Отчёт по лабораторной работе №7

Дисциплина:архитектура компьютера

Лобанов Владислав Олегович

Содержание

# Цель работы

Изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

# Задание

1. Реализация переходов в NASM
2. Изучение структуры файла листинга
3. Задание для самостоятельной работы

# Теоретическое введение

Для реализации ветвлений в ассемблере используются так называемые команды передачи управления или команды перехода. Можно выделить 2 типа переходов:

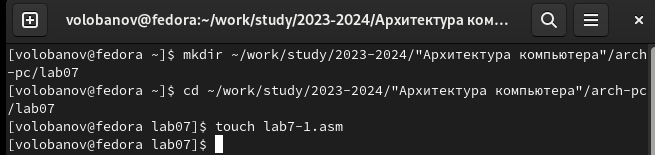
• Условный переход – выполнение или не выполнение перехода в определенную точку программы в зависимости от проверки условия.

• Безусловный переход – выполнение передачи управления в определенную точку про- граммы без каких-либо условий.

# Выполнение лабораторной работы

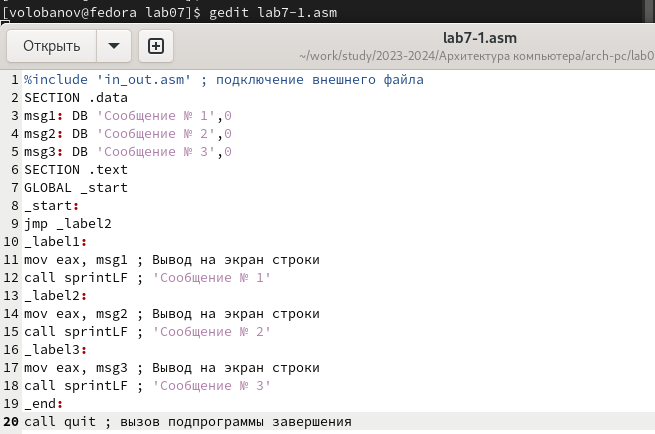
## Реализация переходов в NASM

Создаю каталог для программ для лабораторной работе №7, перехожу в него и создаю файл lab7-1.asm (рис. @fig:001).



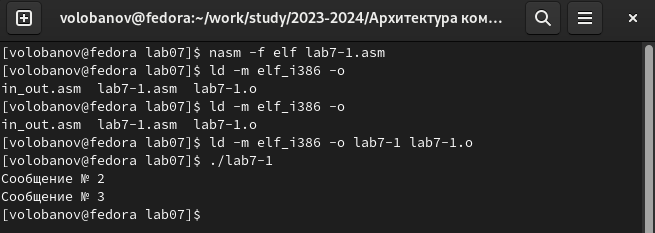
Создание каталога и файла в ней

Ввожу в файл lab7-1.asm текст программы с использованием функции jmp (рис. @fig:002).



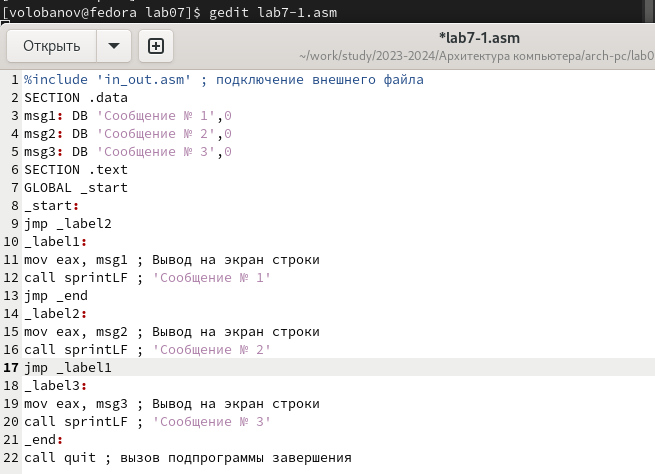
Редактирование файла

Создаю исполняемый файл и запускаю его (рис. @fig:003).



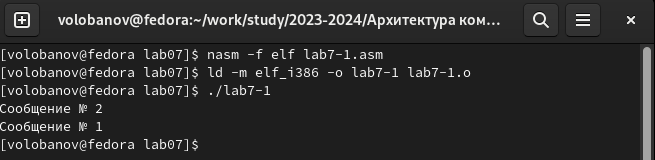
Исполнение программы

Изменяю программу таким образом, чтобы она выводила сначала ‘Сообщение No 2’, потом ‘Сообщение No 1’ и завершала работу (рис. @fig:004).



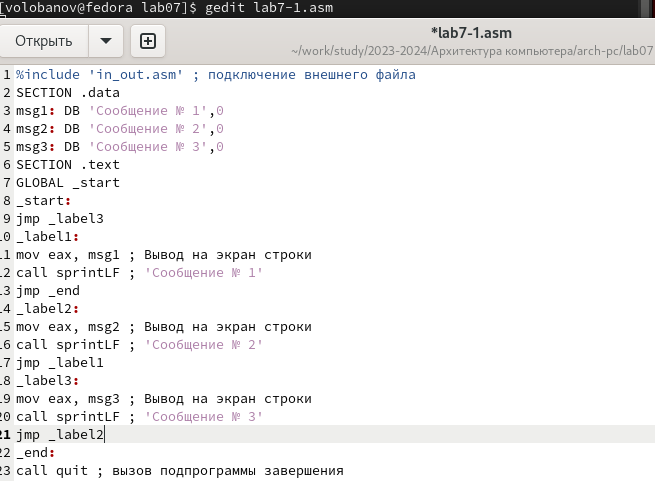
Редактирование файла

Создаю исполняемый файл и проверяю его работу (рис. @fig:005).



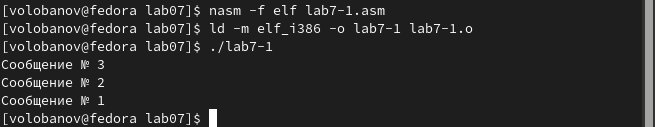
Исполнение программы

Изменяю программу таким образом, чтобы она выводила сначала ‘Сообщение No 3’, потом ‘Сообщение No 2’, потом ‘Сообщение No 1’ и завершала работу (рис. @fig:006).



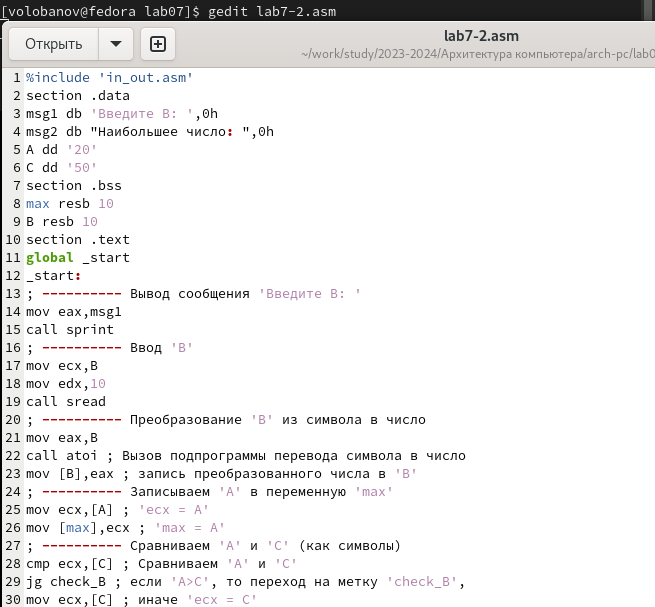
Редактирование файла

Создаю исполняемый файл и проверяю корректность работы программы (рис. @fig:007). Программа отработала корректно.



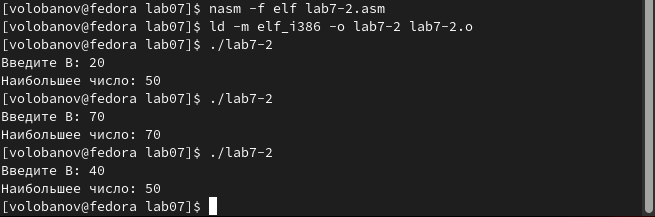
Исполнение программы

Создаю файл lab7-2.asm (рис. @fig:008).



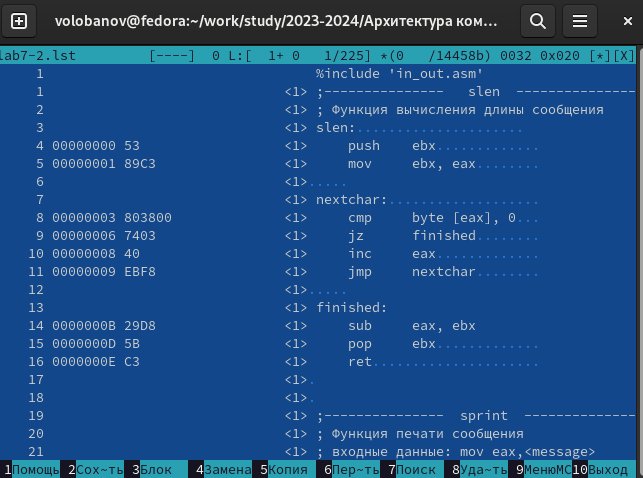
Создание файла

Ввожу в созданны файл текст программы, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных:A, B и C (рис. @fig:009).



Редактирование файла

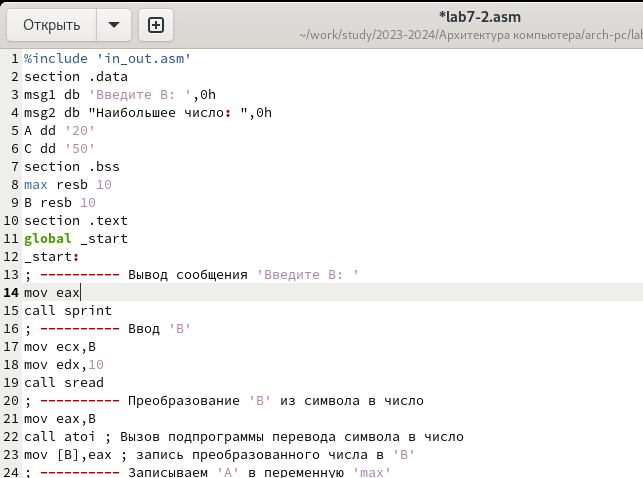
Создаю исполняемый файл и проверяю его работу для разных значений B (рис. @fig:010). Программа сработала корректно.



Исполнение программы для разных значений B

## Изучение структуры файла листинга

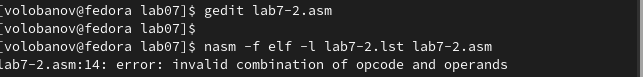
Создание файла листинга и его просмотр в текстовом редакторе gedit (рис. @fig:011).



Название рисунка

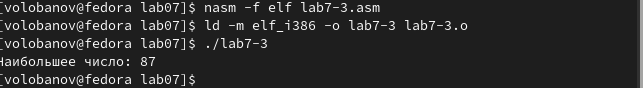
1. В строке 5 содержится собственно номер строки[5], адрес[00000001], машинный код[89C3] и содержимое строки кода[mov ebx, eax].
2. В строке 11 содержится собственно номер строки[11], адрес[00000009], машинный код[EBF8] и содержимое строки кода[jmp nextchar].
3. В строке 14 содержится собственно номер строки[14], адрес[0000000B], машинный код[29D8] и содержимое строки кода[sub eax, ebx].

Открываю файл lab7-2.asm и удаляю в инструкции mov вторгй операнд (рис. @fig:012).



Редактирование файла

Открытие файла листинга после трансляции (рис. @fig:013). Если в коде появляется ошибка, то её описание появится в файле листинга.



Открытие файла листинга

## Выполнение заданий для самостоятельной работы (вар. 20)

Создаю файл lab7-3.asm, пишу в нём программу для нахождения наименьшей из трёх целочисленных переменных a, b и c.

Текст программы в файле lab7-3.asm:

%include ‘in\_out.asm’ section .data msg2 db “Наибольшее число:”,0h A dd ‘54’ B dd ‘62’ C dd ‘87’

section .bss max resb 10

section .text

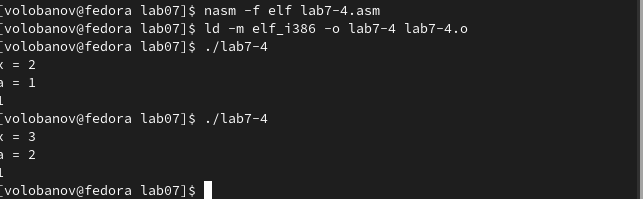
global \_start \_start: mov eax,B call atoi ; Вызов подпрограммы перевода символа в число mov [B],eax ; запись преобразованного числа в ‘B’ ; ———- Записываем ‘A’ в переменную ‘max’ mov ecx,[A] ; ‘ecx = A’ mov [max],ecx ; ‘max = A’ ; ———- Сравниваем ‘A’ и ‘С’ (как символы) cmp ecx,[C] ; Сравниваем ‘A’ и ‘С’ jg check\_B ; если ‘A>C’, то переход на метку ‘check\_B’, mov ecx,[C] ; иначе ‘ecx = C’ mov [max],ecx ; ‘max = C’

; ———- Преобразование ‘max(A,C)’ из символа в число check\_B: mov eax,max call atoi ; Вызов подпрограммы перевода символа в число mov [max],eax ; запись преобразованного числа в max ; ———- Сравниваем ‘max(A,C)’ и ‘B’ (как числа) mov ecx,[max] cmp ecx,[B] ; Сравниваем ‘max(A,C)’ и ‘B’ jg fin ; если ‘max(A,C)>B’, то переход на ‘fin’, mov ecx,[B] ; иначе ‘ecx = B’

mov [max],ecx

; ———- Вывод результата fin: mov eax, msg2 call sprint ; Вывод сообщения ‘Наибольшее число:’ mov eax,[max] call iprintLF ; Вывод ‘max(A,B,C)’ call quit ; Выход

Создаю исполняемый файл и проверяю его работу (рис. @fig:014). Программа отработала корректно.



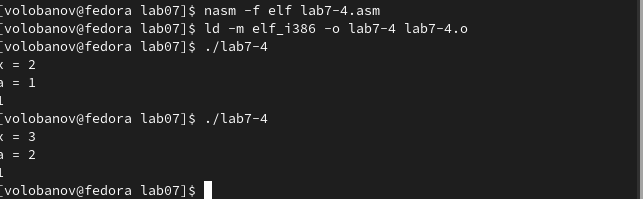
Исполнение программы

Создаю файл lab7-4.asm, пишу в нём програму, которая для введённых с клавиатуры значений x и a вычисляет значение функции f(x), которая равна x-a при x>=a, и 5, когда x < a и выводит результат вычислений.

Текст программы в файле lab7-4.asm:

%include 'in\_out.asm'  
  
section .data  
 msgX db "x = ",0h  
 msgA db "a = ",0h  
  
section .bss  
 x resb 10  
 a resb 10  
 f resb 10  
  
section .text  
global \_start  
  
\_start:  
 ; ---------- Ввод 'X'  
 mov eax, msgX  
 call sprint  
 mov ecx, x  
 mov edx,10  
 call sread  
  
 ; ---------- Ввод 'A'  
 mov eax, msgA  
 call sprint  
 mov ecx, a  
 mov edx,10  
 call sread  
  
 ; ---------- Преобразование 'x' из символа в число  
 mov eax, x  
 call atoi  
 mov [x], eax  
  
 ; ---------- Преобразование 'a' из символа в число  
 mov eax, a  
 call atoi  
 mov [a], eax  
  
  
 mov ecx, [x]  
 cmp ecx, [a]  
  
 jge func  
  
 mov eax, 5  
 jmp fin  
  
func:  
 mov eax, [x]  
 mov ecx, [a]  
 sub eax, ecx  
  
fin:  
 call iprintLF   
 call quit

Создаю исполняемый файл и проверяю его работу для пар x и a (1,2) и (2,1) (рис. @fig:015). Программа отработала верно.



Исполнение программы

# Выводы

В ходе выполнения лабораторной работы я освоил принципы условного и безусловного перехода в NASM.

# Список литературы

1. [Лабораторная работа №6](https://esystem.rudn.ru/pluginfile.php/2089545/mod_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%967.%20%D0%9A%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D1%8B%20%D0%B1%D0%B5%D0%B7%D1%83%D1%81%D0%BB%D0%BE%D0%B2%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%B8%20%D1%83%D1%81%D0%BB%D0%BE%D0%B2%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%BF%D0%B5%D1%80%D0%B5%D1%85%D0%BE%D0%B4%D0%BE%D0%B2%20%D0%B2%20Nasm.%20%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%B2%D0%B5%D1%82%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9..pdf)