Отчёт по лабораторной работе №7

Дисциплина: Архитектура компьютера

Калашникова Ольга Сергеевна

Содержание

# 1 Цель работы

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

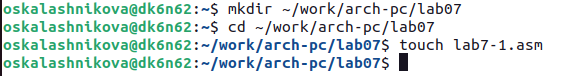
# 2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

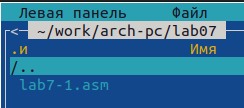
# 3 Выполнение лабораторной работы

## 3.1 Реализация переходов в NASM

Создаём каталог для программам лабораторной работы № 7 (при помощи команды mkdir ~/work/arch-pc/lab07), переходим в него (при помощи команды cd ~/work/arch-pc/lab07) и создаём файл lab7-1.asm (при помощи команды touch lab7-1.asm) (рис. ??),(рис. ??)

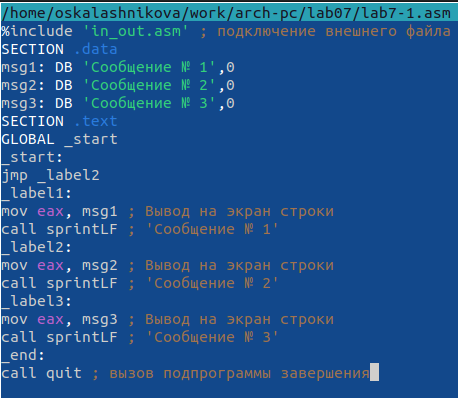


Создание папки, её открытие и создание файла

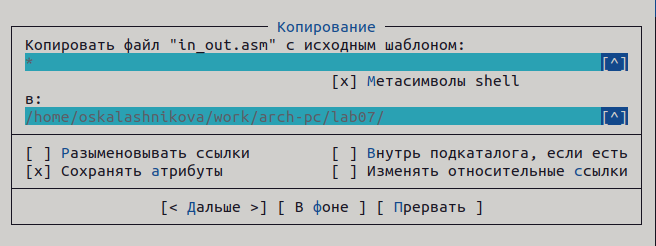


Проверка наличия папки

Введём в файл lab7-1.asm текст программы из листинга 6.1., для корректной работы нужно переместить файл “in\_out.asm” в тот же каталог, где лежит и файл с текстом программы.(рис. ??),(рис. ??)

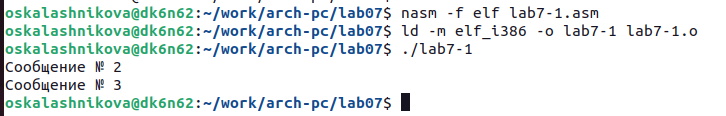


Текст программы



Перемещение файла “in\_out.asm”

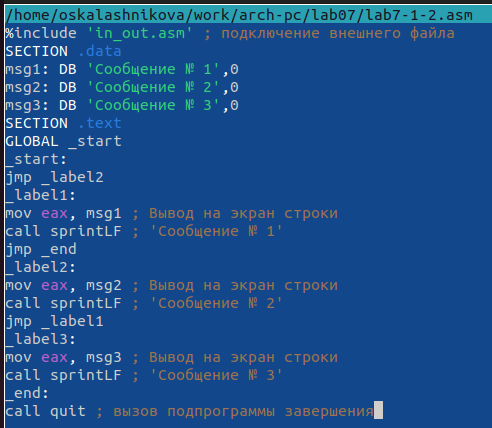
Создаем исполняемый файл и запускаем его (компиляция: nasm -f elf lab7-1.asm , ld -m elf\_i386 -o lab7-1 lab7-1.o , запуск: ./lab7-1) (рис. ??)



Создание и запуск исполняемого файла

Мы видим, что программа вывела лишь 2 строки, потому что использование инструкции jmp \_label2 меняет порядок исполнения инструкций и позволяет выполнить инструкции начиная с метки \_label2, пропустив вывод первого сообщения.

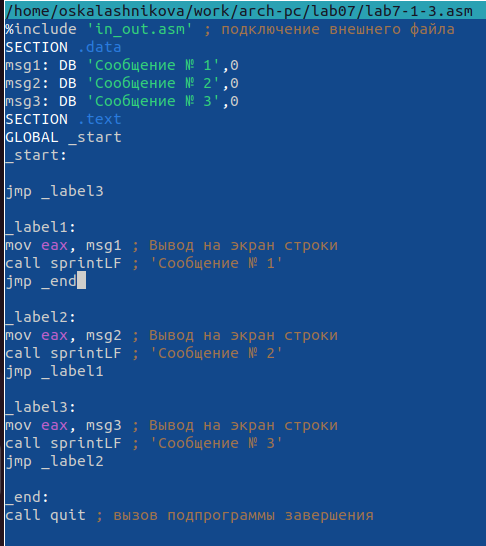
Меняем текст программы (чтобы она выводила сначала ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу) (рис. ??)



Изменённый текст программы

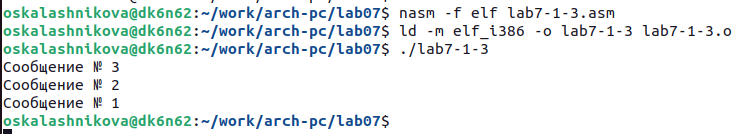
Создаём исполняемый файл и запускаем его (компиляция: nasm -f elf lab7-1-2.asm , ld -m elf\_i386 -o lab7-1-2 lab7-1-2.o , запуск: ./lab7-1-2) (рис. ??)

Меняем текст программы (чтобы она выводила сначала ‘Сообщение № 3’, потом ‘Сообщение № 2’, затем ‘Сообщение № 1’ и завершала работу) (рис. ??)



Изменённый текст программы

Создаём исполняемый файл и запускаем его (компиляция: nasm -f elf lab7-1-3.asm , ld -m elf\_i386 -o lab7-1-3 lab7-1-3.o , запуск: ./lab7-1-3) (рис. ??)



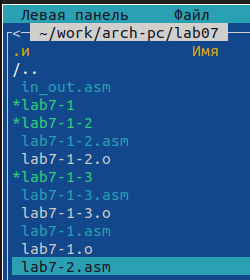
Создание и запуск исполняемого файла

Для этого в начале используем инструкцию jmp \_label3, меняя порядок исполнения инструкций и позволяя выполнить инструкции начиная с метки \_label3. В текст программы после вывода сообщения № 3 добавим инструкцию jmp с меткой \_label2 (т.е. переход к инструкциям вывода сообщения № 2). После вывода сообщения № 2 добавим инструкцию jmp с меткой \_label1 (т.е. переход к инструкциям вывода сообщения № 1). После вывода сообщения № 1 добавим инструкцию jmp с меткой \_end (т.е. переход к инструкции call quit).

Создаём файл “lab7-2.asm” в каталоге ~/work/arch-pc/lab07 (touch ~/work/arch-pc/lab06/lab7-2.asm) (рис. ??), (рис. ??)

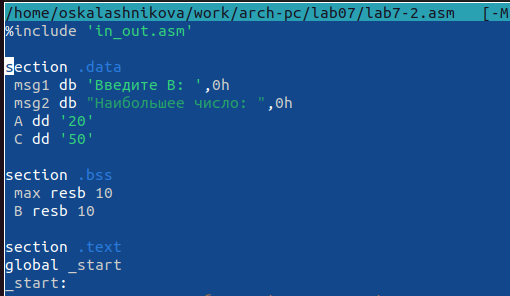
Создание файла

Создание файла

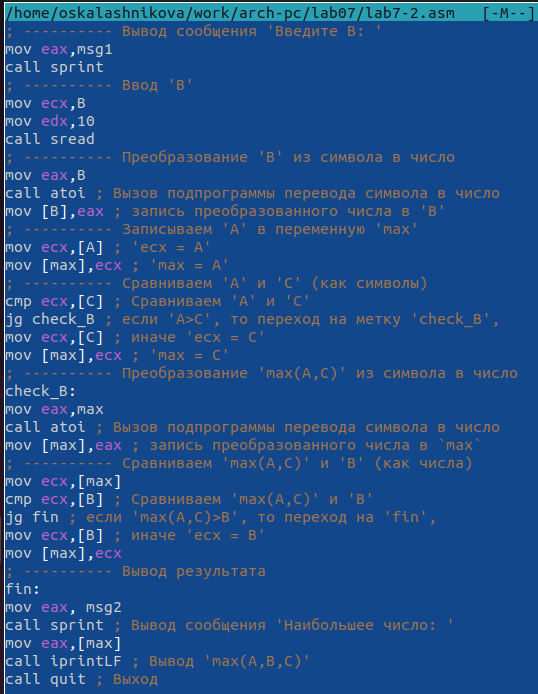


Проверка наличия файла

Вводим в файл текст другой программы, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: A,B и C (Значения для A и C задаются в программе, значение B вводиться с клавиатуры) (рис. ??), (рис. ??)

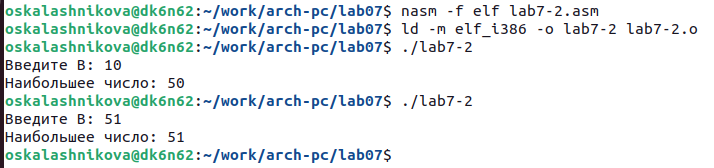


Текст программы 1



Текст программы 1

Создаём исполняемый файл и запускаем его (компиляция: nasm -f elf lab7-2.asm , ld -m elf\_i386 -o lab7-2 lab7-2.o , запуск: ./lab7-2) (рис. ??)



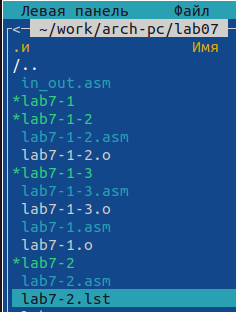
Создание и запуск исполняемого файла

## 3.2 Изучение структуры файлы листинга

Создадим файл листинга для программы из файла lab7-2.asm (nasm -f elf -l lab7-2.lst lab7-2.asm) (рис. ??),(рис. ??)

Создание файла листинга

Создание файла листинга

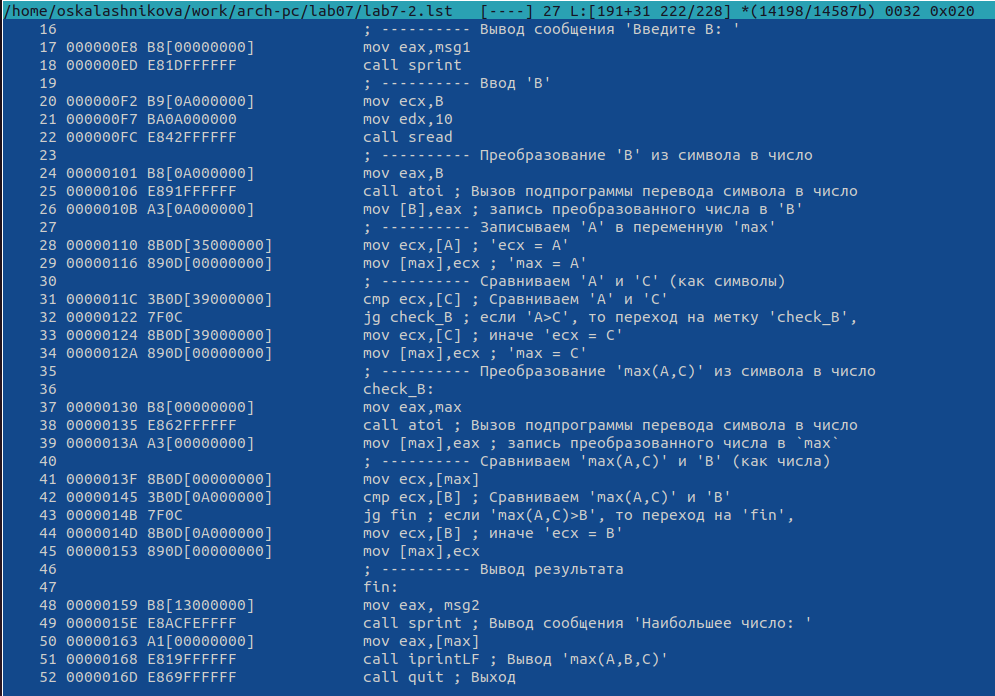


Проверка наличия файла

Откроем файл листинга lab7-2.lst с помощью текстового редактора mcedit (mcedit lab7-2.lst) (рис. ??), (рис. ??)

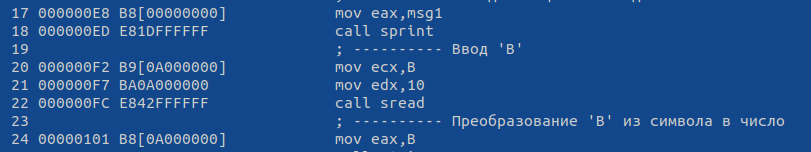
Открытие файла листинга 1

Открытие файла листинга 1



Открытие файла листинга 2

Опишем содержание строк 17, 21, 24 (рис. ??)



Строки листинга, которые будут описаны ниже

17 000000E8 B8[00000000] mov eax,msg1

* 17: Это номер строки в исходном коде программы.
* 000000E8: Это адрес, по которому располагается данная инструкция в сегменте кода.
* B8[00000000]: Это машинный код операции mov для записи адреса msg1 в регистр eax.
* mov eax, msg1: Это исходное описание операции - она копирует значение, хранящееся по адресу msg1, в регистр eax.

21 000000F7 BA0A000000 mov edx,10

* 21: Номер строки.
* 000000F7: Адрес, где находится инструкция mov в сегменте кода.
* BA0A000000: Машинный код для операции mov, записывающей значение 10 в регистр edx.
* mov edx, 10: Исходный код программы, в котором мы помещаем значение 10 в регистр edx.

24 00000101 B8[0A000000] mov eax,B

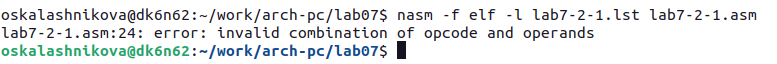
* 24: Номер строки.
* 00000101: Адрес, где находится инструкция mov в сегменте кода.
* B8[0A000000]: Машинный код для операции mov, записывающей значение, хранящееся по адресу 0A000000 в регистр eax.
* mov eax, B: Исходный код программы, в котором мы помещаем значение, хранящееся по адресу 0A000000, в регистр eax.

Откроем файл с программой lab7-2.asm и в любой инструкции с двумя операндами удалим один операнд. (рис. ??)

Изменение кода

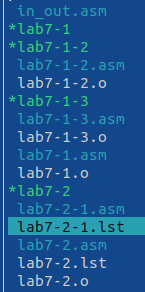
Изменение кода

Выполним трансляцию с получением файла листинга(nasm -f elf -l lab7-2.lst lab7-2.asm) (рис. ??)



Трансляция с получением файла листинга

Мы видим ошибку и создаётся только файл листинга (рис. ??)



Созданный файл

Зайдя в листинг, мы видим что в нём создаётся строка, предупреждающая об ошибке (рис. ??)

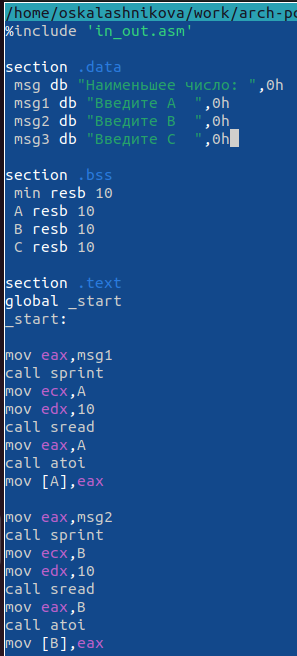
Файл листинга

Файл листинга

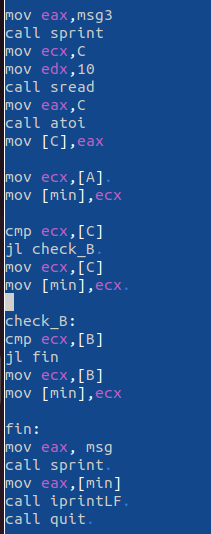
## 3.3 Задание для самостоятельной работы

Задание 1: Напишите программу нахождения наименьшей из 3 целочисленных переменных a,b,c. Значения переменных выбрать из табл. 7.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу.

Вносим изменения в программу Листинга 7.3 для нахождения минимума из трёх введённых с клавиатуры переменных. Используем jl (Переход если a меньше b). (рис. ??), (рис. ??)

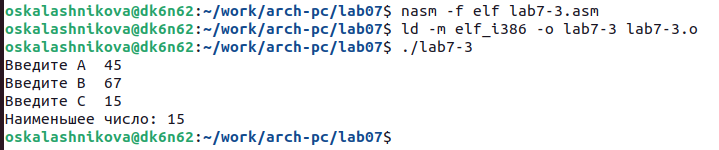


Изменение кода 1



Изменение кода 2

Создаём исполняемый файл и запускаем его (компиляция: nasm -f elf lab7-3.asm , ld -m elf\_i386 -o lab7-3 lab7-3.o , запуск: ./lab7-3), вводим значения из своего варианта (рис. ??), (рис. ??)



Создание и запуск исполняемого файла

Значения в варианте

Значения в варианте

Текст программы:

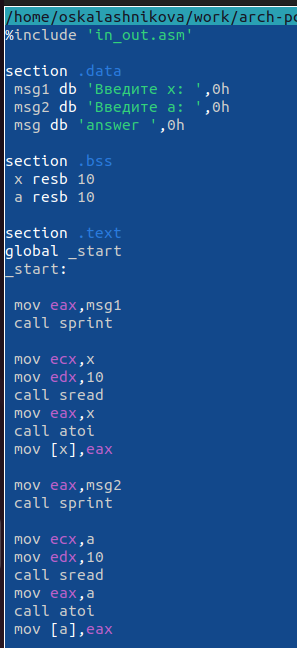
%include 'in\_out.asm'  
  
section .data  
 msg db "Наименьшее число: ",0h  
 msg1 db "Введите A ",0h  
 msg2 db "Введите B ",0h  
 msg3 db "Введите C ",0h  
  
section .bss  
 min resb 10  
 A resb 10  
 B resb 10  
 C resb 10  
  
section .text  
global \_start  
\_start:  
  
mov eax,msg1  
call sprint  
mov ecx,A  
mov edx,10  
call sread  
mov eax,A  
call atoi  
mov [A],eax  
  
mov eax,msg2  
call sprint  
mov ecx,B  
mov edx,10  
call sread  
mov eax,B  
call atoi  
mov [B],eax  
  
mov eax,msg3  
call sprint  
mov ecx,C  
mov edx,10  
call sread  
mov eax,C  
call atoi  
mov [C],eax  
  
mov ecx,[A]   
mov [min],ecx  
  
cmp ecx,[C]  
jl check\_B   
mov ecx,[C]  
mov [min],ecx   
  
check\_B:  
cmp ecx,[B]  
jl fin  
mov ecx,[B]  
mov [min],ecx  
  
fin:  
mov eax, msg  
call sprint   
mov eax,[min]  
call iprintLF   
call quit

Задание 2: Напишите программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции f(x) и выводит результат вычислений. Вид функции f(x) выбрать из таблицы 7.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу для значений x и a из 7.6.

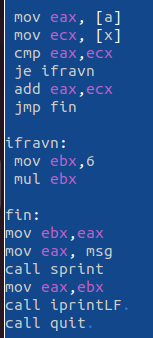
Пишем программу, которая вычисляет значение функции: 6а если x=а и а+х если x!=a (рис. ??), (рис. ??), (рис. ??)

Вариант 7

Вариант 7

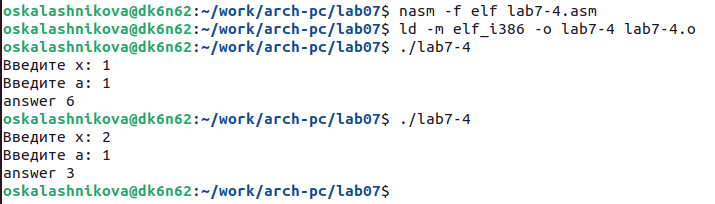


Текст программы 1



Текст программы 2

Создаём исполняемый файл и запускаем его (компиляция: nasm -f elf lab7-4.asm , ld -m elf\_i386 -o lab7-4 lab7-4.o , запуск: ./lab7-4), вводим значения из своего варианта (рис. ??)



Создание и запуск исполняемого файла

%include 'in\_out.asm'  
  
section .data  
 msg1 db 'Введите x: ',0h  
 msg2 db 'Введите а: ',0h  
 msg db 'answer ',0h  
  
section .bss  
 x resb 10  
 a resb 10  
  
section .text  
global \_start  
\_start:  
  
 mov eax,msg1  
 call sprint  
  
 mov ecx,x  
 mov edx,10  
 call sread  
 mov eax,x  
 call atoi  
 mov [x],eax  
  
 mov eax,msg2  
 call sprint  
  
 mov ecx,a  
 mov edx,10  
 call sread  
 mov eax,a  
 call atoi  
 mov [a],eax  
  
 mov eax, [a]  
 mov ecx, [x]  
 cmp eax,ecx  
 je ifravn  
 add eax,ecx  
 jmp fin  
  
ifravn:  
 mov ebx,6  
 mul ebx  
  
fin:  
mov ebx,eax  
mov eax, msg  
call sprint  
mov eax,ebx  
call iprintLF   
call quit

# 4 Выводы

В ходе данной лабораторной работы мы изучили команды условного и безусловного перехода, приобрели навыков написания программ с использованием переходов, познакомились с назначением и структурой файла листинга.