Лабораторная работа №7

Команды безусловного и условного переходов в Nasm. Программирование ветвлений.

Демин Даниил

Содержание

# 1 Цель работы

Целью работы является приобретение практических навыков работы и изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. А так же знакомство с назначением и структурой файла листинга.

# 2 Выполнение лабораторной работы

Создал и першел в директория для лабораторной работы и сделал файл lab7-1.asm (рис. 1).

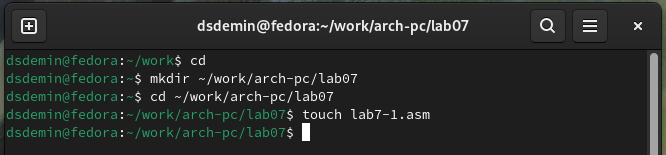


Рис. 1: Директория lab07

Переписал код из листинга (рис. 2).

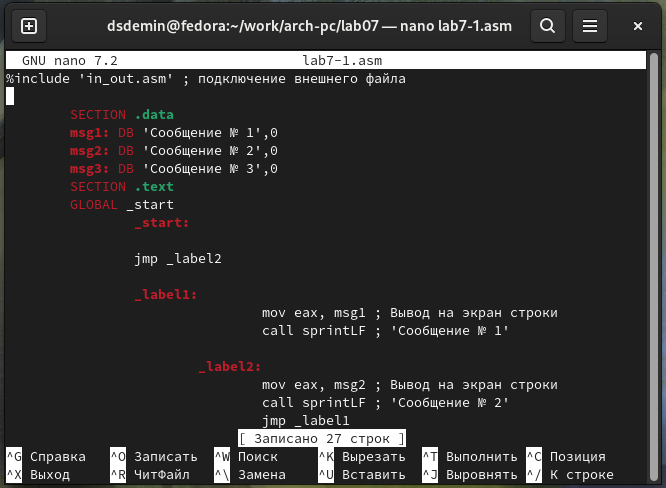


Рис. 2: Листинг кода

Листинг кода

%include 'in\_out.asm' ; подключение внешнего файла  
  
SECTION .data  
msg1: DB 'Сообщение № 1',0  
msg2: DB 'Сообщение № 2',0  
msg3: DB 'Сообщение № 3',0  
SECTION .text  
GLOBAL \_start  
\_start:  
  
jmp \_label2  
  
\_label1:  
mov eax, msg1 ; Вывод на экран строки  
call sprintLF ; 'Сообщение № 1'  
  
\_label2:  
mov eax, msg2 ; Вывод на экран строки  
call sprintLF ; 'Сообщение № 2'  
  
\_label3:  
mov eax, msg3 ; Вывод на экран строки  
call sprintLF ; 'Сообщение № 3'  
  
\_end:  
call quit ; вызов подпрограммы завершения

Создал и запустил исполняемый файл (рис. 3).

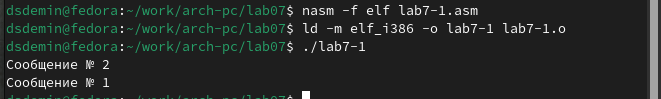


Рис. 3: Результат выполнения программы

Переписал код в файле с листинга (рис. 4).

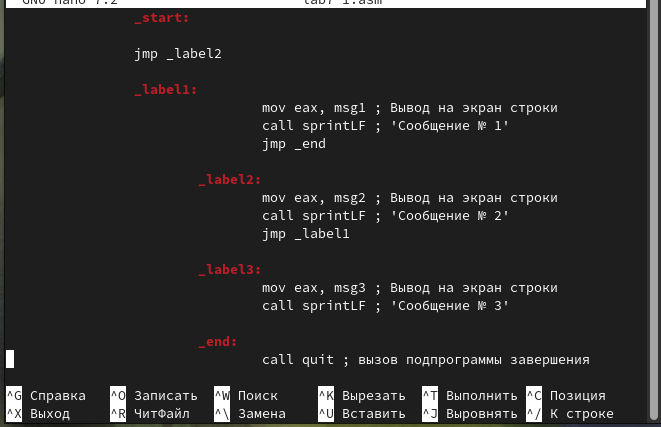


Рис. 4: Обновленный код

Листинг кода

%include 'in\_out.asm' ; подключение внешнего файла  
  
SECTION .data  
msg1: DB 'Сообщение № 1',0  
msg2: DB 'Сообщение № 2',0  
msg3: DB 'Сообщение № 3',0  
SECTION .text  
GLOBAL \_start  
\_start:  
  
jmp \_label2  
  
\_label1:  
mov eax, msg1 ; Вывод на экран строки  
call sprintLF ; 'Сообщение № 1'  
jmp \_end  
  
\_label2:  
mov eax, msg2 ; Вывод на экран строки  
call sprintLF ; 'Сообщение № 2'  
jmp \_label1  
  
\_label3:  
mov eax, msg3 ; Вывод на экран строки  
call sprintLF ; 'Сообщение № 3'  
  
\_end:  
call quit ; вызов подпрограммы завершения

Создал и запустил исполняемый файл. (рис. 5).

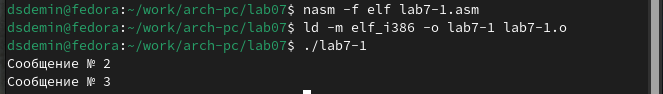


Рис. 5: Результат выполнения программы

Переписал код так, что бы он выводил сообщения в необходимом порядке (рис. 6).

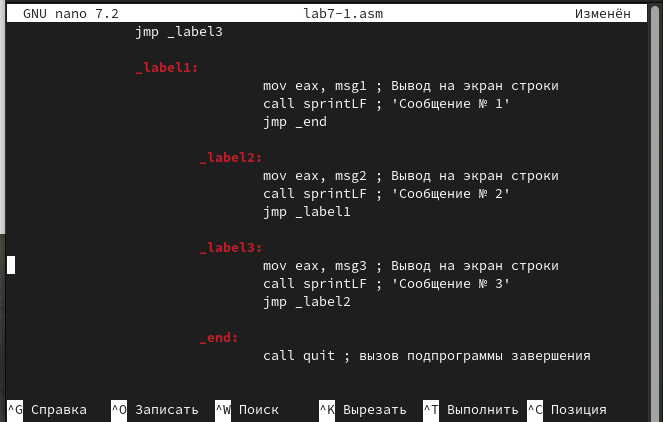


Рис. 6: Листинг кода

Листинг кода 7.2.2:

%include 'in\_out.asm' ; подключение внешнего файла  
  
SECTION .data  
msg1: DB 'Сообщение № 1',0  
msg2: DB 'Сообщение № 2',0  
msg3: DB 'Сообщение № 3',0  
SECTION .text  
GLOBAL \_start  
\_start:  
  
jmp \_label3  
  
\_label1:  
mov eax, msg1 ; Вывод на экран строки  
call sprintLF ; 'Сообщение № 1'  
jmp \_end  
  
\_label2:  
mov eax, msg2 ; Вывод на экран строки  
call sprintLF ; 'Сообщение № 2'  
jmp \_label1  
  
\_label3:  
mov eax, msg3 ; Вывод на экран строки  
call sprintLF ; 'Сообщение № 3'  
jmp \_label2  
  
\_end:  
call quit ; вызов подпрограммы завершения

Создал и запустил исполняемый файл. (рис. 7).

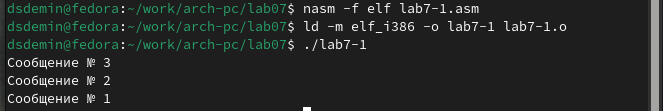


Рис. 7: Результат выполнения программы

Переписал в него код с листинга в новый файл (рис. 8).

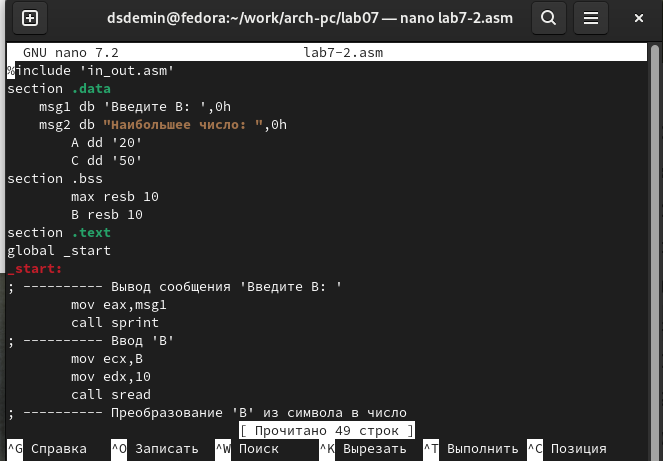


Рис. 8: Листинг кода

Листинг кода

%include 'in\_out.asm'  
section .data  
 msg1 db 'Введите B: ',0h  
 msg2 db "Наибольшее число: ",0h  
 A dd '20'  
 C dd '50'  
   
section .bss  
 max 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 'max'  
mov ecx,[A] ; 'ecx = A'  
mov [max],ecx ; 'max = 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 [max],ecx ; 'max = C'  
; ---------- Преобразование 'max(A,C)' из символа в число  
  
check\_B:  
mov eax,max  
call atoi ; Вызов подпрограммы перевода символа в число  
mov [max],eax ; запись преобразованного числа в `max`  
; ---------- Сравниваем 'max(A,C)' и 'B' (как числа)  
mov ecx,[max]  
cmp ecx,[B] ; Сравниваем 'max(A,C)' и 'B'  
jg fin ; если 'max(A,C)>B', то переход на 'fin',  
mov ecx,[B] ; иначе 'ecx = B'  
mov [max],ecx  
; ---------- Вывод результата  
fin:  
mov eax, msg2  
call sprint ; Вывод сообщения 'Наибольшее число: '  
mov eax,[max]  
call iprintLF ; Вывод 'max(A,B,C)'  
call quit ; Выход

Создал и запустил исполняемый файл. Проверил несколькими числами (рис. 9).

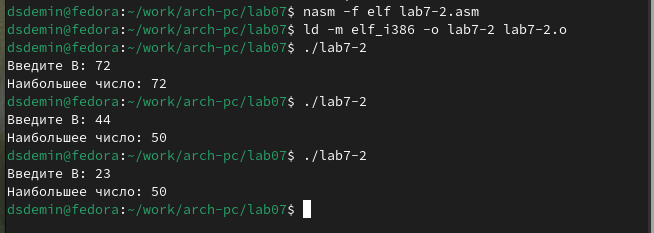


Рис. 9: Результат выполнения программы

Создал листинг кода lab7-2.lst при помощи команды nasm. помимо основной программы помимо листинга основной программы выше находиться код из in\_out.asm (рис. 10).

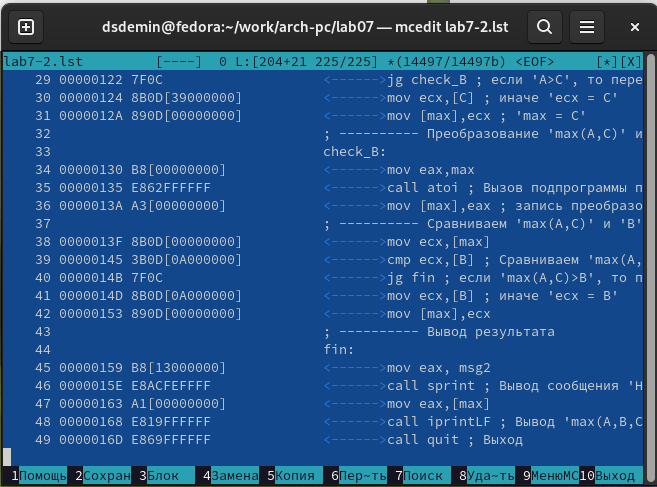


Рис. 10: Листинг кода

Выбрал набор команд в которой есть несколько операнд (рис. 11).

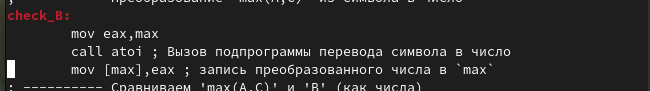


Рис. 11: Листинг кода

Удалил один из операндов (рис. 12).

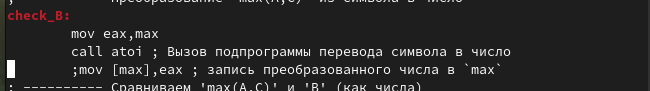


Рис. 12: Обновленный листинг

Создал исполняемый файл и запустил его (рис. 13).

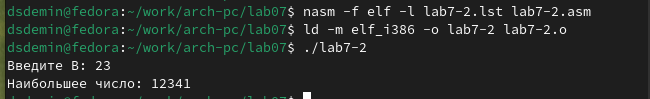


Рис. 13: Результат выполнения программы

При создании появились объектный, исполняемый и файл листинга (рис. 14).

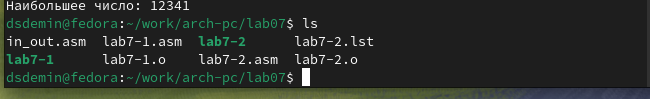


Рис. 14: Созданные файлы

# 3 Выполнение самостоятельной работы

Создал и написал код в файле lab7-4.asm для нахождения наименьшей из 3 целочисленных переменных а,b среди введенных чисел, соответсвующих варианту 8. (рис. 15).

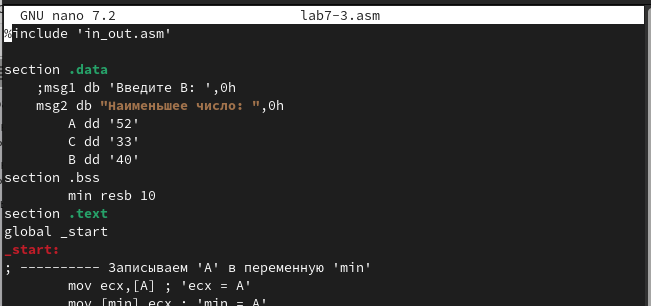


Рис. 15: Листинг кода

Листинг кода

%include 'in\_out.asm'  
  
section .data  
 ;msg1 db 'Введите B: ',0h  
 msg2 db "Наименьшее число: ",0h  
 A dd '52'  
 C dd '33'ы  
 B dd '40'  
section .bss  
 min resb 10  
section .text  
global \_start  
\_start:  
; ---------- Записываем 'A' в переменную 'min'  
 mov ecx,[A] ; 'ecx = A'  
 mov [min],ecx ; 'min = A'  
; ---------- Сравниваем 'A' и 'С' (как символы)  
 cmp [C],ecx ; 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 ; 'max = C  
; ---------- Преобразование 'max(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 [B],ecx ; Сравниваем 'max(A,C)' и 'B'  
 jg 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

Создал исполняемый файл и запустил его. Результат выполнения программы совпадает с ожиданием. (рис. 16).

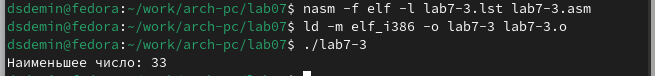


Рис. 16: Результат выполнения программы

Написал программу для подсчета результата функции f(x) в соответствии с 8 вариантом (рис. 17).

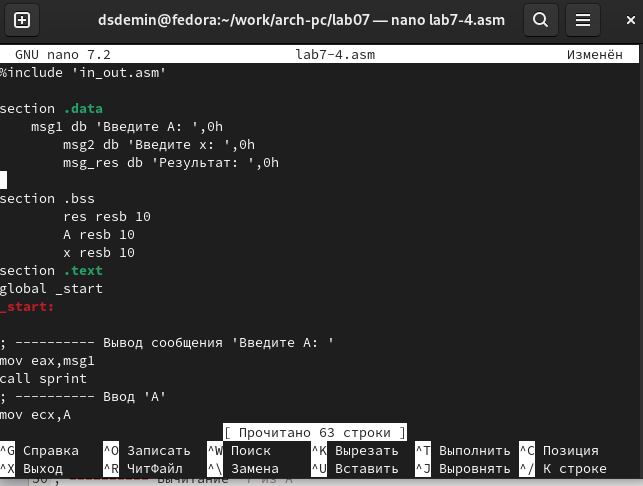


Рис. 17: Листинг кода

Листинг

%include 'in\_out.asm'  
  
section .data  
 msg1 db 'Введите A: ',0h  
 msg2 db 'Введите x: ',0h  
 msg\_res db 'Результат: ',0h  
  
section .bss  
 res resb 10  
 A resb 10  
 x resb 10  
section .text  
global \_start  
\_start:  
  
; ---------- Вывод сообщения 'Введите A: '  
mov eax,msg1  
call sprint  
; ---------- Ввод 'A'  
mov ecx,A  
mov edx,10  
call sread  
; ---------- Преобразование 'A' из символа в число  
mov eax,A  
call atoi ; Вызов подпрограммы перевода символа в число  
mov [A],eax ; запись преобразованного числа в 'A'  
  
; ---------- Вывод сообщения 'Введите B: '  
mov eax,msg2  
call sprint  
; ---------- Ввод 'x'  
mov ecx,x  
mov edx,10  
call sread  
; ---------- Преобразование 'x' из символа в число  
mov eax,x  
call atoi ; Вызов подпрограммы перевода символа в число  
mov [x],eax ; запись преобразованного числа в 'x'  
  
; ---------- Сравниваем 'A' и 3  
  
mov ecx, 3   
cmp [A],ecx ; Сравниваем 'A' и '3'  
jg check\_B ; если 'A>7', то переход на метку 'check\_B',  
mov eax,[A] ; иначе 'f = ax'  
mul ecx  
mov [res], eax  
jmp fin  
  
; ---------- Вычитание '7 из А'   
check\_B:  
mov eax, [A]  
inc eax  
mov [res], eax  
jmp fin  
  
; ---------- Вывод результата  
fin:  
mov eax, msg\_res  
call sprint ; Вывод сообщения 'Результат: '  
mov eax,[res]  
call iprintLF ; Вывод 'f(x)='  
call quit ; Выход

Создал и запустил исполняемый файл. Результат совпадает с ответом, полученнымм аналитическим путем. (рис. 18).

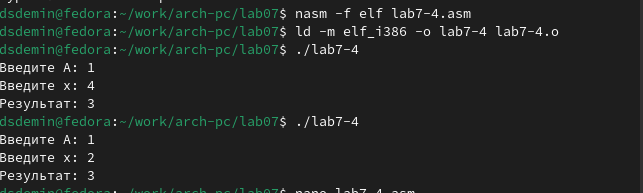


Рис. 18: Результат выполнения программы

# 4 Выводы

Выполнив данную лабараторную работу, я обрел практические навыки работы и изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. А так же знакомство с назначением и структурой файла листинга.