Отчёта по лабораторной работе №7

Команды безусловного и условного переходов в Nasm. Программирование ветвлений

Кичигина Полина Евгеньевна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Освоить условного и безусловного перехода. Ознакомиться с назначением и структурой файла листинга.

2 Задание

Написать программы для решения системы выражений.

3 Выполнение лабораторной работы

1. Создайте каталог для программам лабораторной работы № 7, перейдите в него и со- здайте файл lab7-1.asm(рис. 3.1)

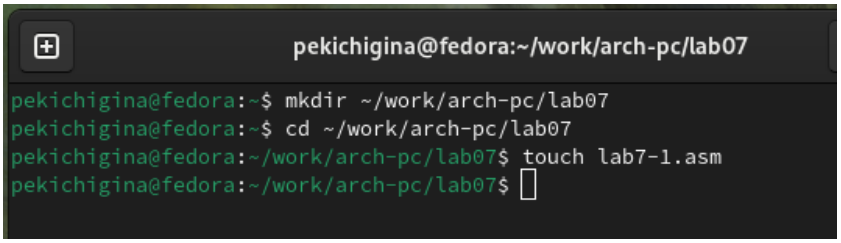

Рис. 3.1: Создаем каталог с помощью команды mkdir и файл с помощью команды touch

2. Инструкция jmp в NASM используется для реализации безусловных пере- ходов. Рассмот- рим пример программы с использованием инструкции jmp. Введите в файл lab7-1.asm текст программы из листинга 7.1(рис. 3.2)

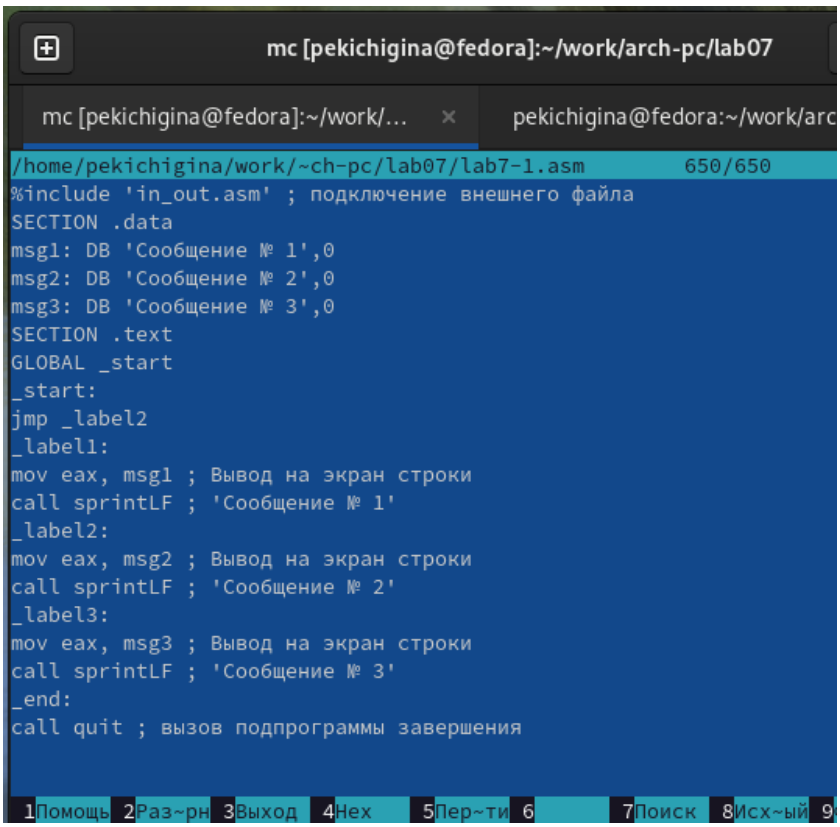

Рис. 3.2: Заполняем файл

Создайте исполняемый файл и запустите его(рис. 3.3)

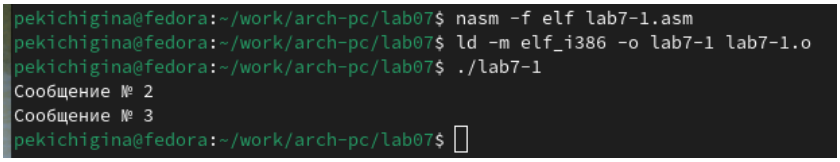

Рис. 3.3: Запускаем файл и смотрим на его работу

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). Измените текст программы в соответствии с листингом 7.2(рис. 3.4)

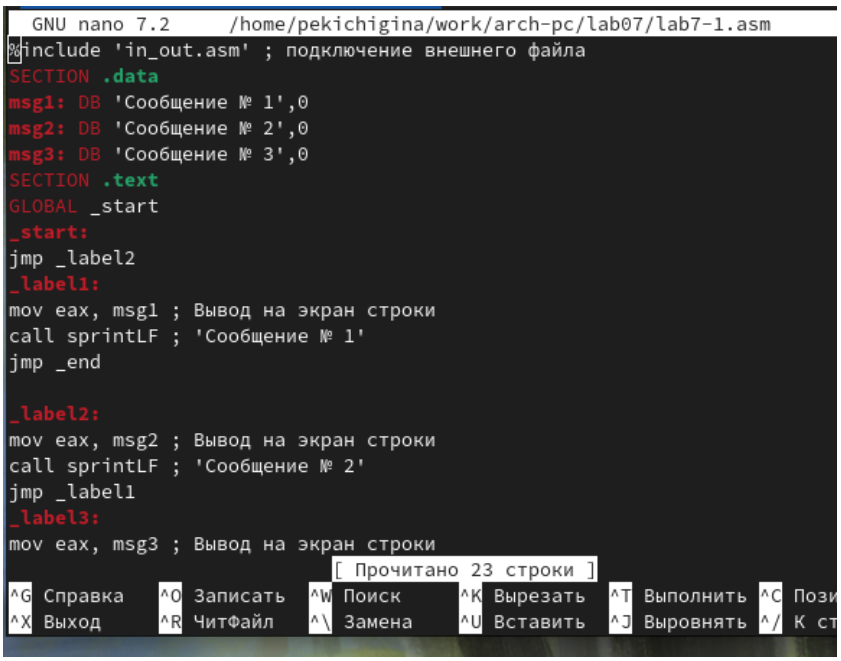

Рис. 3.4: Изменяем файл

Создайте исполняемый файл и проверьте его работу(рис. 3.5)

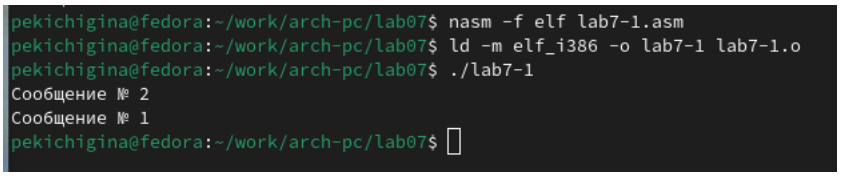

Рис. 3.5: Запускаем файл и смотрим на его работу

Измените текст программы добавив или изменив инструкции jmp, чтобы вывод програм- мы был следующим:

Сообщение № 3 Сообщение № 2 Сообщение № 1(рис. 3.6)

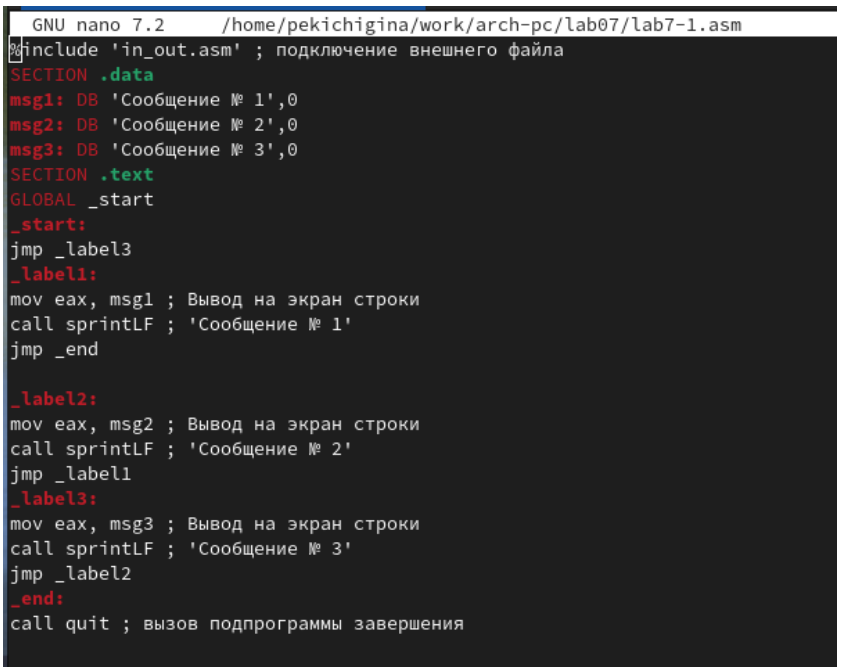

Рис. 3.6: Редактируем файл

Теперь проверяем работу программы(рис. 3.7)

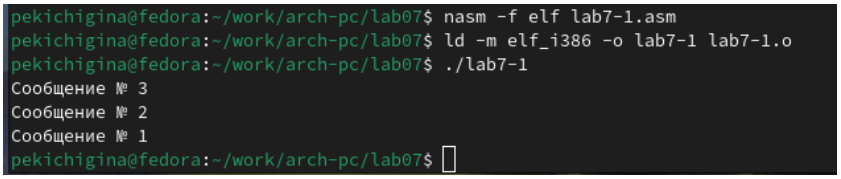

Рис. 3.7: Проверяем работу файла

3. Создайте файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. Внимательно изучите текст программы из листинга 7.3 и введите в lab7-2.asm(рис. 3.8)

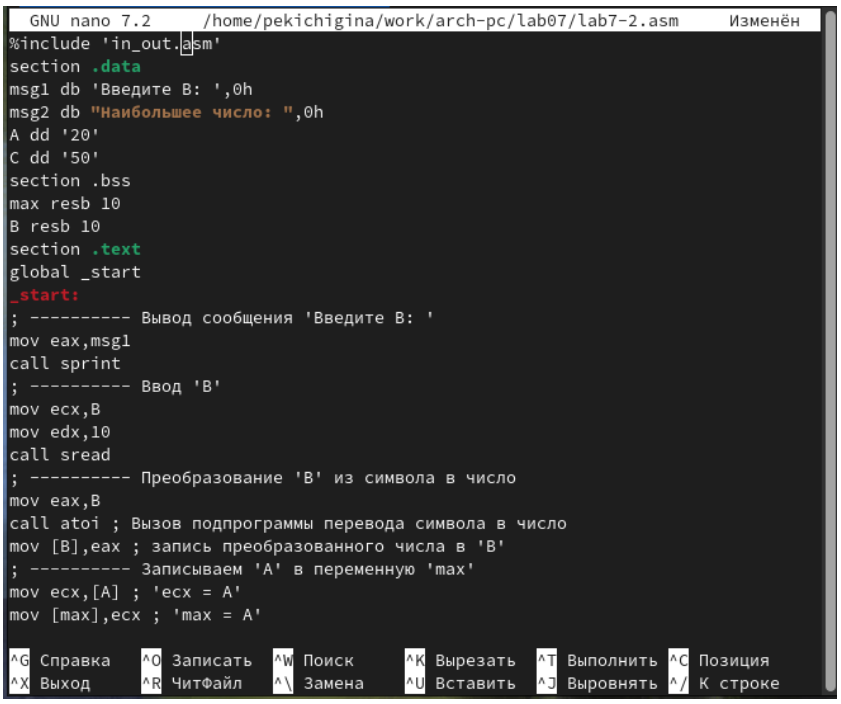

Рис. 3.8: Создаем и заполняем файл

Создайте исполняемый файл и проверьте его работу для разных значений B(рис. 3.9)

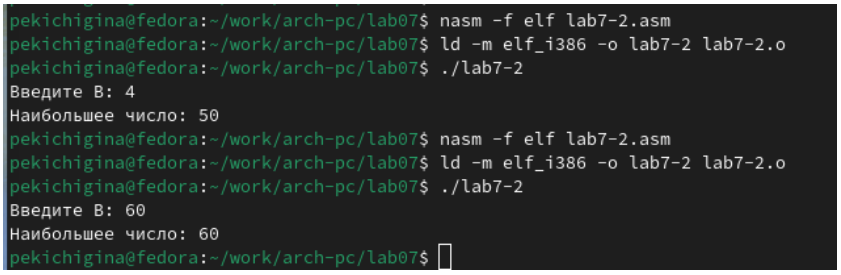

Рис. 3.9: Смотрим на работу программ

4. Создайте файл листинга для программы из файла lab7-2.asm(рис. 3.10)

Рис. 3.10: создаем файл

Откройте файл листинга lab7-2.lst с помощью любого текстового редактора,

например mcedit. Внимательно ознакомиться с его форматом и содержимым. Подробно объяснить содержи- мое трёх строк файла листинга по выбору.

Строка 33: 0000001D-адрес в сегменте кода, BB01000000-машинный код, mov ebx,1-присвоение переменной ecx значения 1.

Строка 34: 00000022-адрес в сегменте кода, B804000000-машинный код, mov eax,4-присвоение переменной eax значения 4.

Строка 35: 00000027-адрес в сегменте кода, CD80-машинный код, int 80h-вызов ядра.

Откройте файл с программой lab7-2.asm и в любой инструкции с двумя операндами удалить один операнд. Выполните трансляцию с получением файла листинга.

Какие выходные файлы создаются в этом случае? Что добавляется в листинге? При трансляции файла, выдается ошибка, но создаются исполнительный файл lab7-2 и lab7-2.lst(рис. 3.11)

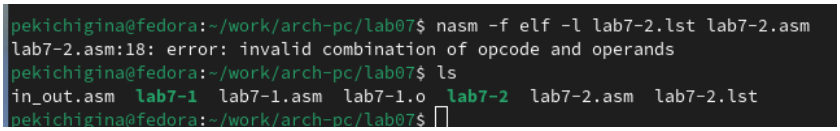

Рис. 3.11: Транслируем файл и наблюдаем его работу

4 Задания для самостоятельной работы

1. Напишите программу нахождения наименьшей из 3 целочисленных переменных a, b и c. Значения переменных выбрать из табл. 7.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 6(рис. 4.1)

```
 GNU nano 7.2     /home/pekichigina/work/arch-pc/lab07/lab7-3.asm
%include 'in_out.asm'
section        .data
   msg1 db 'Введите B: ',0h
   msg2 db "Наименьшее число: ",0h
   A dd '79'
   C dd '41'
section .bss
   min resb 10
   B   resb 10
section       .text
   global _start
_start:
   mov  eax,msg1
   call sprint
   mov ecx,B
   mov  edx,10
   call sread
   mov eax,B
   call atoi
   mov  [B],eax
   mov  ecx,[A]
   mov  [min],ecx
   cmp ecx,[C]
   jl check_B
   mov ecx,[C]
   mov [min],ecx
check_B:
   mov  eax,min
   call atoi
   mov  [min],eax
   mov  ecx,[min]
   cmp  ecx,[B]
   jl   fin
   mov  ecx,[B]
   mov  [min],ecx
fin:
   mov  eax, msg2
   call sprint
   mov  eax,[min]
   call iprintLF
   call quit
```

Рис. 4.1: Создаем и формируем программу

Создайте исполняемый файл и проверьте его работу(рис. 4.2)

```
pekichigina@fedora:~/work/arch-pc/lab07$ nasm -f elf lab7-3.asm
pekichigina@fedora:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-3 lab7-3.o
pekichigina@fedora:~/work/arch-pc/lab07$ ./lab7-3
Введите B: 83
Наименьшее число: 41
pekichigina@fedora:~/work/arch-pc/lab07$
```

Рис. 4.2: Смотрим на рабботу программы(всё верно)

2. Напишите программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции f(x) и выводит результат вы-

числений. Вид функции f(x) выбрать из таблицы 7.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 6(рис. 4.3)

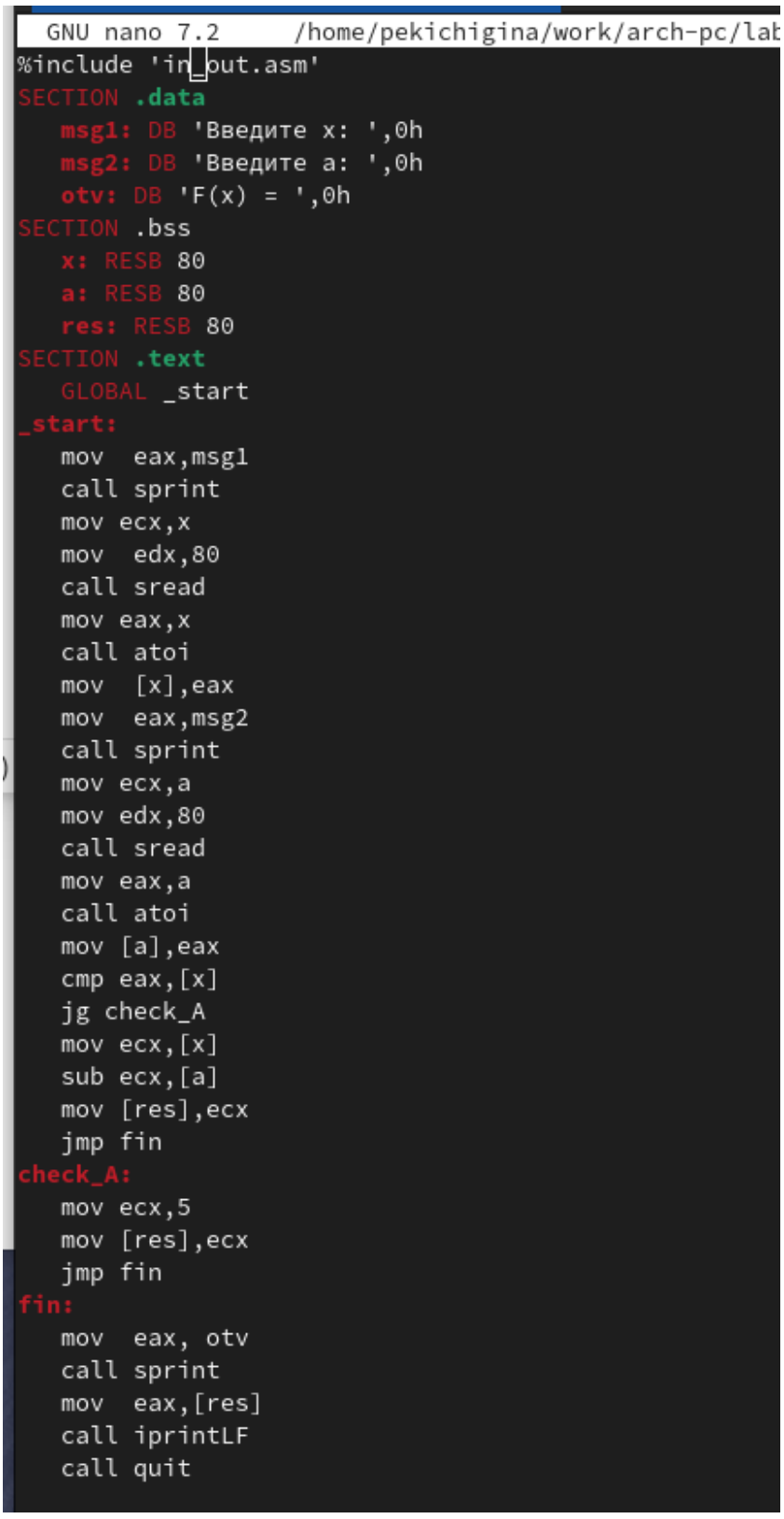

Рис. 4.3: Создаем файл и пишем программу

Создайте исполняемый файл и проверьте его работу для значений x и a из 7.6(рис. 4.4)

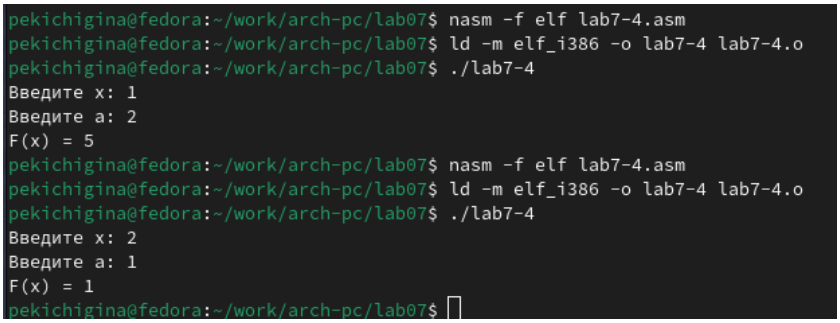

Рис. 4.4: Проверяем работу программы

5 Выводы

Мы познакомились с структурой файла листинга, изучили команды условного и безусловного перехода.