Лабораторная работа №7

Команды безусловного и условного переходов в Nasm. Программирование ветвлений

Глобин Никита Анатольевич

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

Реализация переходов в NASM  
Изучение структуры файлы листинга  
Задание для самостоятельной работы

# 3 Выполнение лабораторной работы

## 3.1 Реализация переходов в NASM

1. Переходим в католог lab07 и создаём файл lab7-1.asm (рис. 1).

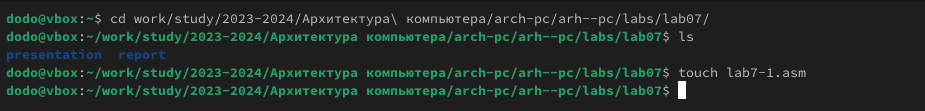


Рис. 1: photo 1

1. переписываем программу из ТУИС в наш файл (рис. 2).

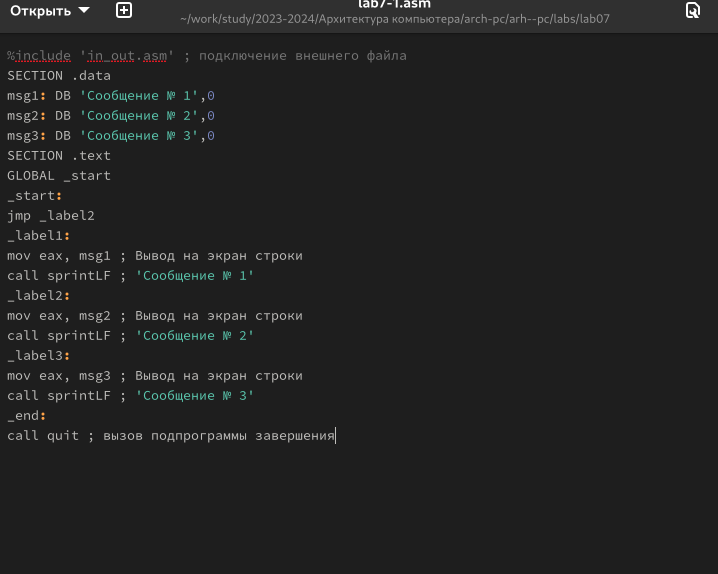


Рис. 2: photo 2

1. компилируем и запускаем этот файл (рис. 3).

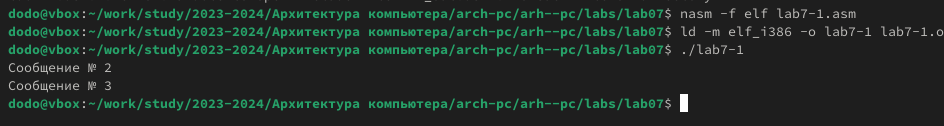


Рис. 3: photo 3

1. теперь переписываем программу как показано в лабораторной работе (рис. 4).

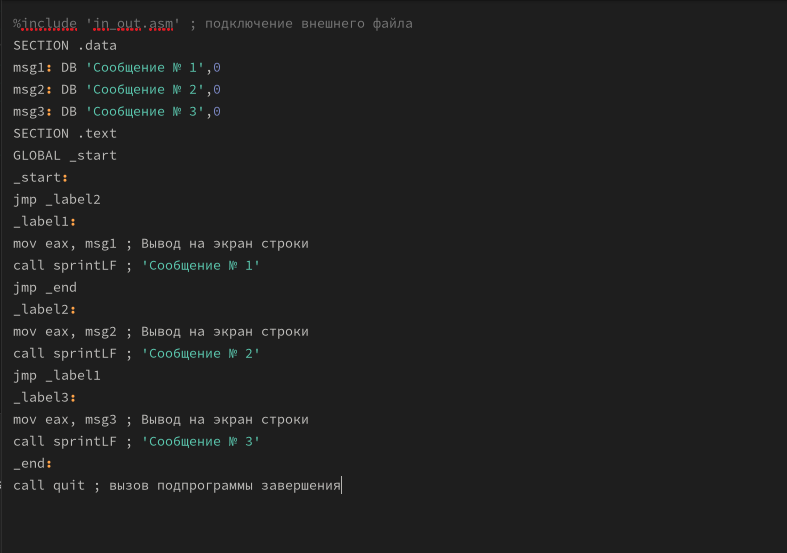


Рис. 4: photo 4

1. компилируем и запускаем этот файл (рис. 5).

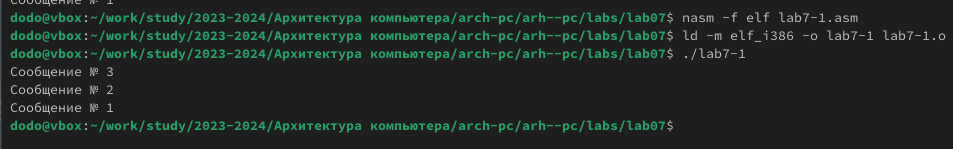


Рис. 5: photo 5

1. создаём файл lab7-2.asm и пишем в нём код новой программы(рис. 6).



Рис. 6: photo 6

1. компилируем и запускаем этот файл (рис. 7).



Рис. 7: photo 7

## 3.2 Изучение структуры файлы листинга

1. создаём файл листинга через команду nasm -f elf -l lab7-2.lst lab7-2.asm (рис. 8).



Рис. 8: photo 8

1. открываем этот файл командой mcedit lab7-2.lst (рис. 9).

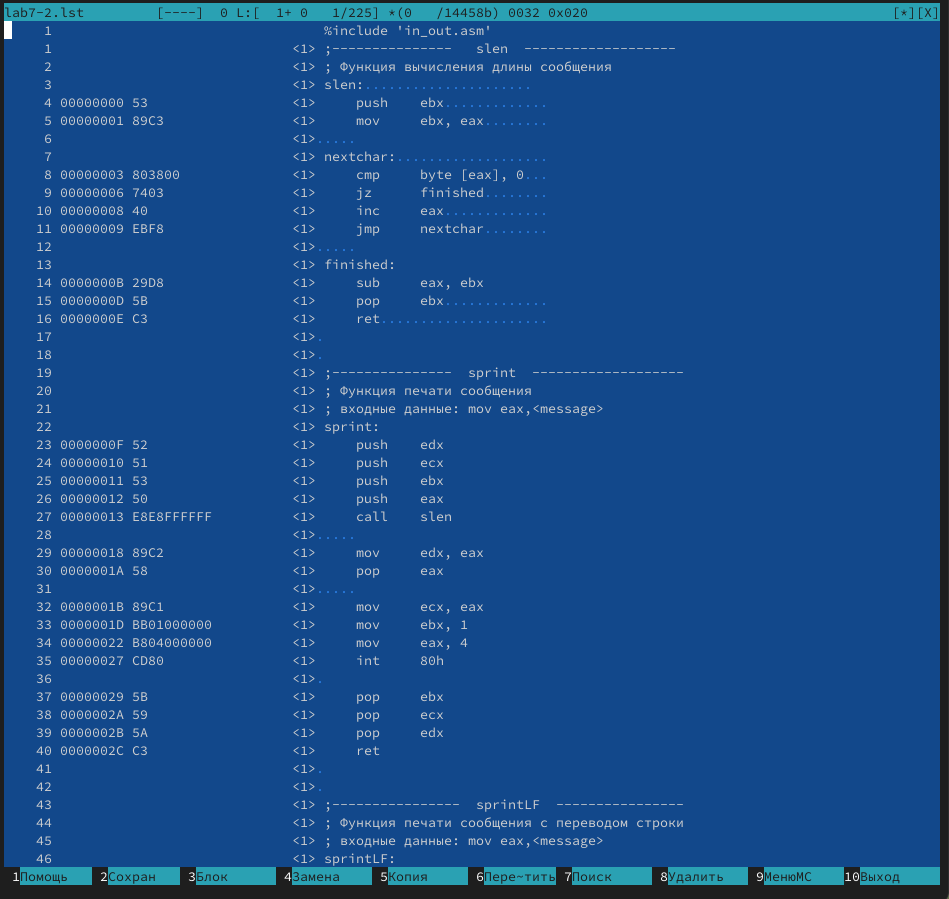


Рис. 9: photo 9

тут в строке: (рис. **¿fig:0010?**). 1. Строка 51: push eax Содержимое регистра eax помещается в стек. Это значение необходимо для последующей операции вывода. 2. Строка 53: call sprint Вызывается функция sprint. эта функция, отвечает за вывод строки на экран. Аргументы для этой функции были предварительно подготовлены и помещены в стек. 3. Строка 55: pop eax Значение, которое было ранее помещено в стек на строке 51, извлекается и помещается обратно в регистр eax. Это делается для того, чтобы восстановить исходное состояние регистра после выполнения функции sprint.

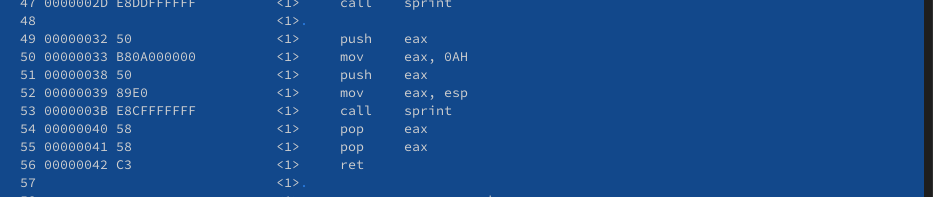


Рис. 10: photo 10

1. открываем lab7-2.asm и попробуем намеренно допустить ошибку в нашем коде, убрав у команды move 1 операнд (рис. **¿fig:0011?**).

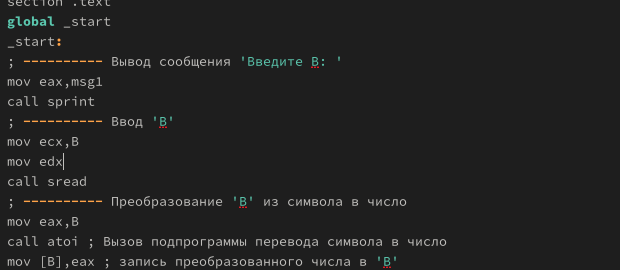


Рис. 11: photo 11

1. компилируем и запускаем этот файл (рис. 12).



Рис. 12: photo 12

1. открываем листинговый файл и видим что там тоже сообщается об ошибке (рис. 13).

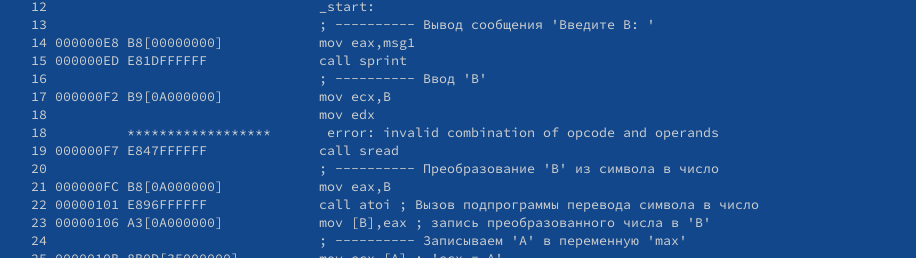


Рис. 13: photo 13

## 3.3 Задание для самостоятельной работы

1. Создадим файл для выполнения самостоятельной работы. Мой вариант - 9 (рис. 14).

photo 14

Рис. 14: photo 14

Напишем код для выполнения задания. Код выглядит так (рис. 15).



Рис. 15: photo 15

1. Соберём, запустим его и посмотрим на результат (рис. 16).

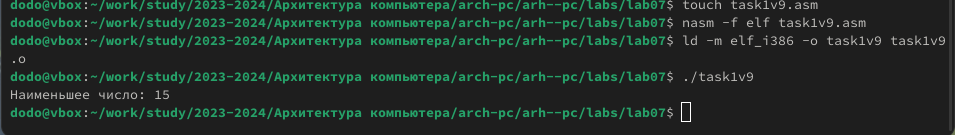


Рис. 16: photo 16

1. Теперь создадим второй файл самостоятельной работы для второго задания (рис. 17).

photo 17

Рис. 17: photo 17

1. Код будет выглядеть так (рис. 18).



Рис. 18: photo 18

1. Соберём исполняемый файл и запустим его (рис. 19).



Рис. 19: photo 19

# 4 Выводы

В результате работы над лабораторной работой были написаны программы, которые используют команды условных и безусловных переходов, были получены навыки работы с этими командами, а также были созданы и успешно прочитаны листинги для некоторых из программ.

# Список литературы