Отчет по выполнению лабораторной работы №7

Дисциплина: архитектура компьютеров

Намруев Максим Саналович

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

1. Реализация переходов в NASM
2. Изучение структуры файлы листинга

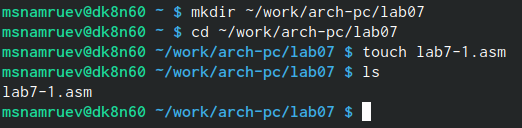
# 3 Теоретическое введение

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. Листинг (в рамках понятийного аппарата NASM) — это один из выходных файлов, созда- ваемых транслятором. Он имеет текстовый вид и нужен при отладке программы, так как кроме строк самой программы он содержит дополнительную информацию.

# 4 Выполнение лабораторной работы

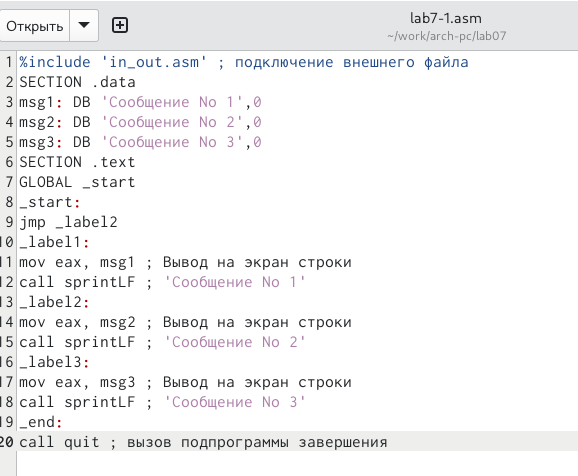
##Реализация переходов в NASM

Открываю терминал и создаю каталог для программ лабораторной работы, перехожу в него и создаю файл lab7-1.asm (рис. [??]).



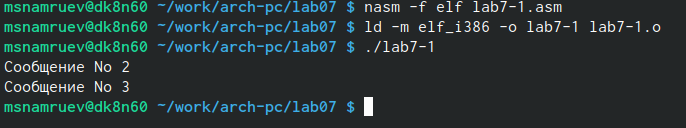
Создание файла

Ввожу в файл lsb7-1.asm текст программы из листинга 7.1.(рис. [??])



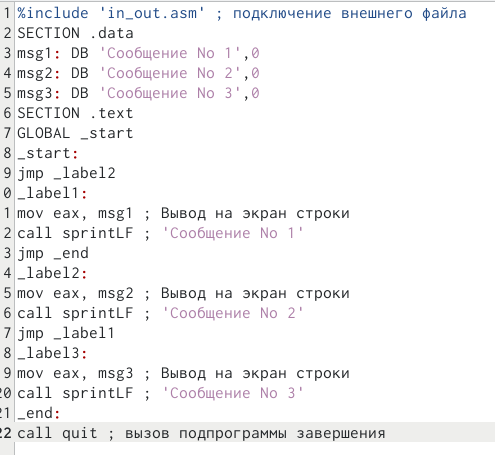
Ввод текста из листинга 7.1

Создаю исполняемый файл и запускаю его.(рис. [??])



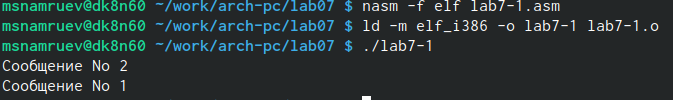
Запуск файла

Изменяю текст программы в соответствии с листингом 7.2.(рис. [??])



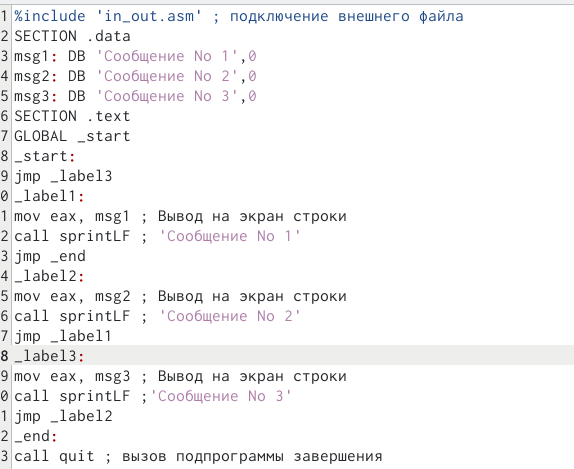
Изменение файла

Создаю исполняемый файл и запускаю его.(рис. [??])



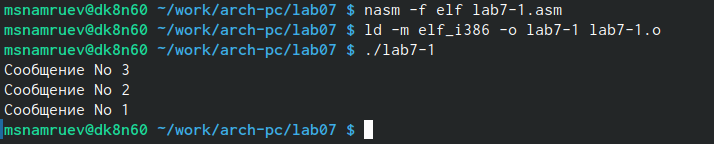
Запуск измененного файла

Далее изменяю файл так, чтобы он выводил “Сообщения” в обратном порядке.(рис. [??])



Редактирование файла

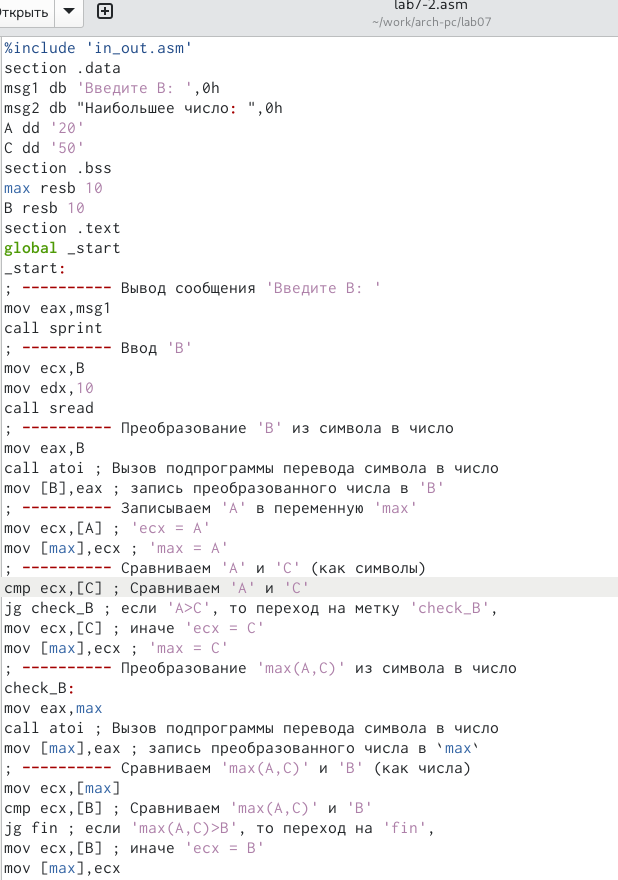
Создаю исполняемый файл и запускаю его.(рис. [??])



Запуск программы

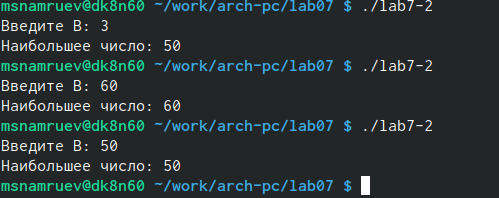
Убеждаюсь, что всё работает верно.

Создаю файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 и вставляю в него текст программы из листинга 7.3.(рис. [??])



Ввод текста из листинга 7.3

Создаю исполняемый файл и запускаю его.(рис. [??])



Запуск программы

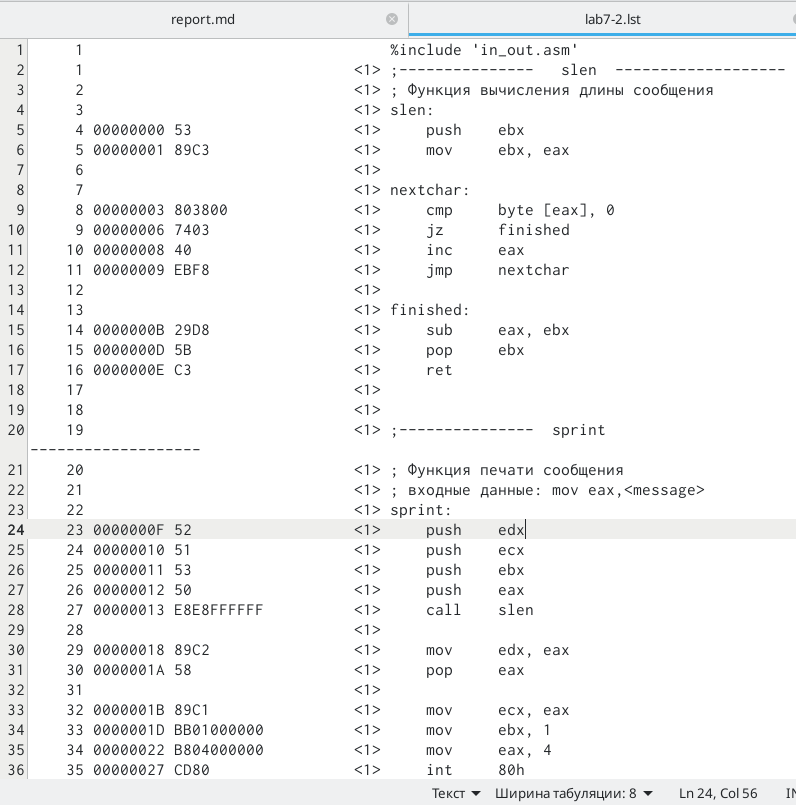
## 4.1 Изуение структуры файла листинга

Создаю файл лситинга для программы из файла lab7-2.asm.(рис. [??])

Создание файла листинга

Создание файла листинга

Далее открываю файл листинга lab7-2.lst с помощью текстового редактора gedit.(рис. [??])



Открывание файла листинга

Опишу строчку номер 16:

Здесь “15”-это номер строчки в коде программы “0000000D”- это адрес “5В”- это машинный код “ret” - исходный кол программы

Опишу строчку номер 36:

Здесь “35”-это номер строчки в коде программы “00000027”- это адрес “CD80”- это машинный код “int” - исходный кол программы

Опишу строчку номер 24:

Здесь “23”-это номер строчки в коде программы “0000000F”- это адрес “52”- это машинный код “push” - исходный кол программы

Теперь открываю файл с программой lab7-2.asm и удаляю один операнд в случайном месте.(рис. [??])

Удаление операнда

Удаление операнда

Пытаюсь создать файл листинга, но он не создается из-за ошибки.(рис. [??])

Попытка создание файла

Попытка создание файла

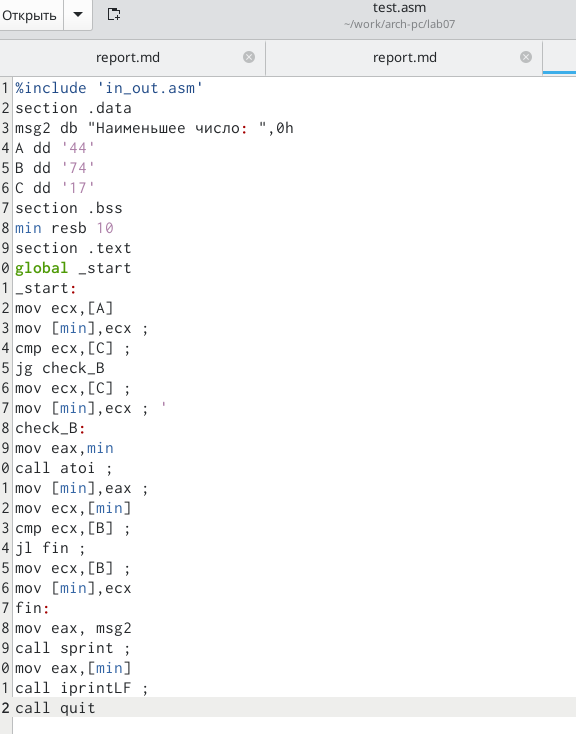
# 5 Задание для самостоятельной работы

1. Создаю файл test.asm.(рис. [??])

Создание файла для самостотельной работы

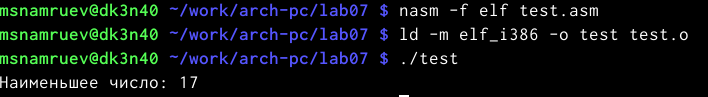
Создание файла для самостотельной работы

Далее открываю его и пишу программу для нахождения наименьшей из переменных a,b,c в соответствии с моим вариантом (вариант 16).(рис. [??])



Написание программы

Создаю исполняемый файл и запускаю его.(рис. [??])



Проверка работы программы

Программа работает верно

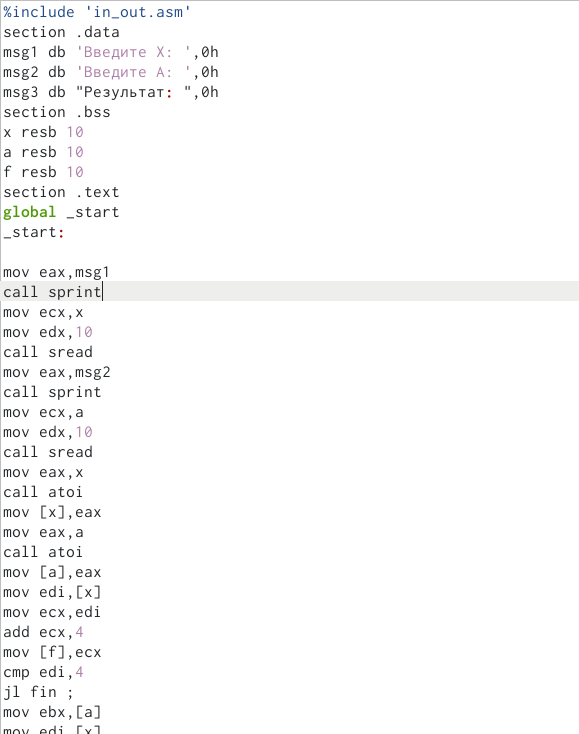
Код программы: %include ‘in\_out.asm’  
section .data  
msg2 db “Наименьшее число:”,0h  
A dd ‘44’  
B dd ‘74’  
C dd ‘17’  
section .bss  
min resb 10  
section .text  
global \_start  
\_start:  
mov ecx,[A]  
mov [min],ecx ;  
cmp ecx,[C] ;  
jg check\_B  
mov ecx,[C] ;  
mov [min],ecx ; ’  
check\_B:  
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

1. Создаю файл test2.asm.(рис. [??])

Создание файла

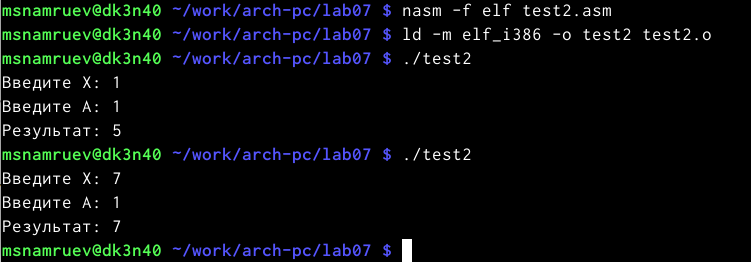
Создание файла

Начинаю написание программы, которая для введенных с клавиатуры значение x и a вычисляет значение заданной функции f(x) и выводит результат вычислений.(рис. [??])



Написание программы

Создаю исполняемый файл и запускаю его.(рис. [??])



Проверка программы

Программы работает верно

Код программы:  
%include ‘in\_out.asm’  
section .data  
msg1 db ‘Введите X:’,0h  
msg2 db ‘Введите A:’,0h  
msg3 db “Результат:”,0h  
section .bss  
x resb 10  
a resb 10  
f resb 10  
section .text  
global \_start  
\_start:

mov eax,msg1  
call sprint  
mov ecx,x  
mov edx,10  
call sread  
mov eax,msg2  
call sprint  
mov ecx,a  
mov edx,10  
call sread  
mov eax,x  
call atoi  
mov [x],eax  
mov eax,a  
call atoi  
mov [a],eax  
mov edi,[x]  
mov ecx,edi  
add ecx,4  
mov [f],ecx  
cmp edi,4  
jl fin ;  
mov ebx,[a]  
mov edi,[x]  
mul edi ;  
mov [f],eax ;  
fin:  
mov eax, msg3  
call sprint  
mov eax,[f]  
call iprintLF ;  
call quit

# 6 Выводы

После выполнения данной лабораторной работы я изучил команды условного и безусловного переходов, приобрел навыки написания программ с использованием переходов и познакомился с назнчением и структурой файла листинга.

# Список литературы