Отчет по лабораторной работе №7

Дисциплина: архитектура компьютера

Гончарь Анастасия Александровна

Содержание

# 1 Цель работы

Целью данной лабораторной работы является изучение команд условного и безусловного переходов, приобретение навыков написания программ с использованием переходов и знакомство с назначением и структурой файла листинга.

# 2 Задание

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

# 3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. 1 приведено краткое описание стандартных каталогов Unix.

Таблица 1: Описание некоторых каталогов файловой системы GNU Linux

| Имя каталога | Описание каталога |
| --- | --- |
| / | Корневая директория, содержащая всю файловую |
| /bin | Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям |
| /etc | Общесистемные конфигурационные файлы и файлы конфигурации установленных программ |
| /home | Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя |
| /media | Точки монтирования для сменных носителей |
| /root | Домашняя директория пользователя root |
| /tmp | Временные файлы |
| /usr | Вторичная иерархия для данных пользователя |

Более подробно про Unix см. в [1–4].

# 4 Выполнение лабораторной работы

## 4.1 Реализация переходов в NASM

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

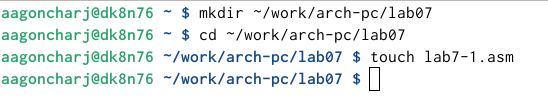


Рис. 1: Cоздание каталога и файла lab8-1.asm

Ввожу в файл lab7-1.asm текст программы из листинга 7.1 (рис. 2).

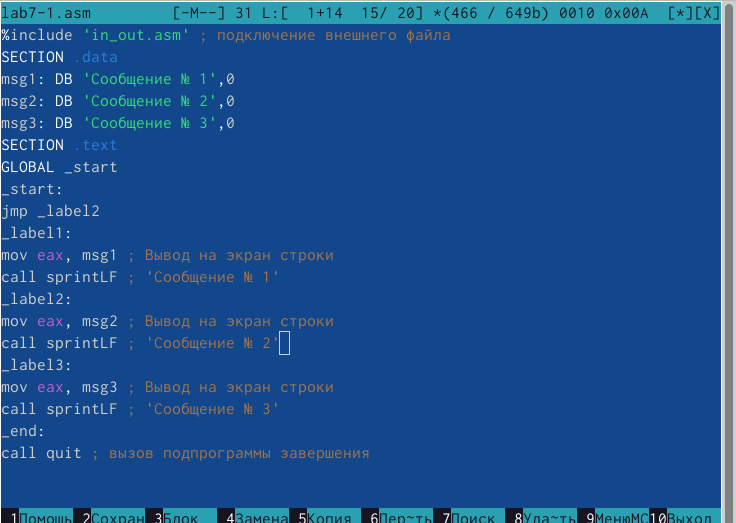


Рис. 2: Текст программы

Создаю исполняемый файл и запускаю его (рис. 3).

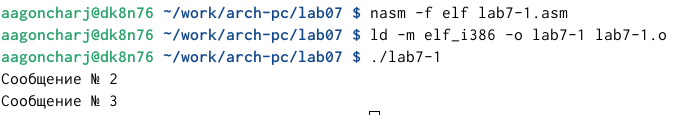


Рис. 3: Запуск файла

Изменяю текст программы в файле lab7-1.asm в соответствии с листингом 7.2 (рис. 4)

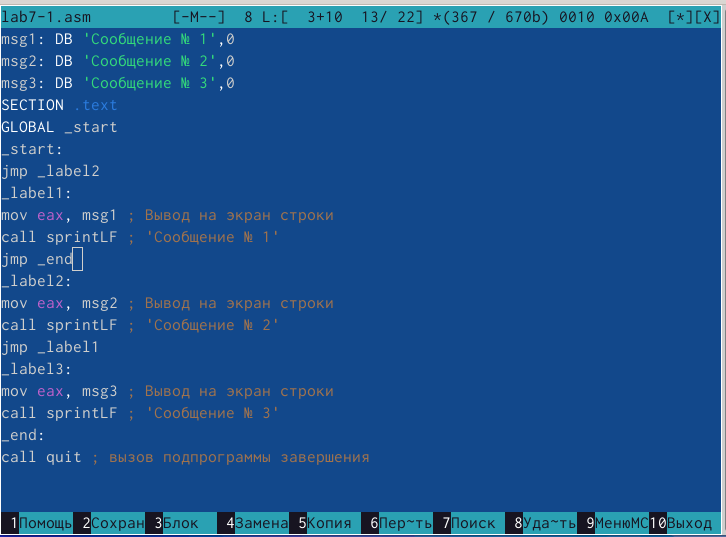


Рис. 4: Изменение текста программы

Создаю исполняемый файл и запускаю его (рис. 5).

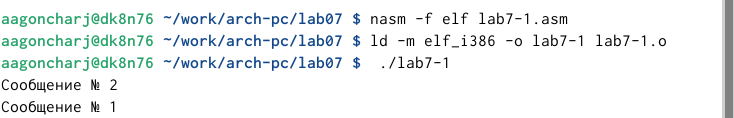


Рис. 5: Запуск файла

Теперь изменяю текст программы в этом же файле, чтобы программа выводила сначала “Сообщение №3”,затем “Сообщение №2” и “Сообщение №1” (рис. 6)

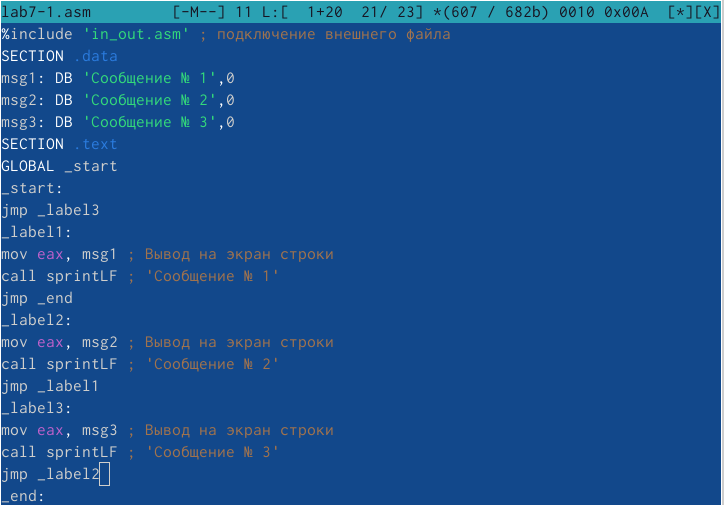


Рис. 6: Изменение текста программы

Создаю исполняемый файл и запускаю его (рис. 7).

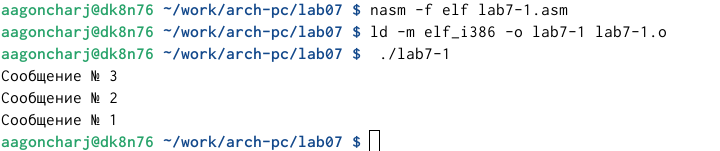


Рис. 7: Запуск файла

Создаю файл lab7-2.asm (рис. 8).

Рис. 8: Создание файла

Рис. 8: Создание файла

Ввожу в этот файл текст программы из листинга 7.3 (рис. **¿fig:009?**).

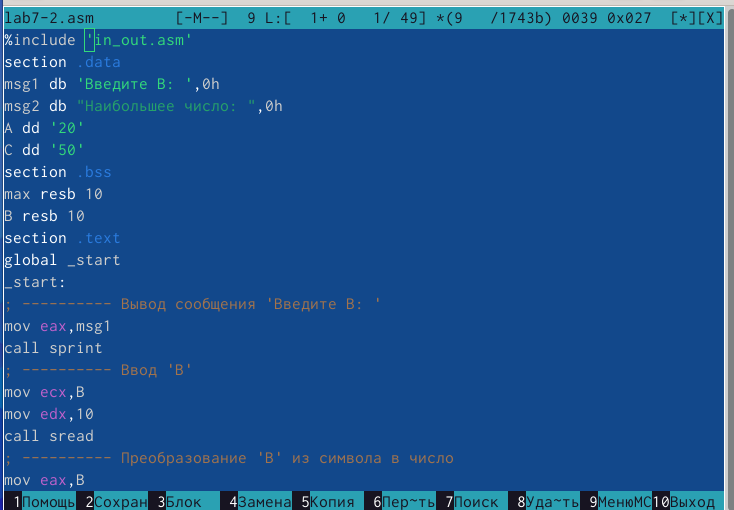


Рис. 9: Текст программы в файле

Создаю исполняемый файл и запускаю его (рис. 10).

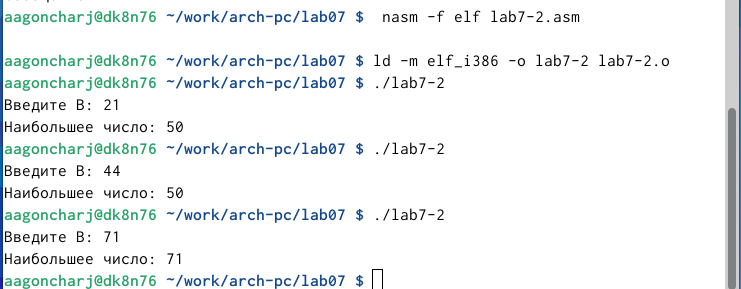


Рис. 10: Программа для сравнения чисел

## 4.2 Изучение структуры файлы листинга

Создаю файл листинга для программы из файла lab7-2.asm с помощью ключа -l (рис. 11).

Рис. 11: Файл листинга lab8-2.lst

Рис. 11: Файл листинга lab8-2.lst

Далее открываю файл lab7-2.lst для ознокомления с его форматом и содержимым (рис. 12).

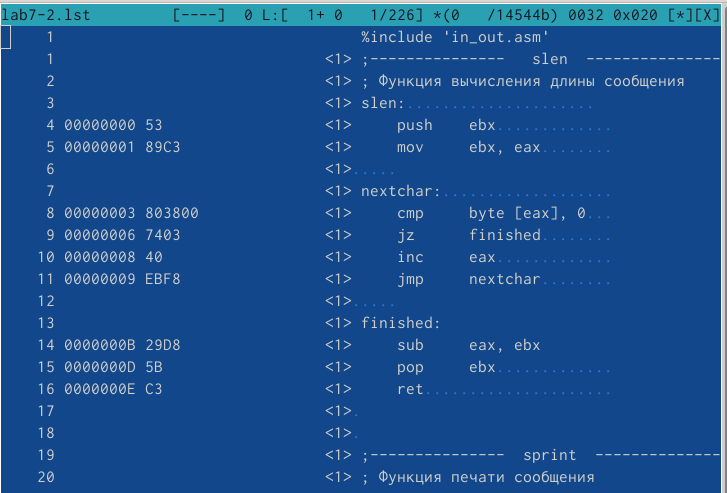


Рис. 12: Объяснения третьей строки

Теперь в строке mov eax,max я убраю max в файле lab7-2.asm (рис. 13).

Рис. 13: Создание файла без одного операнда

Рис. 13: Создание файла без одного операнда

Пробую запустить файл, но программы выдает ошибку, так как для программы нужно два операнда (рис. 14).

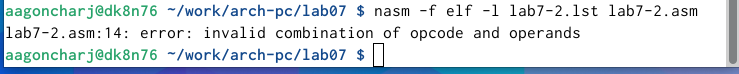


Рис. 14: Запуск файла

В файле листинга тоже отображается ошибка (рис. 15).

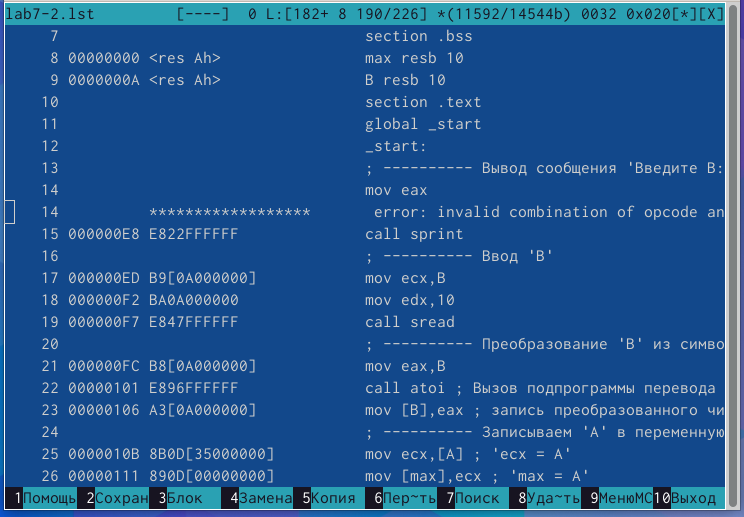


Рис. 15: Файл листинга без одного операнда

## 4.3 Задание для самостоятельной работы

1.Сначала я создаю файл lab7-3.asm (рис. 16).

Рис. 16: Создание файла

Рис. 16: Создание файла

Так как я меня 15 вариант, я написала программу для нахождения наименьшего из трех чисел для 32, 6, 54 (рис. 17).

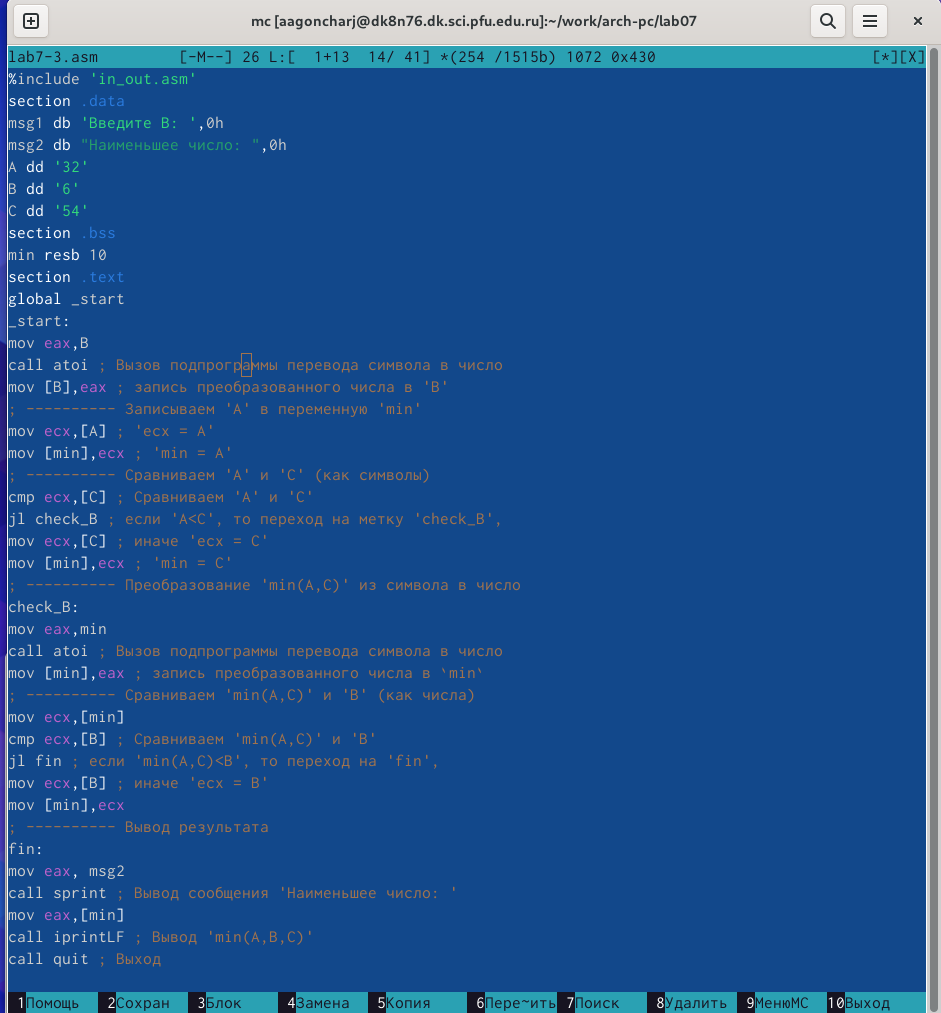


Рис. 17: Текст программы в файле

Теперь создаю исполняемый файл и запускаю его. Программа вывела меньшее из этих чисел (рис. 18).

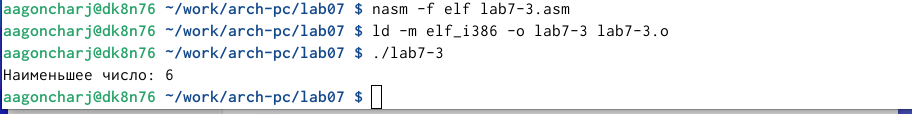


Рис. 18: Результат работы программы

2.Сначала я создаю файл lab7-4.asm (рис. 19).

Рис. 19: Создание файла

Рис. 19: Создание файла

Теперь мне необходимо написать программу, для вычисления значения функции при введенных Х и А (рис. 20).

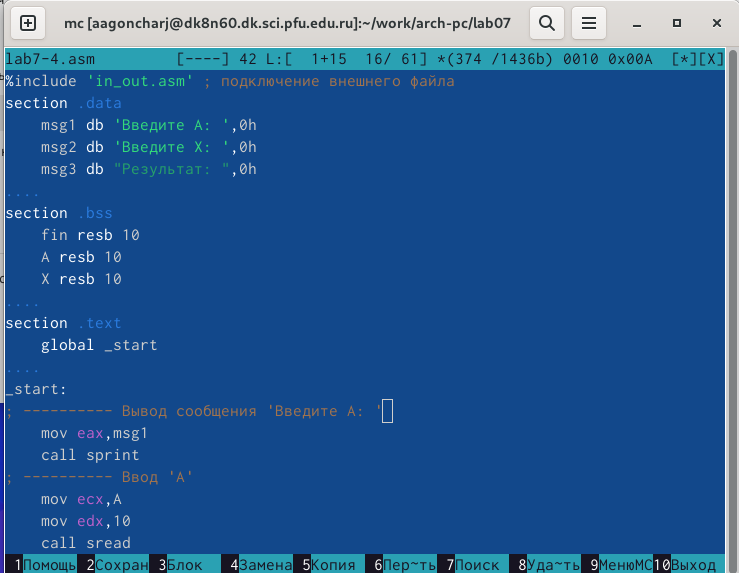


Рис. 20: Текст программы в файле

Теперь создаю исполняемый файл и запускаю его. Программа работает верно (рис. 21).

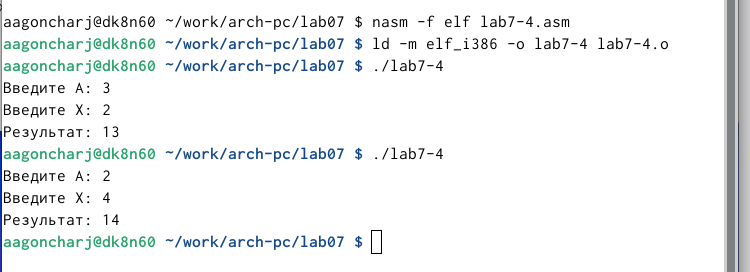


Рис. 21: Результат работы программы

## 4.4 Листинг№1 для нахождения наименьшего из 3 чисел

%include 'in\_out.asm'  
section .data  
msg1 db 'Введите B: ',0h  
msg2 db "Наименьшее число: ",0h  
A dd '32'  
B dd '6'  
C dd '54'  
section .bss  
min resb 10  
section .text  
global \_start  
\_start:  
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'  
jl 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'  
jl 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

## 4.5 Листинг№2 для вычисления значения функции

%include 'in\_out.asm' ; подключение внешнего файла  
section .data  
 msg1 db 'Введите A: ',0h  
 msg2 db 'Введите X: ',0h  
 msg3 db "Результат: ",0h  
   
section .bss  
 fin 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  
  
; ---------- Вывод сообщения 'Введите X: '  
 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  
  
  
 mov ecx,[X]  
 mov ebx,[A]  
 cmp ecx,ebx  
 jge func2  
 mov edx,[A]  
 add edx,10  
 mov [fin],edx  
 jmp final  
   
func2:  
 mov ax,[X]  
 add ax,10  
 mov [fin],ax  
 jmp final  
  
final:  
 mov eax,msg3  
 call sprint ; Вывод сообщения 'Результат: '  
 mov eax,[fin]  
 call iprintLF ; Вывод fin  
 call quit ; Выход

# 5 Выводы

При выполнении данной лабораторной работы я изучила команды условного и безусловного переходов, приобрела навыки написания программ с использованием переходов и ознакомилась с назначением и структурой файла листинга.

# Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.

2. Robbins A. Bash Pocket Reference. O’Reilly Media, 2016. 156 с.

3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.

4. Newham C. [Learning the bash Shell: Unix Shell Programming](http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658). O’Reilly Media, 2005. 354 с.