Отчёт по лабораторной работе №7

Дисциплина: архитектура компьютера

Бондаренко Кристина Антоновна

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 Выполнение лабораторной работы

С помощью утилиты mkdir создаю директорию lab07, перехожу в нее и создаю файл для работы. (рис. [[1](#fig:001)])

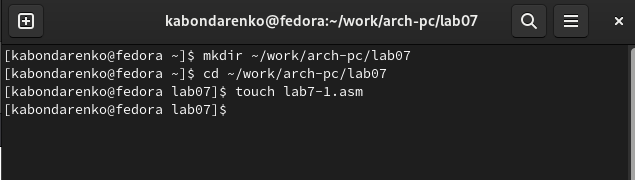


Figure 1: Создание директории

Открываю созданный файл lab7-1.asm, вставляю в него программу реализации безусловных переходов (рис. [[2](#fig:002)]).

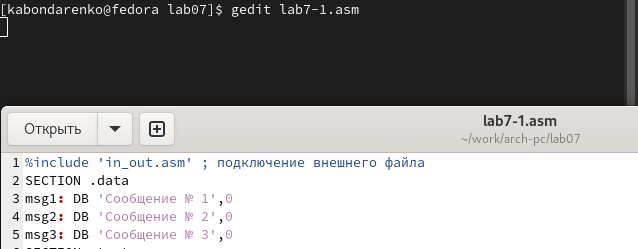


Figure 2: Создание копии файла для дальнейшей работы

Создаю исполняемый файл программы и запускаю его. Инструкции jmp \_label2 меняет порядок исполнения инструкций и позволяет выполнить инструкции начиная с метки \_label2.(рис. [[3](#fig:003)]).

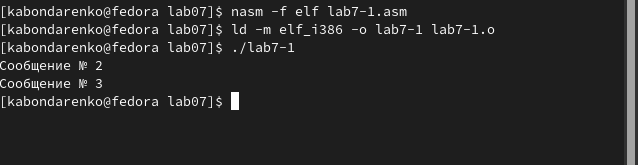


Figure 3: исполняемый файл

Изменяю текст программы так, чтобы она выводила сначала ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу. (рис. [[4](#fig:004)]).

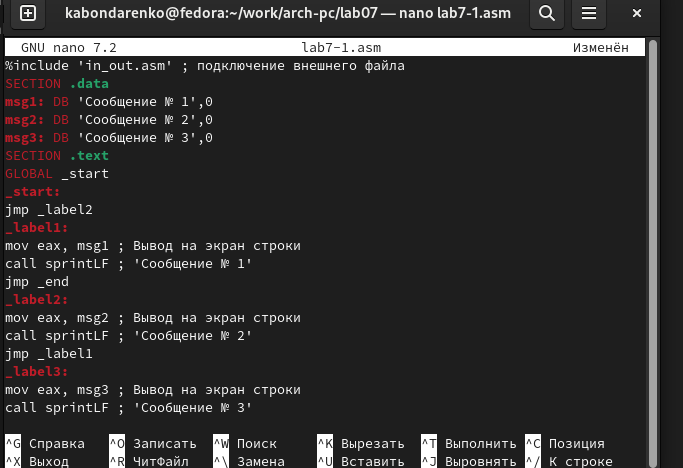


Figure 4: редактирование файла

Создаю новый исполняемый файл программы и запускаю его. Убеждаюсь в том, программа раотает верно.(рис. [[5](#fig:005)]).

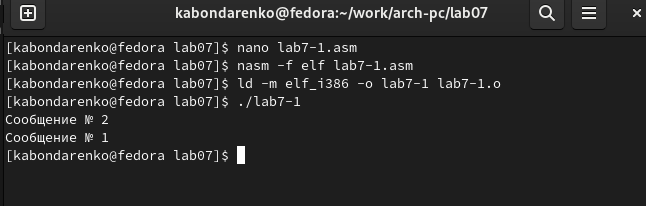


Figure 5: исполняемый файл

Изменяю текст программы, так чтобы вывод происходил в обратном порядке (рис. [[6](#fig:006)]).

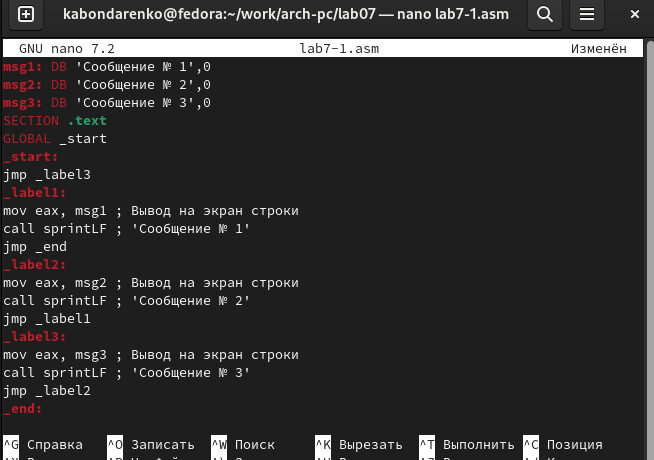


Figure 6: Изменяю текст

Создаю исполняемый файл и проверяю работу программы. Программа отработало верно.(рис. [[7](#fig:007)]).

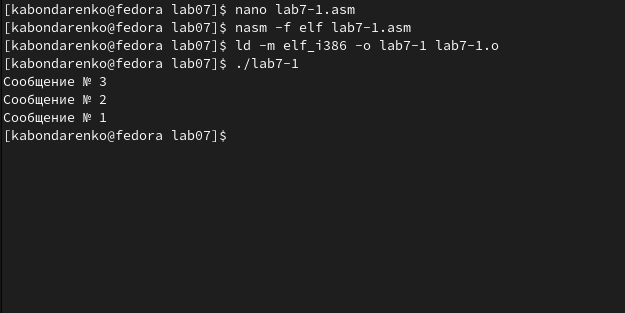


Figure 7: исполняемый файл

Создаю новый файл lab7-2.asm для программы с условным оператором.(рис. [[8](#fig:008)]).

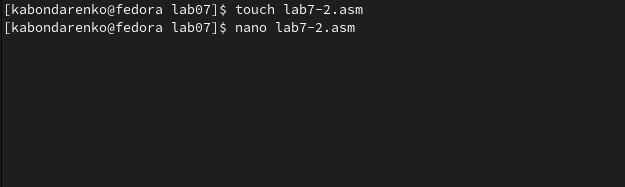


Figure 8: Создание новый файл

Вставляю программу, которая определяет и выводит на экран наибольшее число (рис. [[9](#fig:009)]).

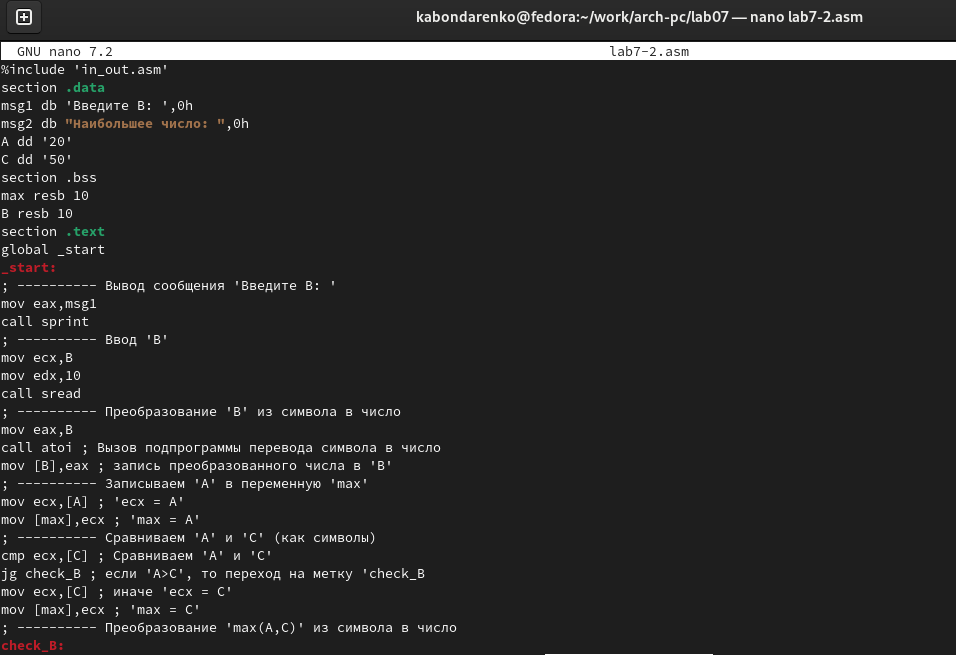


Figure 9: редактирование файла

Создаю и запускаю новый исполняемый файл, проверяю работу программы для разных B при А=20 и С=50 (рис.[[10](#fig:010)]).

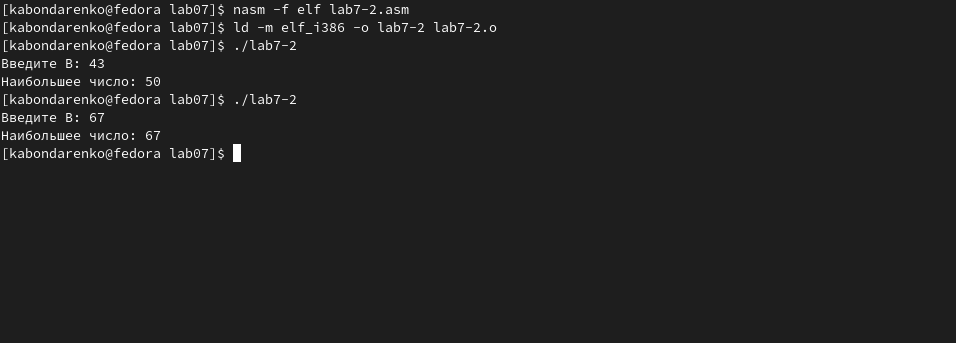


Figure 10: запускаю исполняемый файл,

Создаю файл листинга для программы в файле lab7-2.asm(рис. [[11](#fig:011)]).

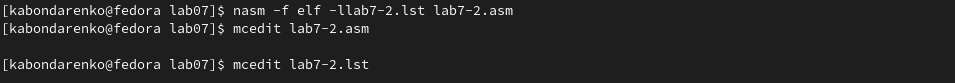


Figure 11: файл листинга

Открываю файл листинга с помощью редактора mcedit. Расмотрим 9-11 строки:(рис. [[12](#fig:012)]).

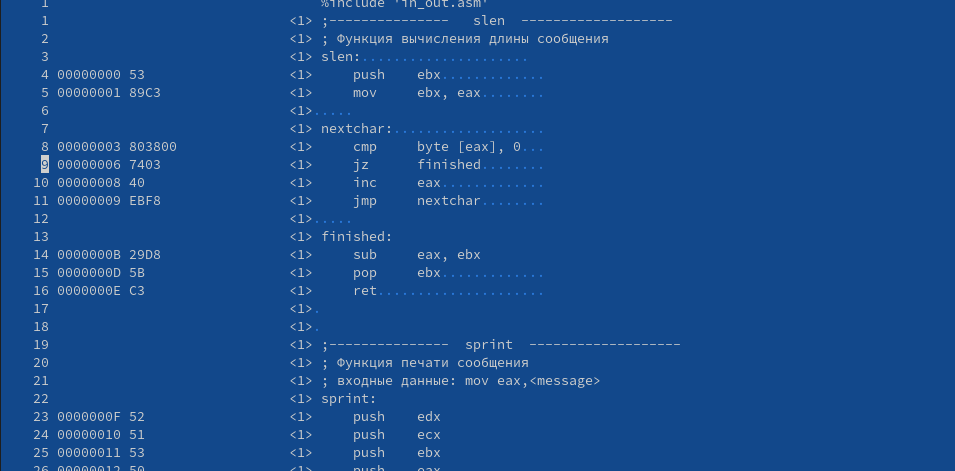


Figure 12: Редактирование файла

9 строка:

* Перые цифры [9] - это номер строки файла листинга.
* Cледующие цифры [00000006] адрес — это смещение машинного кода от начала текущего сегмента, состоит из 8 чисел.
* следующие числа [7403] - это машинный код, который представляет собой ассемблированную исходную строку в виде шестнадцатеричной последовательности, поэтоу и появляются буквы латынского алфавита.
* следющее [jz finished] - исходный текст программы, которая просто состоит из строкк исходной программы вместе с комментариями.

10 строка:

* Перые цифры [10] - это номер строки файла листинга.
* Cледующие цифры [00000008] адрес — это смещение машинного кода от начала текущего сегмента, состоит из 8 чисел.
* следующие числа [40] - это машинный код, который представляет собой ассемблированную исходную строку в виде шестнадцатеричной последовательности, поэтоу и появляются буквы латынского алфавита.
* следющее [inc eax] - исходный текст программы, которая просто состоит из строкк исходной программы вместе с комментариями

11 строка:

* Перые цифры [11] - это номер строки файла листинга.
* Cледующие цифры [00000009] адрес — это смещение машинного кода от начала текущего сегмента, состоит из 8 чисел.
* следующие числа [EBF8] - это машинный код, который представляет собой ассемблированную исходную строку в виде шестнадцатеричной последовательности, поэтоу и появляются буквы латынского алфавита.
* следющее [jmp nextchar] - исходный текст программы, которая просто состоит из строкк исходной программы вместе с комментариями

Открываю файл lab7-2.asm с помощью редактора и Удаляю один операнд в инструкции cmp. (рис. [[13](#fig:013)]).

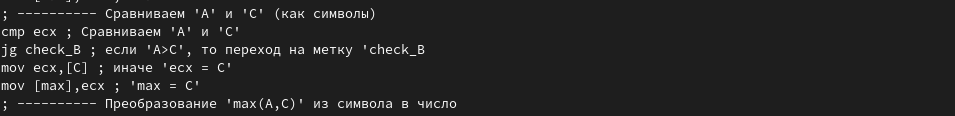


Figure 13: Файл листинга

Открываю файл листинга с помощью редактора mcedit и замечаю, что в файле листинга появляется ошибка. (рис. [[14](#fig:014)]).

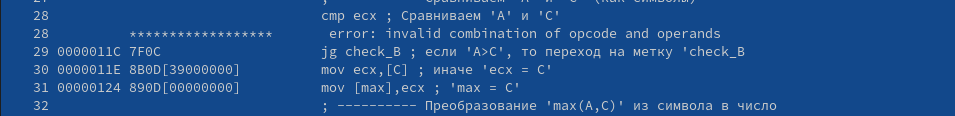


Figure 14: Файл листинга

Отсюда можно сделать вывод, что, если в коде появляется ошибка, то ее описание появится в файле листин

# 3 Самостоятельная работа

Создаю файл lab7-3.asm с помощью утилиты touch (рис. [[15](#fig:016)]).



Figure 15: Создание файла

Ввожу в созданный файл текст программы для вычисления наименьшего из 3 чисел. Числа беру, учитывая свой вариант из прошлой лабораторной работы. 5 вариант (рис. [[16](#fig:017)]).

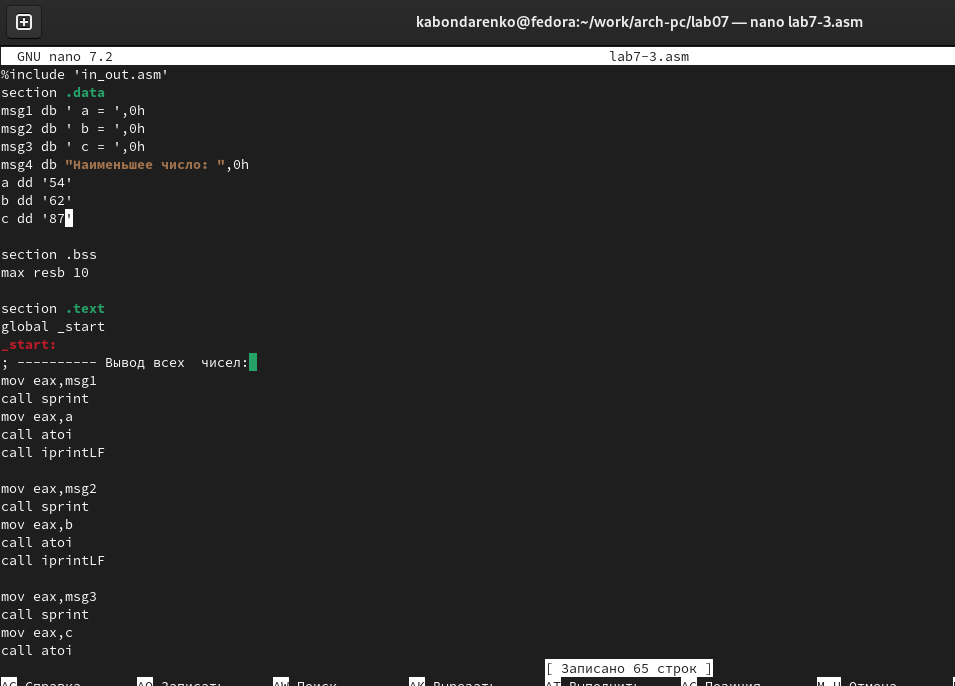


Figure 16: Редактирование файла

Создаю исполняемый файл и запускаю его (рис. [[17](#fig:018)]).

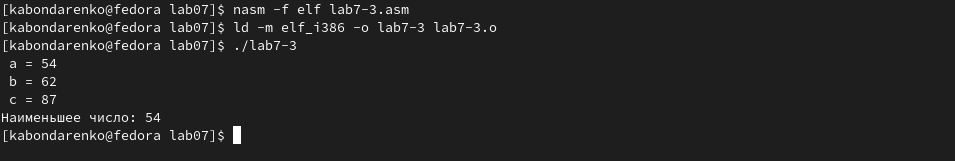


Figure 17: Запуск исполняемого файла

**Текст программы**

%include 'in\_out.asm'  
section .data  
msg1 db ' а = ',0h  
msg2 db ' b = ',0h  
msg3 db ' c = ',0h  
msg4 db "Наименьшее число: ",0h  
a dd '54'  
b dd '62'  
c dd '87'  
  
section .bss  
max resb 10  
  
section .text  
global \_start  
\_start:  
; ---------- Вывод всех чисел:   
mov eax,msg1  
call sprint  
mov eax,a  
call atoi  
call iprintLF  
  
mov eax,msg2  
call sprint  
mov eax,b  
call atoi  
call iprintLF  
  
mov eax,msg3  
call sprint  
mov eax,c  
call atoi  
call iprintLF  
  
;-------------сравнивание чисел  
mov eax,b  
call atoi ;перевод символа в число  
mov [b],eax ; запись преобразованного числа в b  
;------------ запись b в переменную мах  
mov ecx,[a] ;  
mov [max],ecx ;  
;------------сравнивание чисел a c  
cmp ecx,[c]; if a>c  
jl check\_b ; то перход на метку  
mov ecx,[c] ;  
mov [max],ecx ;  
;-------метка check\_b  
check\_b:  
mov eax,max ;  
call atoi  
mov [max],eax ;  
;------------  
mov ecx,[max] ;  
cmp ecx,[b] ;  
jl check\_c ;  
mov ecx,[b] ;  
mov [max],ecx ;  
;-------------  
check\_c:  
mov eax,msg4 ;  
call sprint ;   
mov eax,[max];  
call iprintLF ;  
call quit

Создаю новый файл lab7-4 и ввожу в него программу, которая выводит значения функции. Функцию беру из таблицы в соответствии со своим 5 вариантом (Вариант рис. [[19](#fig:019)]).

Figure 18: Редактирование файла

Figure 18: Редактирование файла



Figure 19: ввод программы в файл

Создаю испольняемый файл и проверяю её выполнение при (x=1, a=2) при (x=2 и a=1) (рис. [[20](#fig:020)]). Программа отработала верно!

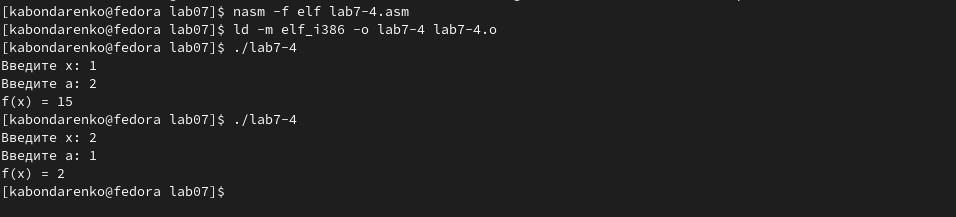


Figure 20: запуск исполняемого файла

**Текст программы**

%include 'in\_out.asm'  
section .data  
msg1 db 'Введите x: ',0h  
msg2 db 'Введите a: ',0h  
msg3 db 'f(x) = ',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 ecx,[a]  
cmp ecx,[x] ;x>a  
jl check\_a ;  
mov ebx,15  
mov ecx,ebx  
jmp \_end  
check\_a:  
mov ecx,[x]  
sub ecx,[a]  
mov eax,ecx  
mov ebx,2  
mul ebx  
mov ecx,eax  
;----------  
\_end:  
mov eax,msg3 ;  
call sprint ;  
mov eax,ecx ;  
call iprintLF;  
call quit ;

# 4 Выводы

При выполнении данной лабораторной работы я освоила инструкции условного и безусловного вывода и ознакомилась с структурой файла листинга.