Отчет по лабораторной работе №7

Команды безусловного и условного переходов в Nasm. Программирование ветвлений

Мальянц Виктория Кареновна

Содержание

# 1 Цель работы

Изучить команды условного и безусловного переходов. Приобрести навыки написания программ с использованием переходов. Познакомиться с назначением и структурой файла листинга.

# 2 Задание

1. Реализация переходов в NASM
2. Изучение структуры файлы листинга
3. Выполнение заданий для самостоятельной работы

# 3 Выполнение лабораторной работы

## 3.1 Реализация переходов в NASM

Создаю каталог для программ лабораторной работы № 7, перехожу в него и создаю файл lab7-1.asm (рис. 1).

Рис. 1: Создание каталога и файла для программы

Рис. 1: Создание каталога и файла для программы

С помощью команды cp копирую файл in\_out.asm и просматриваю содержимое каталога lab07 с помощью команды ls (рис. 2).

Рис. 2: Копирование файла и просматривание содержимого каталога

Рис. 2: Копирование файла и просматривание содержимого каталога

Открываю файл lab7-1.asm в текстовом редакторе gedit через терминал (рис. 3).

Рис. 3: Открытие файла lab7-1.asm в текстовом редакторе gedit

Рис. 3: Открытие файла lab7-1.asm в текстовом редакторе gedit

Ввожу в файл lab7-1.asm программу с использованием инструкции jmp (рис. 4).

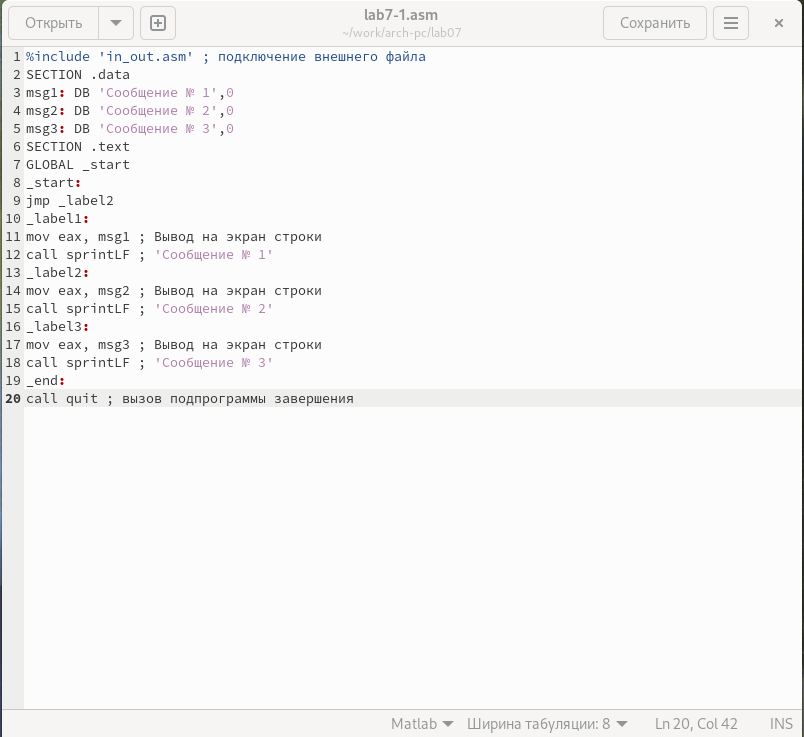


Рис. 4: Редактирование файла

Создаю исполняемый файл и запускаю его. Убеждаюсь в том, что безусловный переход изменяет порядок выволнения инструкций (рис. 5).

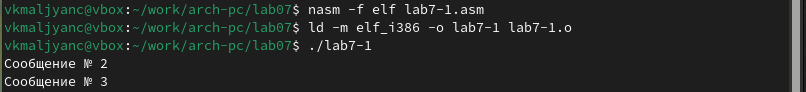


Рис. 5: Запуск исполняемого файла

Открываю файл lab7-1.asm в текстовом редакторе gedit через терминал (рис. 6).

Рис. 6: Открытие файла lab7-1.asm в текстовом редакторе gedit

Рис. 6: Открытие файла lab7-1.asm в текстовом редакторе gedit

Изменяю программу таким образом, чтобы сначала выводилось “Сообщение № 2”, а затем “Сообщение № 3” (рис. 7).

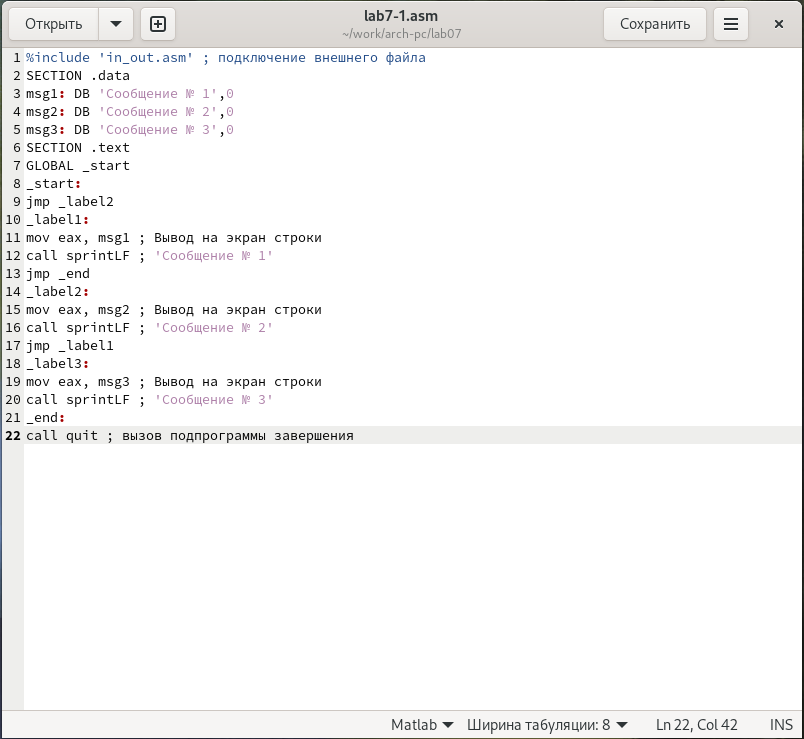


Рис. 7: Редактирование файла

Создаю исполняемый файл и запускаю его. Убеждаюсь в том, что изменения применены корректно (рис. 8).

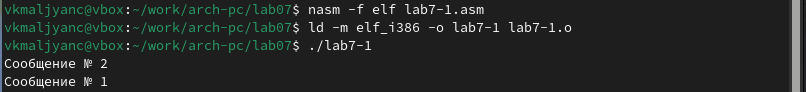


Рис. 8: Запуск исполняемого файла

Открываю файл lab7-1.asm в текстовом редакторе gedit через терминал (рис. 9).

Рис. 9: Открытие файла lab7-1.asm в текстовом редакторе gedit

Рис. 9: Открытие файла lab7-1.asm в текстовом редакторе gedit

Изменяю программу таким образом, чтобы сначала выводилось “Сообщение № 3”, потом “Сообщение № 2”, а затем “Сообщение № 1” (рис. 10).

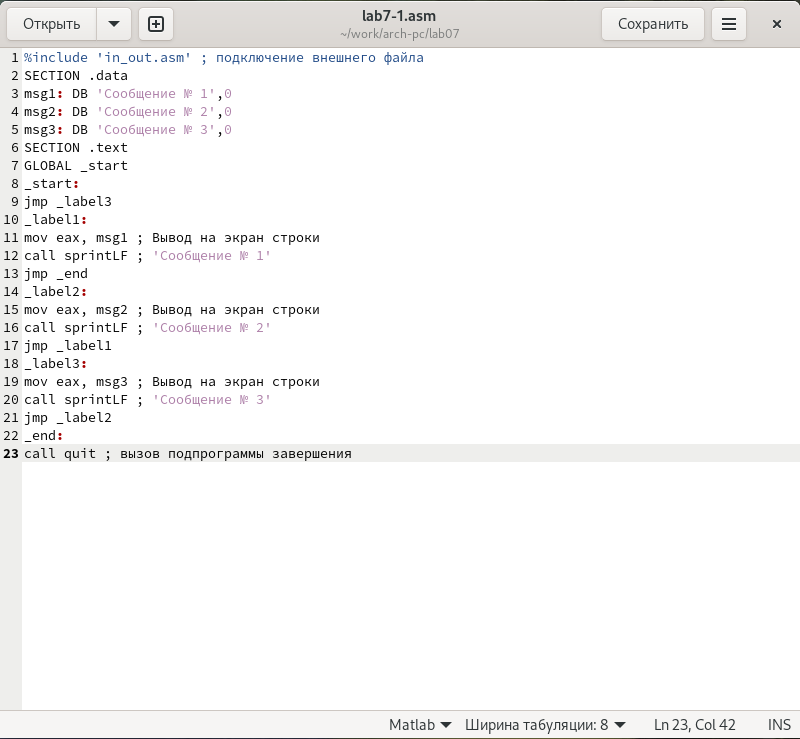


Рис. 10: Редактирование файла

Создаю исполняемый файл и запускаю его. Убеждаюсь в том, что изменения применены корректно (рис. 11).

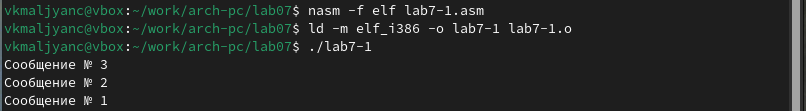


Рис. 11: Запуск исполняемого файла

С помощью команды touch создаю файл lab7-2.asm (рис. 12).

Рис. 12: Создание файла

Рис. 12: Создание файла

Открываю файл lab7-2.asm в текстовом редакторе gedit через терминал (рис. 13).

Рис. 13: Открытие файла lab7-2.asm в текстовом редакторе gedit

Рис. 13: Открытие файла lab7-2.asm в текстовом редакторе gedit

Ввожу в файл lab7-2.asm программу, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: A,B и C (рис. 14) (рис. 15).

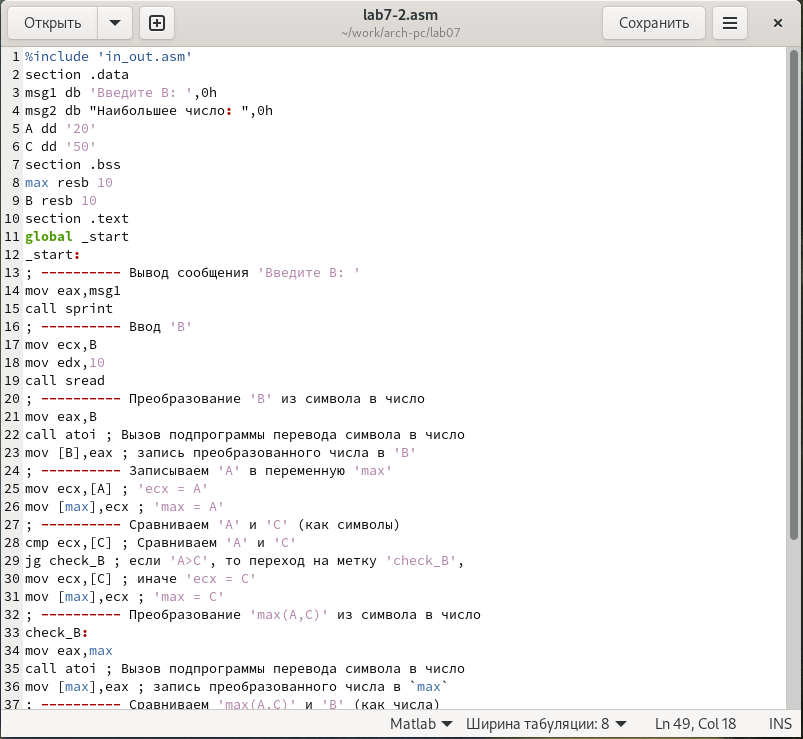


Рис. 14: Редактирование файла

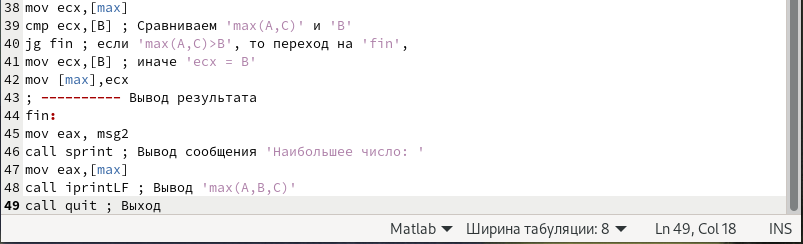


Рис. 15: Редактирование файла

Создаю исполняемый файл и запускаю его. Проверяю работу исполняемого файла для значения B равного 10 (рис. 16), B равного 30 (рис. 17) и B равного 60 (рис. 18). Убеждаюсь в том, что программа корректно выводит на экран наибольшую из 3 целочисленных переменных: A,B и C.

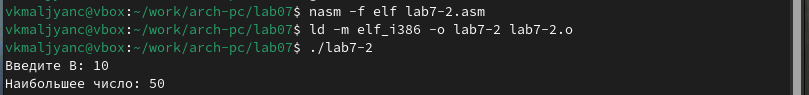


Рис. 16: Запуск исполняемого файла

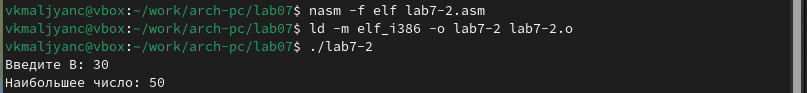


Рис. 17: Запуск исполняемого файла

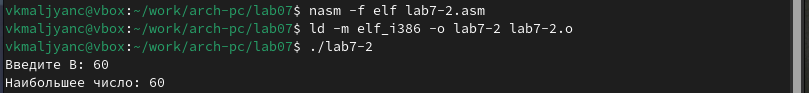


Рис. 18: Запуск исполняемого файла

## 3.2 Изучение структуры файлы листинга

Создаю файл листинга для программы из файла lab7-2.lst, указав ключ -l и задав имя файла листинга в командной строке (рис. 19)

Рис. 19: Создание файла листинга

Рис. 19: Создание файла листинга

Открываю файл lab7-2.lst в текстовом редакторе mcedit через терминал (рис. 20).

Рис. 20: Открытие файла lab7-2.lst в текстовом редакторе mcedit

Рис. 20: Открытие файла lab7-2.lst в текстовом редакторе mcedit

Объяснение содержимого трех строк файла листинга:

1. Строка под номером 15 (рис. 21). Первое значение - номер строки (15), второе вхождение - адрес (000000ED), третье вхождение - машинный код (E81DFFFFFF, из которого E8 - префикс команды call, 1DFFFFFF - смещение до адреса подпрограммы sprint), четвертое вхождение - инструкция (call sprint - команда, которая вызывает функцию sprint).

Рис. 21: Строка под номером 15

Рис. 21: Строка под номером 15

1. Строка под номером 19 (рис. 22). Первое значение - номер строки (19), второе вхождение - адрес (000000FC), третье вхождение - машинный код (E842FFFFFF, из которого E8 - префикс команды call, 42FFFFFF - смещение до адреса подпрограммы sread), четвертое вхождение - инструкция (call sread - команда, которая вызывает функцию sread).

Рис. 22: Строка под номером 19

Рис. 22: Строка под номером 19

1. Строка под номером 22 (рис. 23). Первое значение - номер строки (22), второе вхождение - адрес (00000106), третье вхождение - машинный код (E891FFFFFF, из которого E8 - префикс команды call, 91FFFFFF - смещение до адреса подпрограммы atoi), четвертое вхождение - инструкция (call atoi - команда, которая вызывает функцию atoi).

Рис. 23: Строка под номером 22

Рис. 23: Строка под номером 22

Открываю файл lab7-2.asm в текстовом редакторе gedit через терминал (рис. 24).

Рис. 24: Открытие файла lab7-2.asm в текстовом редакторе gedit

Рис. 24: Открытие файла lab7-2.asm в текстовом редакторе gedit

Удаляю из строки 38 операнд [max] (рис. 25).

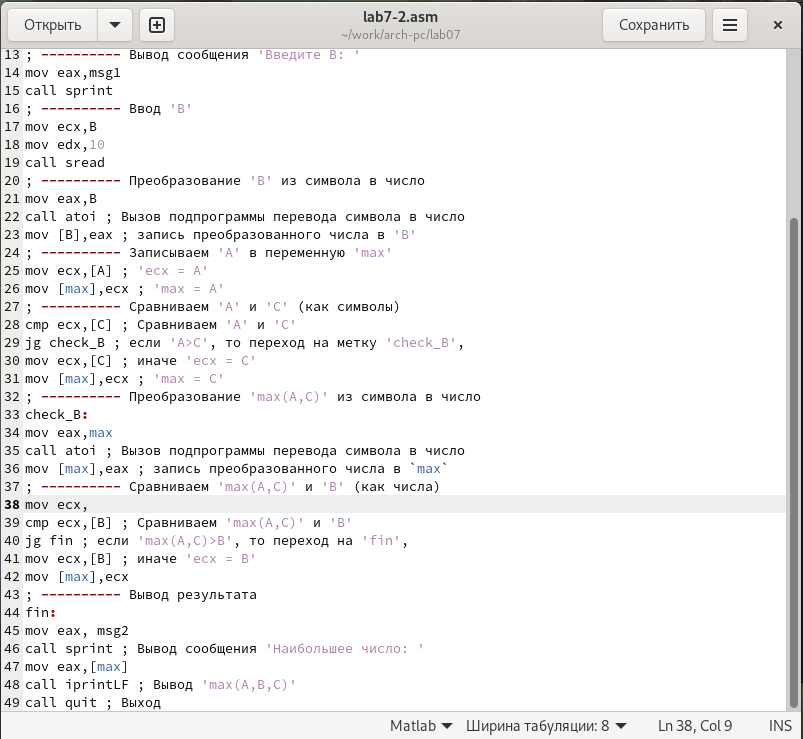


Рис. 25: Редактирование файла

Выполняю трансляцию с получением файла листинга (рис. 26). Никакие выходные файла кроме листинга не создаются.

Рис. 26: Запуск исполняемого файла

Рис. 26: Запуск исполняемого файла

Открываю файл lab7-2.lst в текстовом редакторе mcedit через терминал (рис. 27).

Рис. 27: Открытие файла lab7-2.lst в текстовом редакторе mcedit

Рис. 27: Открытие файла lab7-2.lst в текстовом редакторе mcedit

В листинге добавляется ошибка (рис. 28).

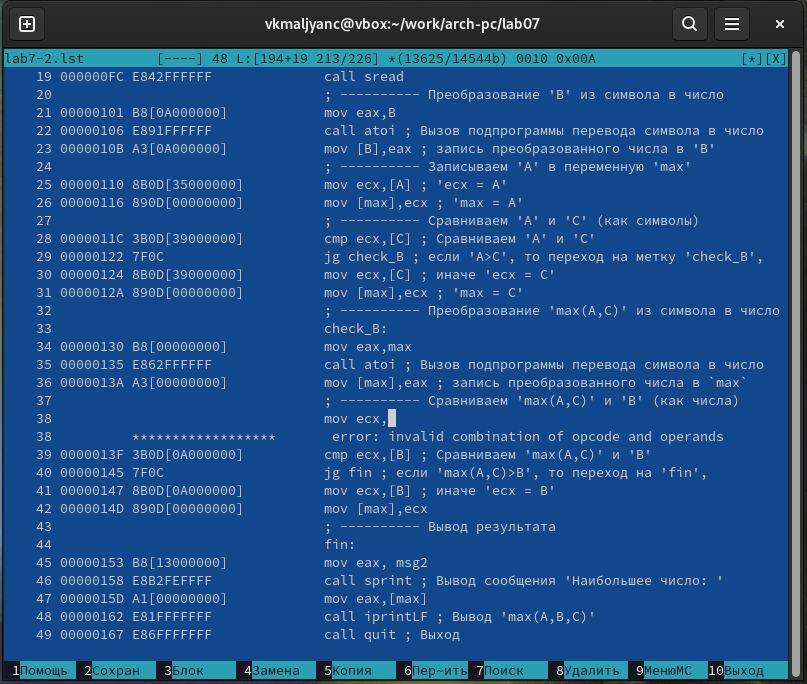


Рис. 28: Просмотр файла

## 3.3 Выполнение заданий для самостоятельной работы

Задание № 1. С помощью команды touch создаю файл lab7-3.asm (рис. 29).

Рис. 29: Создание файла

Рис. 29: Создание файла

Открываю файл lab7-3.asm в текстовом редакторе gedit через терминал (рис. 30).

Рис. 30: Открытие файла lab7-3.asm в текстовом редакторе gedit

Рис. 30: Открытие файла lab7-3.asm в текстовом редакторе gedit

Ввожу в файл lab7-3.asm программу для нахождения наименьшей из 3 целочисленных переменных a,b и c. Ввожу функцию из варианта №1 (рис. 31).

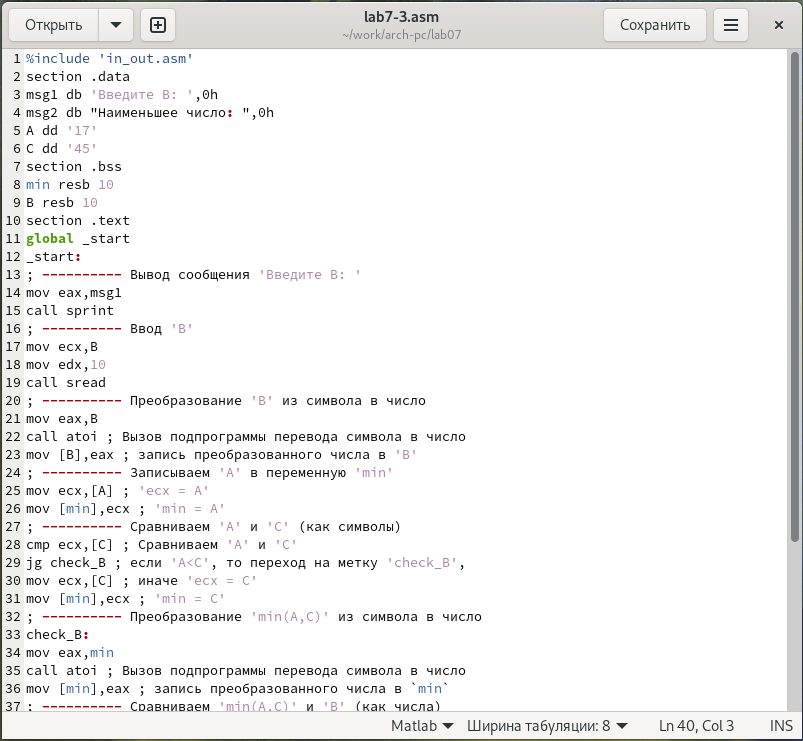


Рис. 31: Редактирование файла

Создаю исполняемый файл и запускаю его. Задаю значение B равное 23 (рис. 32). Убеждаюсь в том, что результат выводится корректно.

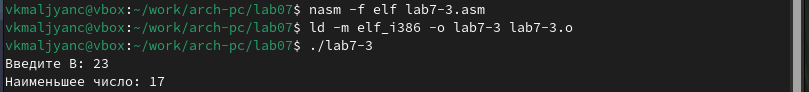


Рис. 32: Запуск исполняемого файла

Листинг программы:

%include 'in\_out.asm'  
section .data  
msg1 db 'Введите B: ',0h  
msg2 db "Наименьшее число: ",0h  
A dd '17'  
C dd '45'  
section .bss  
min resb 10  
B resb 10  
section .text  
global \_start  
\_start:  
; ---------- Вывод сообщения 'Введите B: '  
mov eax,msg1  
call sprint  
; ---------- Ввод 'B'  
mov ecx,B  
mov edx,10  
call sread  
; ---------- Преобразование 'B' из символа в число  
mov eax,B  
call atoi ; Вызов подпрограммы перевода символа в число  
mov [B],eax ; запись преобразованного числа в 'B'  
; ---------- Записываем 'A' в переменную 'min'  
mov ecx,[A] ; 'ecx = A'  
mov [min],ecx ; 'min = 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 [min],ecx ; 'min = C'  
; ---------- Преобразование 'min(A,C)' из символа в число  
check\_B:  
mov eax,min  
call atoi ; Вызов подпрограммы перевода символа в число  
mov [min],eax ; запись преобразованного числа в `min`  
; ---------- Сравниваем 'min(A,C)' и 'B' (как числа)  
mov ecx,[min]  
cmp ecx,[B] ; Сравниваем 'min(A,C)' и 'B'  
jb fin ; если 'min(A,C)<B', то переход на 'fin',  
mov ecx,[B] ; иначе 'ecx = B'  
mov [min],ecx  
; ---------- Вывод результата  
fin:  
mov eax, msg2  
call sprint ; Вывод сообщения 'Наименьшее число: '  
mov eax,[min]  
call iprintLF ; Вывод 'min(A,B,C)'  
call quit ; Выход

Задание № 2. С помощью команды touch создаю файл lab7-4.asm (рис. 33).

Рис. 33: Создание файла

Рис. 33: Создание файла

Открываю файл lab7-4.asm в текстовом редакторе gedit через терминал (рис. 34).

Рис. 34: Открытие файла lab7-4.asm в текстовом редакторе gedit

Рис. 34: Открытие файла lab7-4.asm в текстовом редакторе gedit

Ввожу в файл lab7-4.asm программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции f(x) и выводит результат вычислений. Ввожу функцию из варианта №1 (рис. 35).

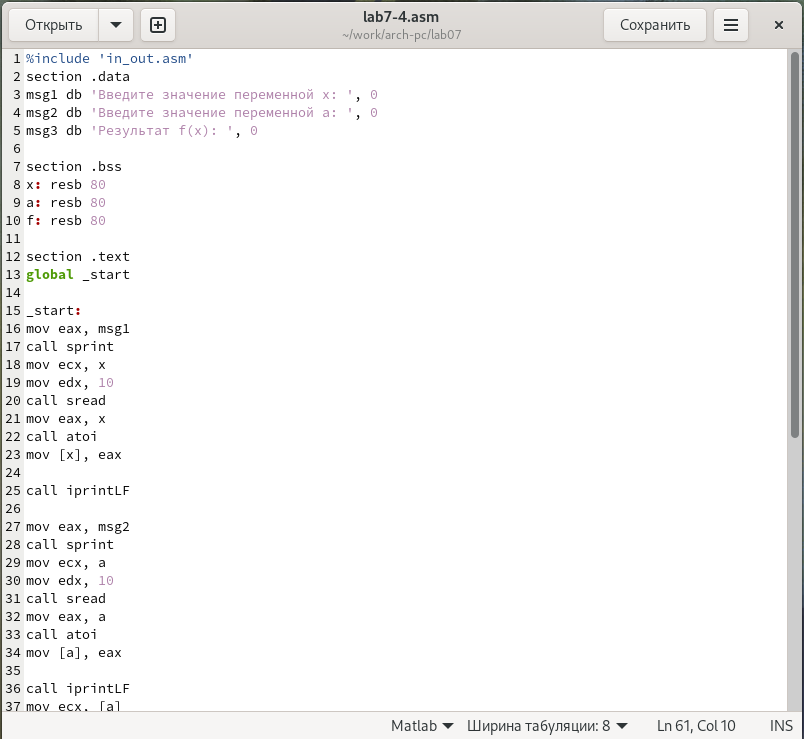


Рис. 35: Редактирование файла

Создаю исполняемый файл и запускаю его. Проверяю работу исполняемого файла для значения x равного 1 и a равного 2 (рис. 36) и x равного 2 и a равного 1 (рис. 37). Убеждаюсь в том, что результат программы выводится корректно.

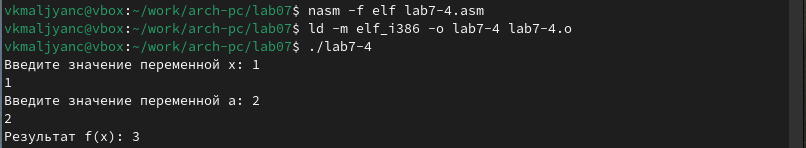


Рис. 36: Запуск исполняемого файла

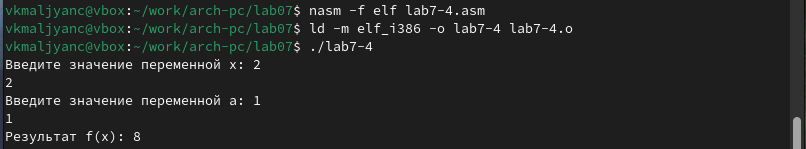


Рис. 37: Запуск исполняемого файла

Листинг программы:

%include 'in\_out.asm'  
section .data  
msg1 db 'Введите значение переменной x: ', 0  
msg2 db 'Введите значение переменной a: ', 0  
msg3 db 'Результат f(x): ', 0  
  
section .bss  
x: resb 80  
a: resb 80  
f: resb 80  
  
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  
  
call iprintLF   
  
mov eax, msg2  
call sprint  
mov ecx, a  
mov edx, 10  
call sread  
mov eax, a  
call atoi  
mov [a], eax  
  
call iprintLF   
mov ecx, [a]  
cmp ecx, [x]  
jge less  
  
mov edx, 8  
mov [f], edx  
jmp fin  
  
less:  
mov ebx, [x]  
mov ax, 2  
mul ax  
sub eax, ebx  
mov [f], eax  
  
fin:  
mov eax, msg3  
call sprint  
mov eax, [f]  
call iprintLF  
call quit

# 4 Выводы

Я изучила команды условного и безусловного переходов. Приобрела навыки написания программ с использованием переходов. Познакомилась с назначением и структурой файла листинга.