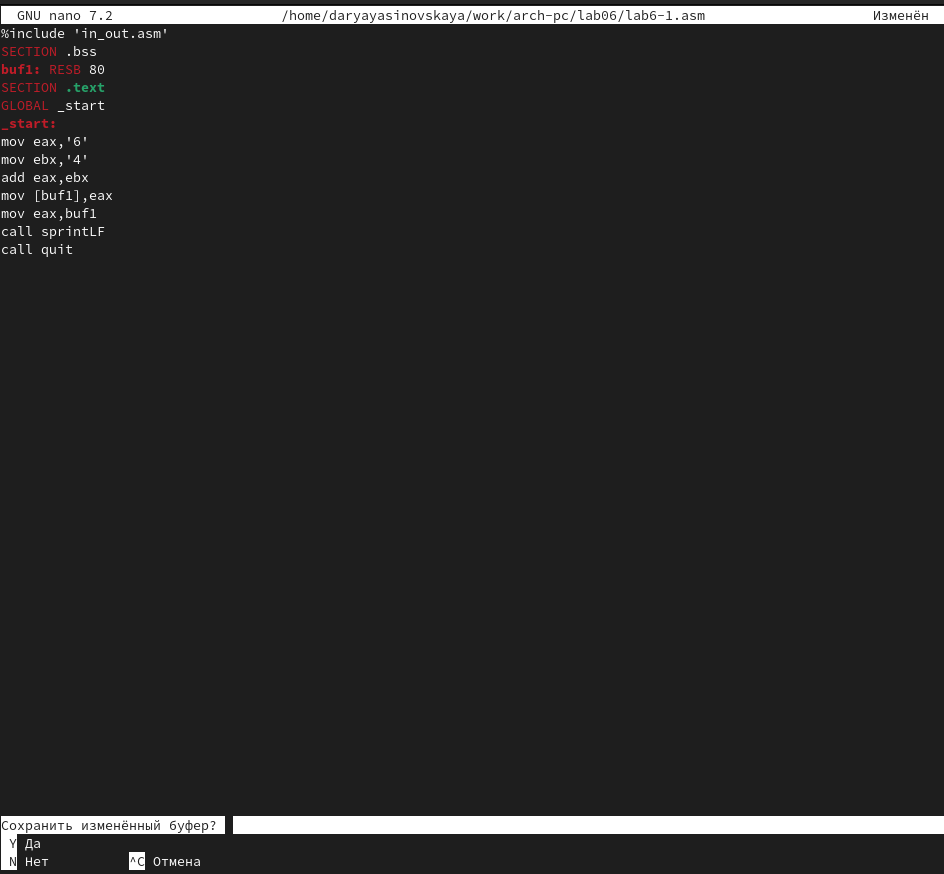
# Выполнение лабораторной работы

Создала каталог для выполнения лабораторной работы №6, перешла в него и создала файл lab6-1.asm



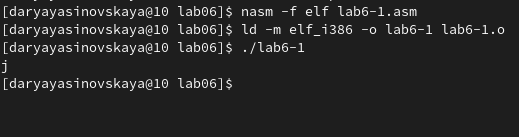
Создание каталога и файла в нем

Ввела в файл lab6-1.asm текст из листинга 6.1



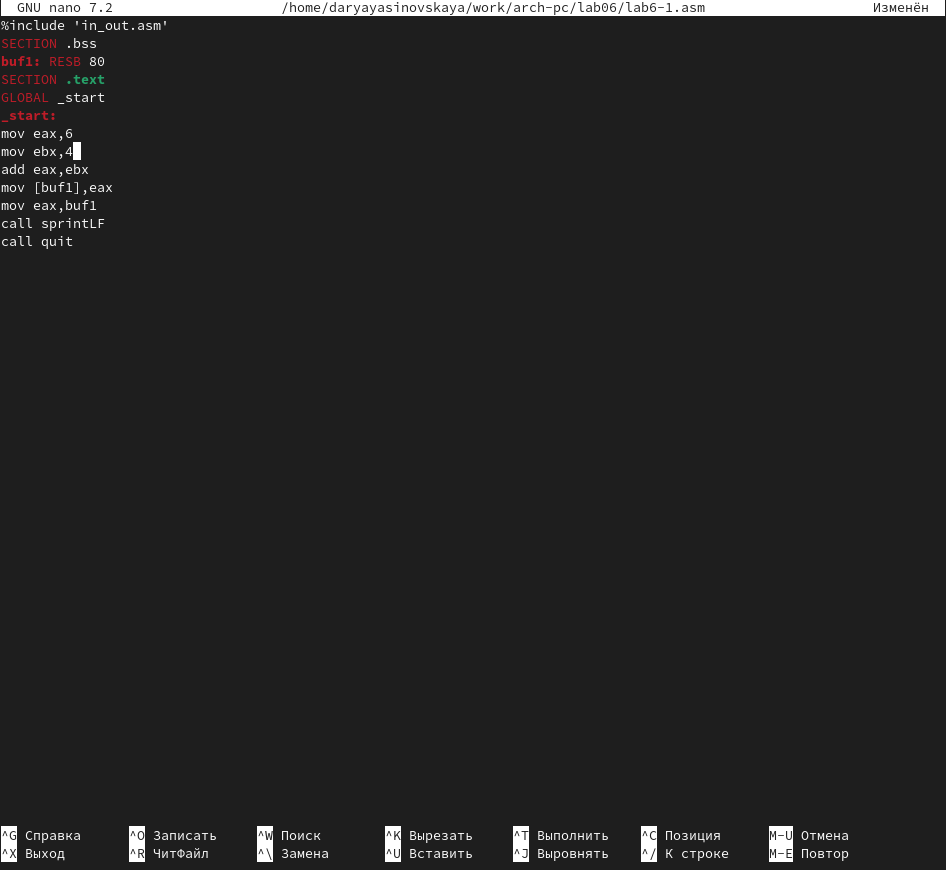
Ввод текста

Создала исполняемый файл и запустила его. В выводе получила символ j.



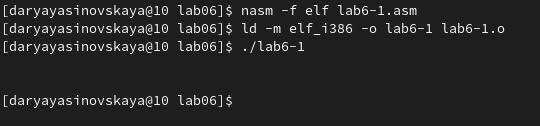
Создание и запуск исполняемого файла

Изменила текст программы, заменив символы на числа



Изменение текста

Создала исполняемый файл и запустила его. Теперь вывелся символ с кодом 10, это символ перевода строки. Этот символ не отображается при выводе на экран.

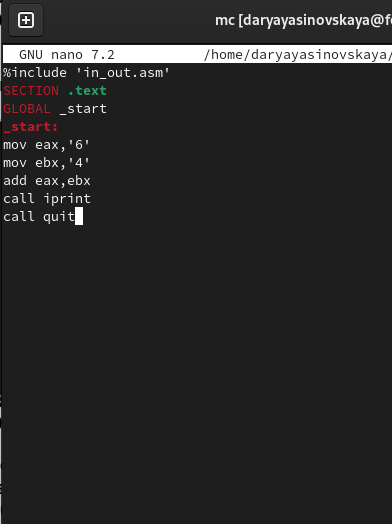


Создание и запуск исполняемого файла

Создала файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и ввела в него текст программы из листинга 6.2

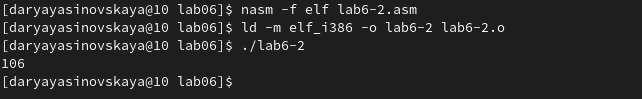
Создание файла

Создание файла



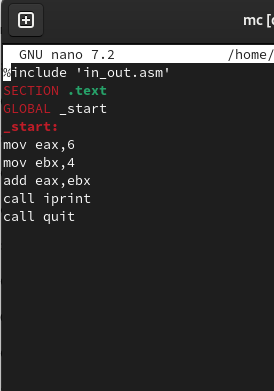
Ввод текста

Создала исполняемый файл и запустила его. В выводе получила число 106



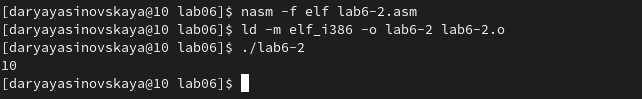
Создание и запуск исполняемого файла

Изменила файл, заменив символы на числа.



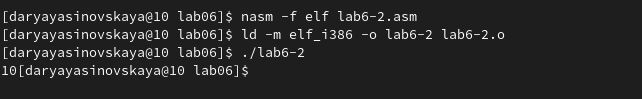
Изменение текста

Создала исполняемый файл и запустила его. В результате выполнения программы получила число 10.



Создание и запуск исполняемого файла

Заменила функцию iprintLF на iprint, затем создала исполняемый файл и запустила его. Теперь следующая команда вводится на той же строке.



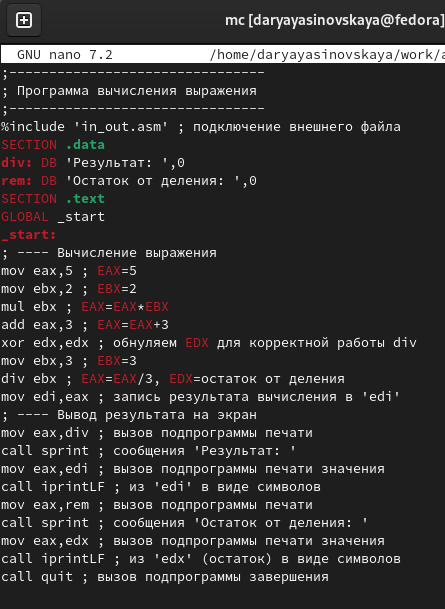
Создание каталога и файла в нем

Создала файл lab6-3.asm в каталоге ~/work/arch-pc/lab06

Создание файла в каталоге

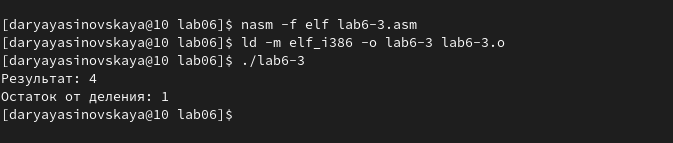
Создание файла в каталоге

ВВела в файл текст из листинга 6.3



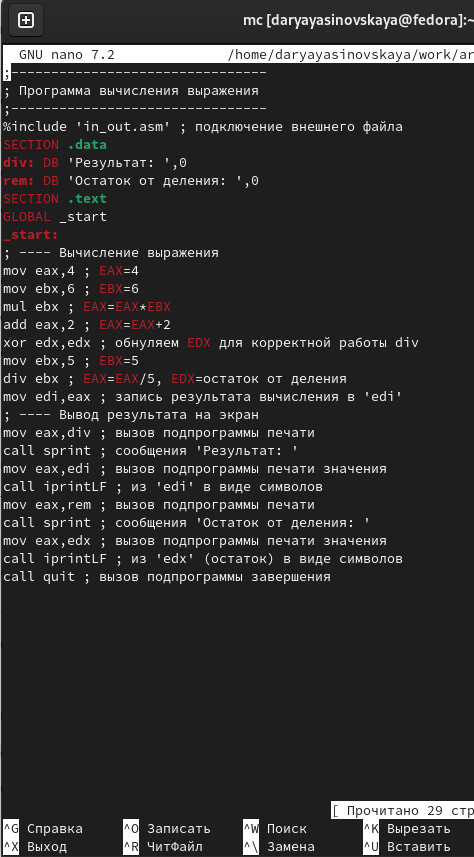
Ввод текста

Создала исполняемый файл и запустила его



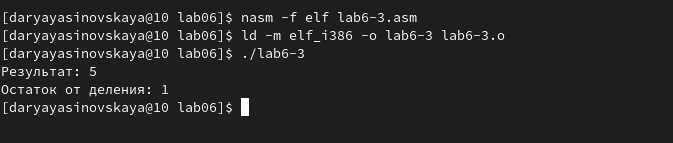
Создание и запуск исполняемого файла

Изменила текст программы для вычисления выражения f(х) = (4 \* 6 + 2)/5



Изменение текста

Создала исполняемый файл и проверила его работу. Теперь после строки “Результат” выводится 5, а после строки “Остаток от деления” выводится 1



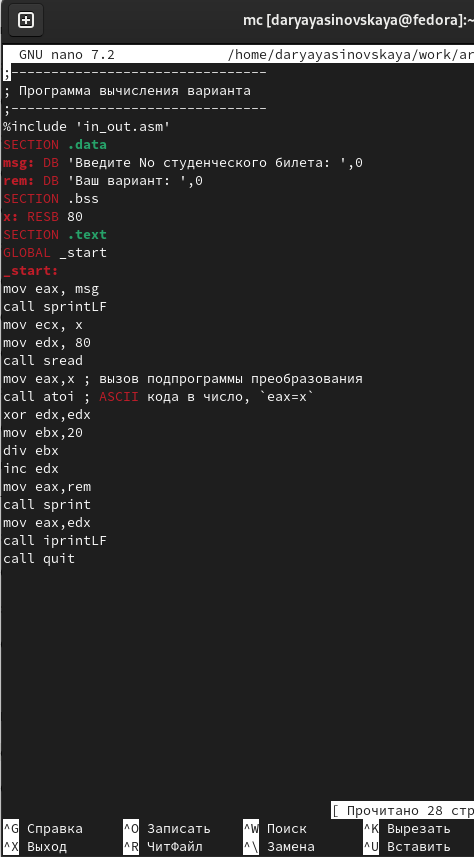
Создание и запуск исполняемого файла

Создаю файл variant.asm

Создание файла

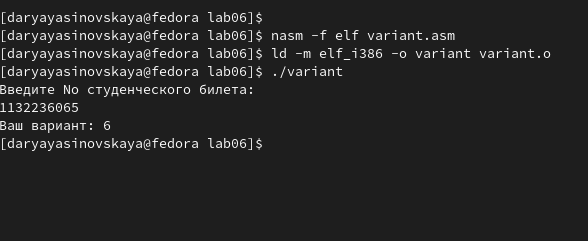
Создание файла

Ввела в файл текст программы из листинга 6.4



Ввод текста

Создала исполняемый файл и запустила его.



Создание и запуск исполняемого файла

# Ответы на вопросы

1. За вывод на экран сообщения “Ваш вариант:” отвечают строки:

mov eax,rem call sprint

1. 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
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:

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

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

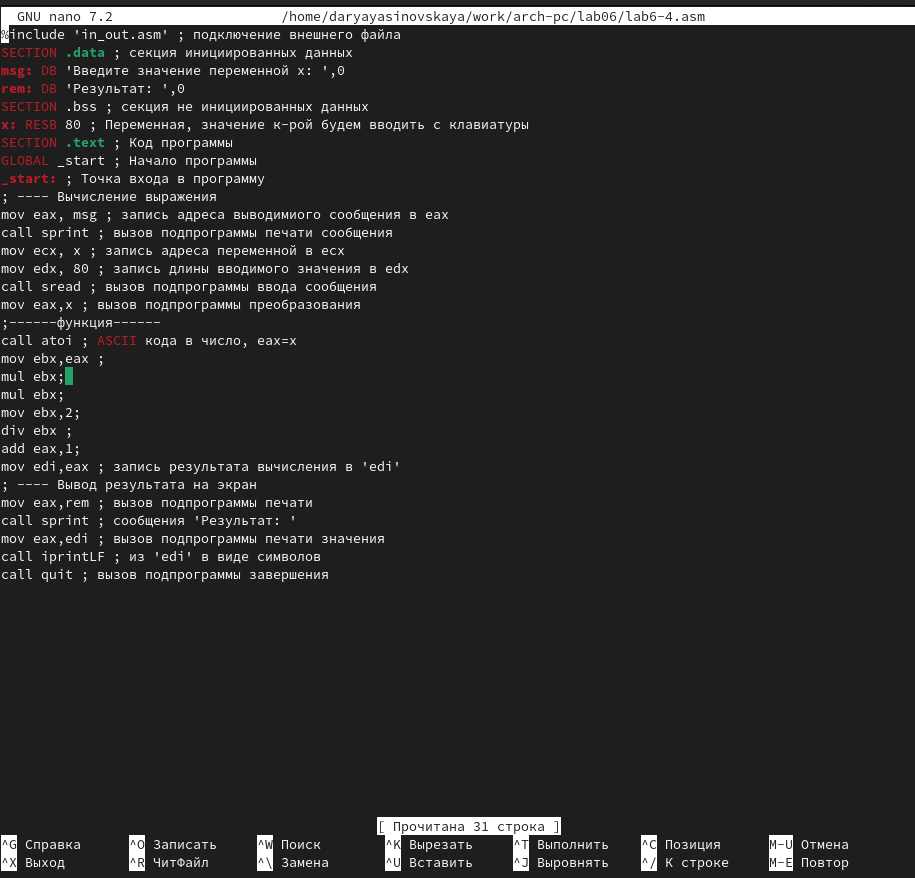
# Самостоятельная работа

Создала файл для программы вычисления выражения y = f(x)

Создание файла

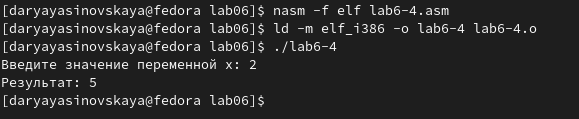
Создание файла

Написала программу для вычисления функции



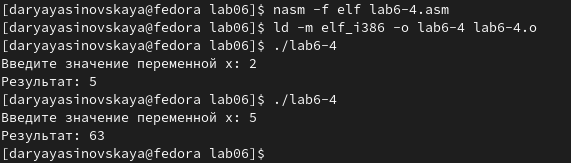
Текст программы

Создала исполняемый файл и проверила его работу для х1



Создание и запуск исполняемого файла

Создала исполняемый файл и проверила его работу для х2



Создание и запуск исполняемого файла

# Листинги программ

Листинг 6.1. Программа вывода значения регистра eax %include ‘in\_out.asm’ SECTION .bss buf1: RESB 80 SECTION .text GLOBAL \_start \_start: mov eax,‘6’ mov ebx,‘4’ add eax,ebx mov [buf1],eax mov eax,buf1 call sprintLF call quit

Листинг 6.2. Программа вывода значения регистра eax %include ‘in\_out.asm’ SECTION .text GLOBAL \_start \_start: mov eax,‘6’ mov ebx,‘4’ add eax,ebx call iprintLF call quit

Листинг 6.3. Программа вычисления выражения f(x) = (5 \* 2 + 3)/3 ;——————————– ; Программа вычисления выражения ;——————————– %include ‘in\_out.asm’ ; подключение внешнего файла SECTION .data div: DB ‘Результат:’,0 rem: DB ‘Остаток от деления:’,0 SECTION .text GLOBAL \_start \_start: ; —- Вычисление выражения mov eax,5 ; EAX=5 mov ebx,2 ; EBX=2 mul ebx ; EAX=EAX\*EBX add eax,3 ; EAX=EAX+3 xor edx,edx ; обнуляем EDX для корректной работы div mov ebx,3 ; EBX=3 div ebx ; EAX=EAX/3, EDX=остаток от деления mov edi,eax ; запись результата вычисления в ‘edi’ ; —- Вывод результата на экран mov eax,div ; вызов подпрограммы печати call sprint ; сообщения ‘Результат:’ mov eax,edi ; вызов подпрограммы печати значения call iprintLF ; из ‘edi’ в виде символов mov eax,rem ; вызов подпрограммы печати call sprint ; сообщения ‘Остаток от деления:’ mov eax,edx ; вызов подпрограммы печати значения call iprintLF ; из ‘edx’ (остаток) в виде символов call quit ; вызов подпрограммы завершения

Листинг 6.4. Программа вычисления вычисления варианта задания по номеру студенческого билета ;——————————– ; Программа вычисления варианта ;——————————– %include ‘in\_out.asm’ SECTION .data msg: DB ‘Введите No студенческого билета:’,0 rem: DB ‘Ваш вариант:’,0 SECTION .bss x: RESB 80 SECTION .text GLOBAL \_start \_start: mov eax, msg call sprintLF mov ecx, x mov edx, 80 call sread mov eax,x ; вызов подпрограммы преобразования call atoi ; ASCII кода в число, eax=x xor edx,edx mov ebx,20 div ebx inc edx mov eax,rem call sprint mov eax,edx call iprintLF call quit

# Выводы

Я освоила арифметические инструкции языка ассемблера NASM

# Список литературы