Отчет по лабораторной работе №7

Дисциплина: архитектура компьютера

Гаязов Рузаль Ильшатович

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 файлов листинга
3. Самостоятельное написание программ по материалам лабораторной 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

В созданом каталоге для программам лабораторной работы № 7, перехожу в него и создаю файл lab7-1.asm. После чего ввожу в файл lab7-1.asm текст программы из листинга 7.1. (рис. -@fig:001).



Рис.1

Создаю исполняемый файл и запускаю его.(рис. -@fig:002).

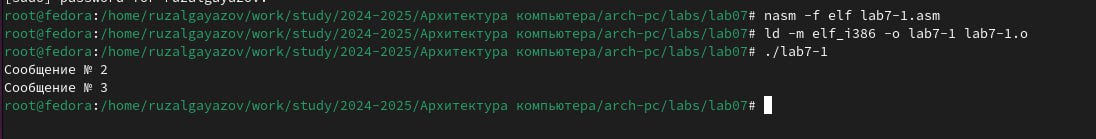


Рис.2

Изменим программу таким образом, чтобы она выводила сначала ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу. Для этого в текст программы после вывода сообщения № 2 добавим инструкцию jmp с меткой \_label1 (т.е. переход к инструкциям вывода сообщения № 1) и после вывода сообщения № 1 добавим инструкцию jmp с меткой \_end (т.е. переход к инструкции call quit). Изменим текст программы в соответствии с листингом 7.2. (рис. -@fig:003).



Рис.3

Создаю исполняемый файл и запускаю его. (рис. -@fig:004).

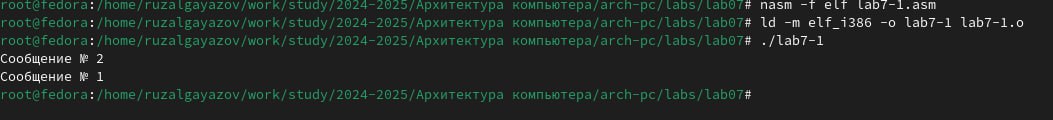


Рис.4

Изменяю текст программы, чтобы вывод программы был следующим: user@dk4n31:~$ ./lab7-1 Сообщение № 3 Сообщение № 2 Сообщение № 1 user@dk4n31:~$ (рис. -@fig:005).

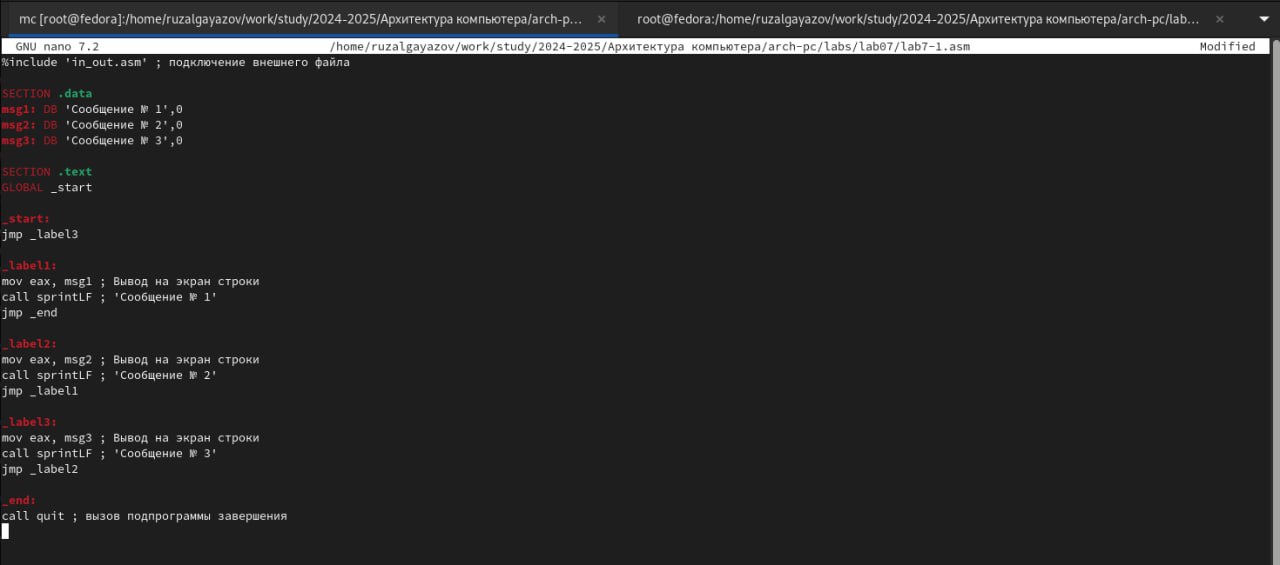


Рис.5

Создаю исполняемый файл и запускаю его. (рис. -@fig:006).

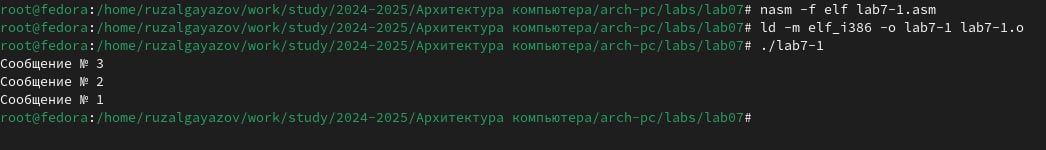


Рис.6

Внимательно изучаю текст программы из листинга 7.3 и ввожу в lab7-2.asm. (рис. -@fig:007).

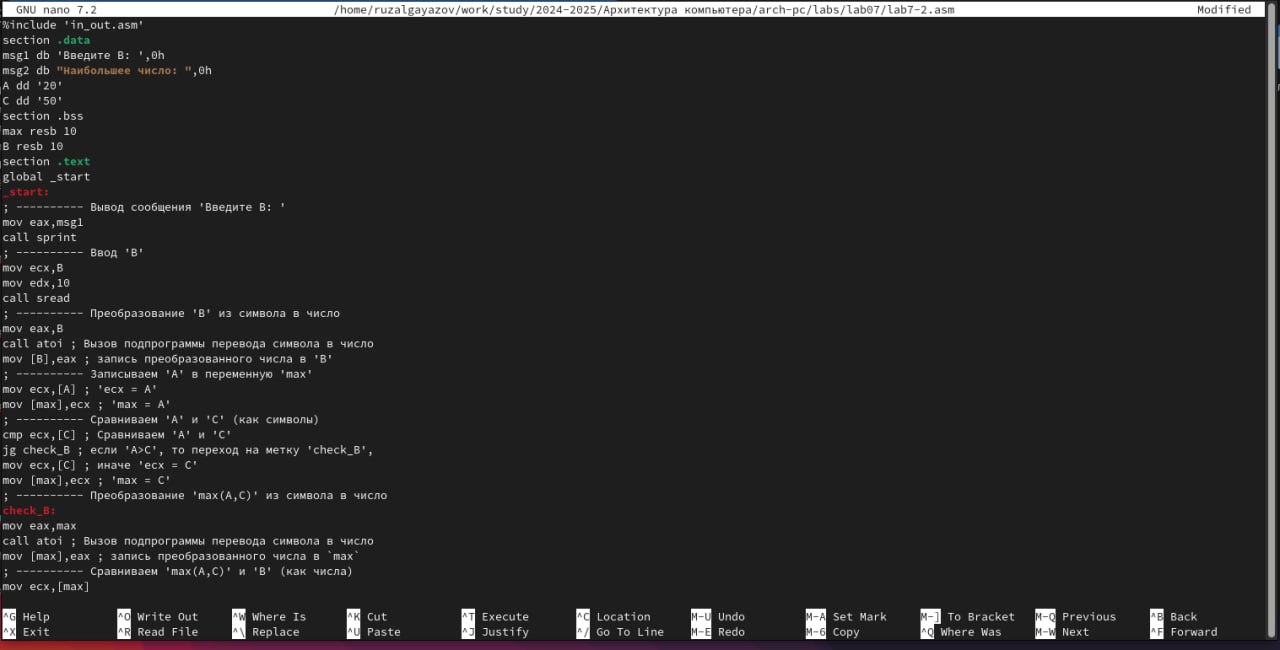


Рис.7

Создаю исполняемый файл и запускаю его. (рис. -@fig:008).

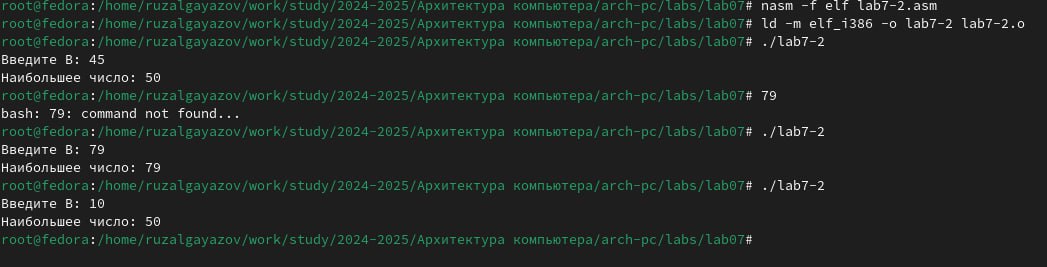


Рис.8

Создаю файл листинга для программы из файла lab7-2.asm и открываю его.(рис. -@fig:009).

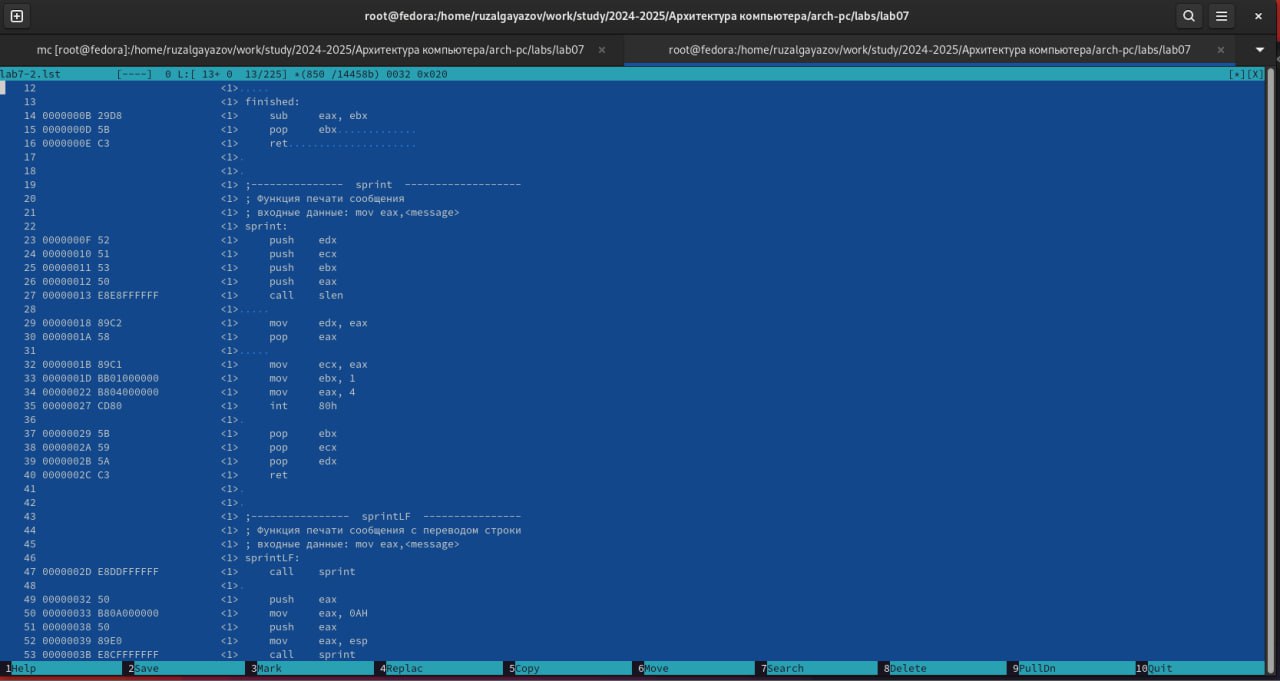


Рис.9

Первое значение в файле листинга - номер строки, и он может вовсе не совпадать с номером строки изначального файла. Второе вхождение - адрес, смещение машинного кода относительно начала текущего сегмента, затем непосредственно идет сам машинный код, а заключает строку исходный текст прогарммы с комментариями.

Удаляю один операнд из случайной инструкции, чтобы проверить поведение файла листинга в дальнейшем. В новом файле листинга должно показывать ошибку, которая возникла при попытке трансляции файла. Никакие выходные файлы при этом помимо файла листинга не создаются.(в моем случае ошибка в терминале) (рис. -@fig:010).

Рис.10

Рис.10

## Задания для самостоятельной работы

Программа к заданию 1. (рис. -@fig:011)



Рис.11

Код:

%include ‘in\_out.asm’

SECTION .data msg\_input\_x: DB ‘Введите значение x:’, 0 msg\_input\_a: DB ‘Введите значение a:’, 0 msg\_result: DB ‘Результат вычислений:’, 0

SECTION .bss x: RESD 1 a: RESD 1 result: RESD 1

SECTION .text GLOBAL \_start

\_start: ; Ввод x mov eax, msg\_input\_x call sprint

mov ecx, x  
mov edx, 10  
call sread  
  
mov eax, x  
call atoi  
mov [x], eax  
  
; Ввод a  
mov eax, msg\_input\_a  
call sprint  
  
mov ecx, a  
mov edx, 10  
call sread  
  
mov eax, a  
call atoi  
mov [a], eax  
  
; Вычисление f(x)  
mov eax, [x]  
mov ebx, [a]  
cmp eax, ebx  
jle .less\_or\_equal  
  
; f(x) = 2 \* (x - a)  
sub eax, ebx  
shl eax, 1  
mov [result], eax  
jmp .print\_result

.less\_or\_equal: ; f(x) = 15 mov eax, 15 mov [result], eax

.print\_result: mov eax, msg\_result call sprint

mov eax, [result]  
call iprintLF  
call quit

Создаю исполняемый файл и запускаю его. (рис. -@fig:012)

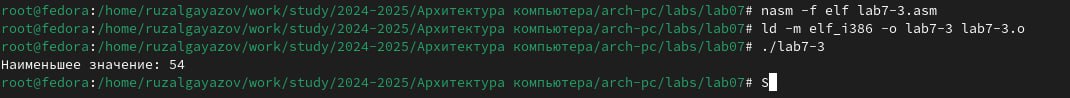


Рис.12

Программа к заданию 2. (рис. -@fig:013).

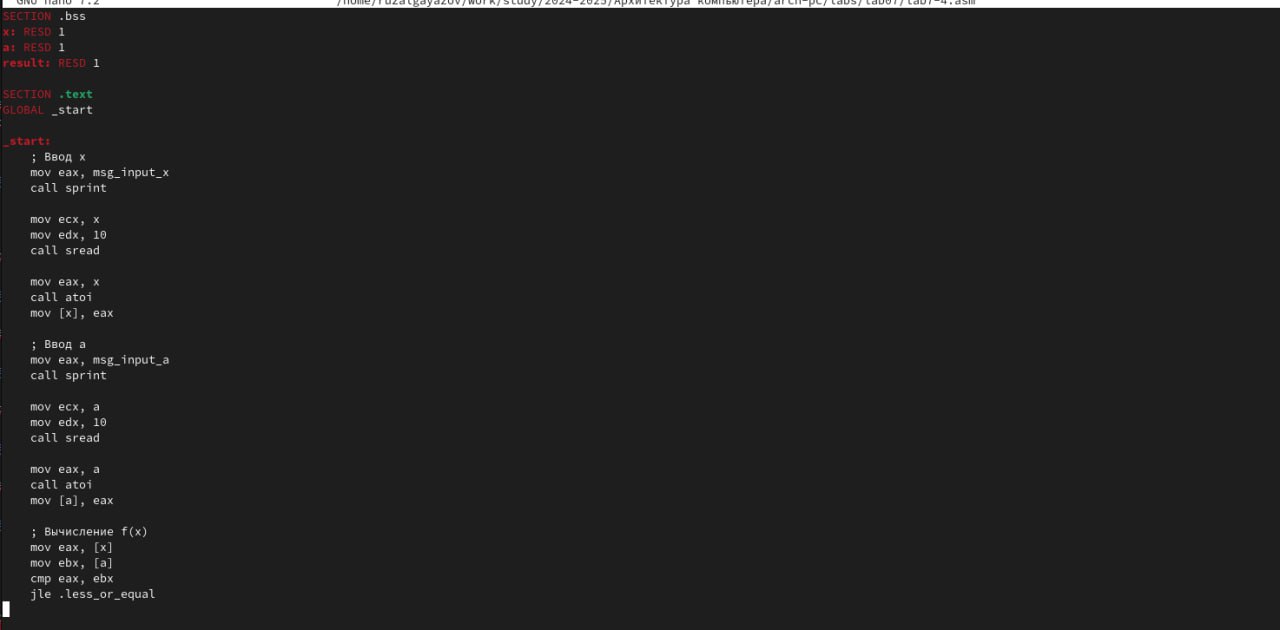


Рис.13

Код:

%include ‘in\_out.asm’

SECTION .data msg\_input\_x: DB ‘Введите значение x:’, 0 msg\_input\_a: DB ‘Введите значение a:’, 0 msg\_result: DB ‘Результат вычислений:’, 0

SECTION .bss x: RESD 1 a: RESD 1 result: RESD 1

SECTION .text GLOBAL \_start

\_start: ; Ввод x mov eax, msg\_input\_x call sprint

mov ecx, x  
mov edx, 10  
call sread  
  
mov eax, x  
call atoi  
mov [x], eax  
  
; Ввод a  
mov eax, msg\_input\_a  
call sprint  
  
mov ecx, a  
mov edx, 10  
call sread  
  
mov eax, a  
call atoi  
mov [a], eax  
  
; Вычисление f(x)  
mov eax, [x]  
mov ebx, [a]  
cmp eax, ebx  
jle .less\_or\_equal  
  
; f(x) = 2 \* (x - a)  
sub eax, ebx  
shl eax, 1  
mov [result], eax  
jmp .print\_result

.less\_or\_equal: ; f(x) = 15 mov eax, 15 mov [result], eax

.print\_result: mov eax, msg\_result call sprint

mov eax, [result]  
call iprintLF  
call quit

Создаю исполняемый файл и запускаю его. (рис. -@fig:014).

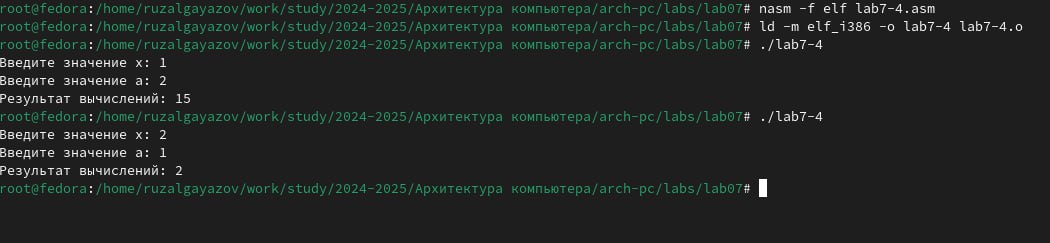


Рис.14

# Выводы

При выполнении лабораторной работы я изучил команды условных и безусловных переходво, а также приобрел навыки написания программ с использованием перходов, познакомился с назначением и структурой файлов листинга.

# Список литературы

1. [Курс на ТУИС](https://esystem.rudn.ru/course/view.php?id=112)
2. [Лабораторная работа №7](https://esystem.rudn.ru/pluginfile.php/2089087/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)
3. [Программирование на языке ассемблера NASM Столяров А. В.](https://esystem.rudn.ru/pluginfile.php/2088953/mod_resource/content/2/%D0%A1%D1%82%D0%BE%D0%BB%D1%8F%D1%80%D0%BE%D0%B2%20%D0%90.%20%D0%92.%20-%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%BD%D0%B0%20%D1%8F%D0%B7%D1%8B%D0%BA%D0%B5%20%D0%B0%D1%81%D1%81%D0%B5%D0%BC%D0%B1%D0%BB%D0%B5%D1%80%D0%B0%20NASM%20%D0%B4%D0%BB%D1%8F%20%D0%9E%D0%A1%20Unix.pdf)