Лабораторная работа №8

Команды безусловного и условного переходов в Nasm. Программирование ветвлений

Медникова Екатерина Михайловна

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 Теоретическое введение

Для реализации ветвлений в ассемблере используются так называемые команды передачи управления или команды перехода. Можно выделить 2 типа переходов:

• условный переход – выполнение или не выполнение перехода в определенную точку программы в зависимости от проверки условия.

• безусловный переход – выполнение передачи управления в определенную точку программы без каких-либо условий.

Безусловный переход выполняется инструкцией jmp (от англ. jump – прыжок), которая включает в себя адрес перехода, куда следует передать управление:

jmp

Адрес перехода может быть либо меткой, либо адресом области памяти, в которую предварительно помещен указатель перехода. Кроме того, в качестве операнда можно использовать имя регистра, в таком случае переход будет осуществляться по адресу, хранящемуся в этом регистре.

Как отмечалось выше, для условного перехода необходима проверка какого-либо условия. В ассемблере команды условного перехода вычисляют условие перехода анализируя флаги из регистра флагов.

Флаг – это бит, принимающий значение 1 («флаг установлен»), если выполнено некоторое условие, и значение 0 («флаг сброшен») в противном случае. Флаги работают независимо друг от друга, и лишь для удобства они помещены в единый регистр — регистр флагов, отражающий текущее состояние процессора.

# 3 Выполнение лабораторной работы

1. Создала каталог для программам лабораторной работы No 8, перешла в него и создала файл lab8-1.asm:

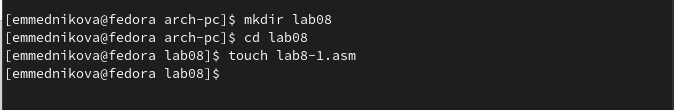


Figure 1: Создание каталога и файла

1. Ввела в файл lab8-1.asm текст программы из листинга 8.1.

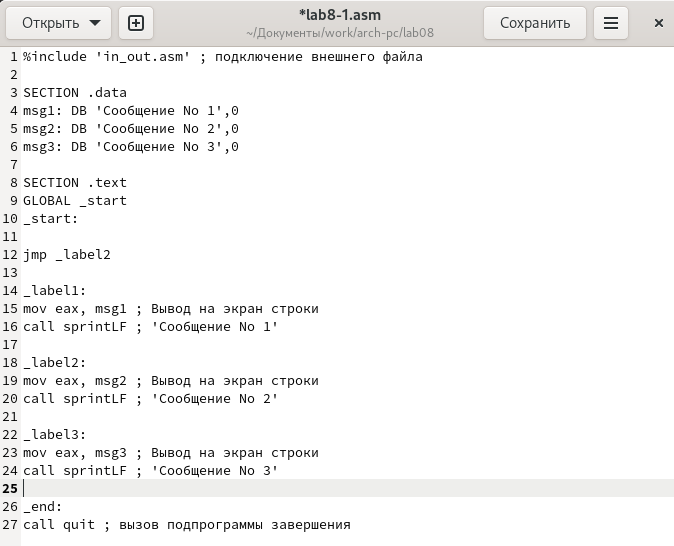


Figure 2: Ввод текста из листинга 8.1

Создала исполняемый файл и запустила его.

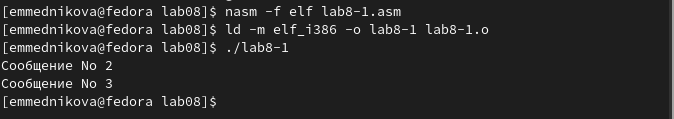


Figure 3: Создание и запуск файла

Изменила текст программы в соответствии с листингом 8.2.

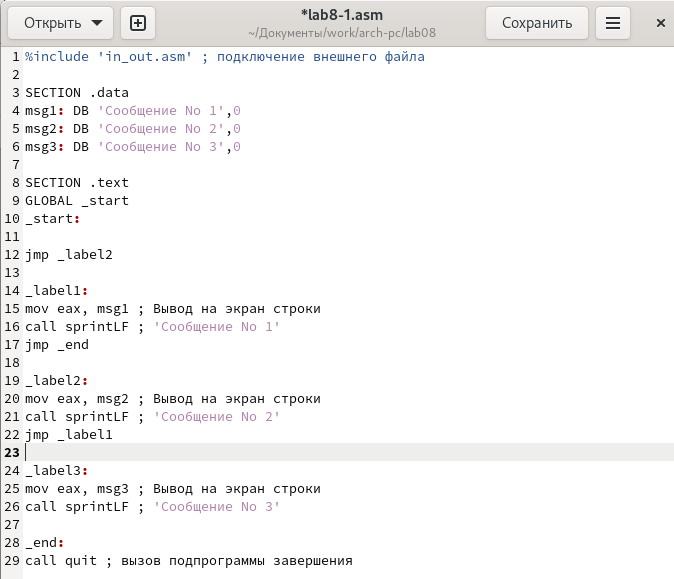


Figure 4: Изменение текста программы

Создала исполняемый файл и проверила его работу.

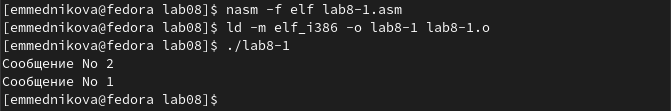


Figure 5: Создание и проверка работы файла

Изменила текст программы так, чтобы её вывод соответствовал инструкции, прописанной в лабораторной работе.

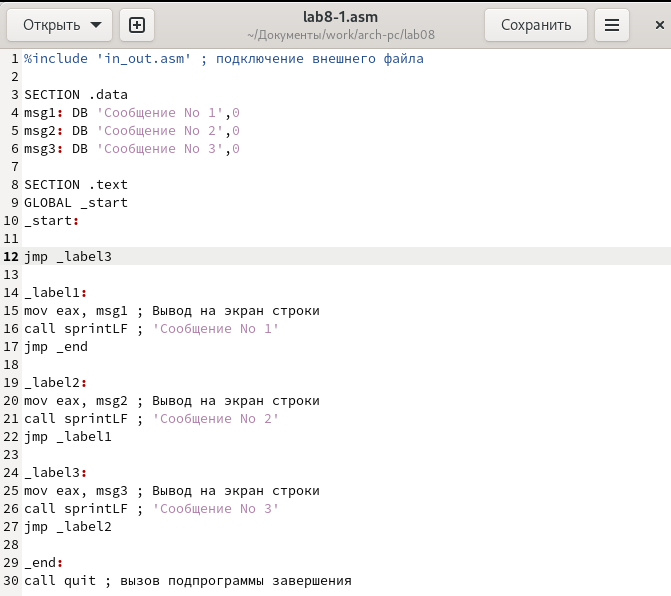


Figure 6: Изменение текста программы

Создала исполняемый файл и проверила его работу.

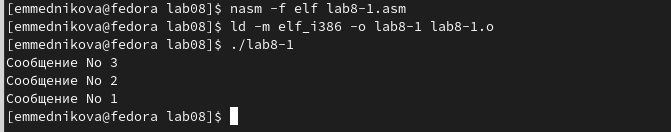


Figure 7: Создание и проверка работы файла

1. Создала файл lab8-2.asm в каталоге ~/work/arch-pc/lab08.

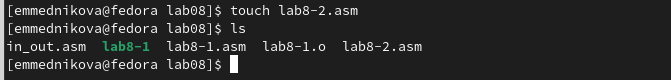


Figure 8: Создание файла lab8-2.asm

Ввела в созданный файл текст программы из листинга 8.3.

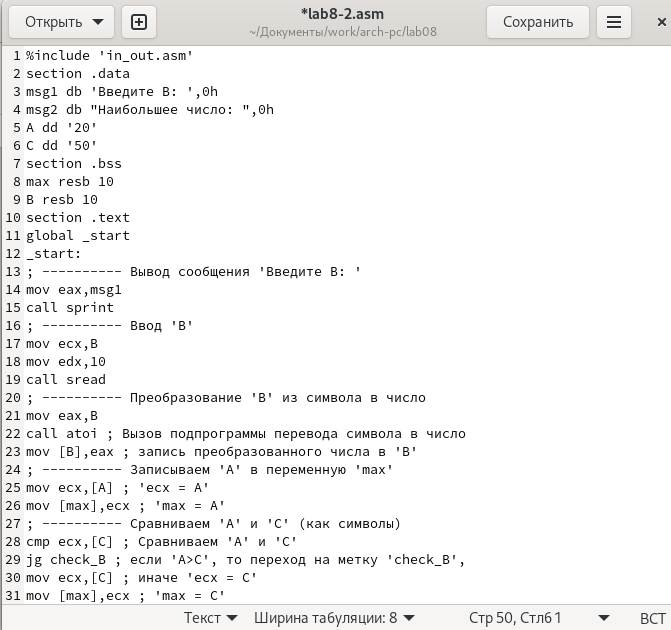


Figure 9: Ввод программы из листинга 8.3

Создала исполняемый файл и проверила его работу.

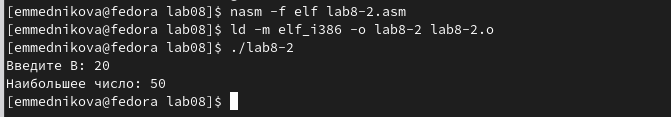


Figure 10: Создание исполняемого файла и проверка его работы

1. Создала файл листинга для программы из файла lab8-2.asm. Открыла файл листинга lab8-2.lst с помощью текстового редактора mcedit.

Figure 11: Создание файла листинга

Figure 11: Создание файла листинга

27 строка. Адрес 00000012. Машинный код 50. Push eax (исходный текст программы) выделяет место наверху стека и помещает туда значение из регистра eax.

Figure 12: push eax

Figure 12: push eax

55 строка. Адрес 00000040. Машинный код 58. Pop eax (исходный текст программы) переносит любые данные из верхней части стека в eax и освобождает эту область памяти.

Figure 13: pop eax

Figure 13: pop eax

95 строка. Адрес 00000073. Машинный код 49. Dec eсx (исходный текст программы) уменьшает значение ecx на единицу.

Figure 14: dec ecx

Figure 14: dec ecx

Открыла файл с программой lab8-2.asm и удалила один операнд - max.

Figure 15: Удаление max

Figure 15: Удаление max

Выполнила трансляцию с получением файла листинга. Выдалась ошибка.

Figure 16: Выполнение трансляции и получение ошибки

Figure 16: Выполнение трансляции и получение ошибки

# 4 Самостоятельная работа

1. Создала файлы для самостоятельной работы.

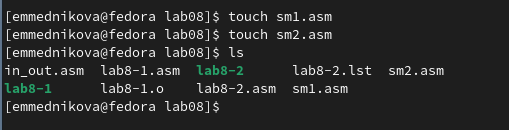


Figure 17: Файлы

Написала программу для нахождения наименьшей из 3 целочисленных переменных a, b и c. Значения перемнных взяла из таблицы 8.5 в соответствии с вариантом. У меня вариант 10. Значения: 41, 62, 35.

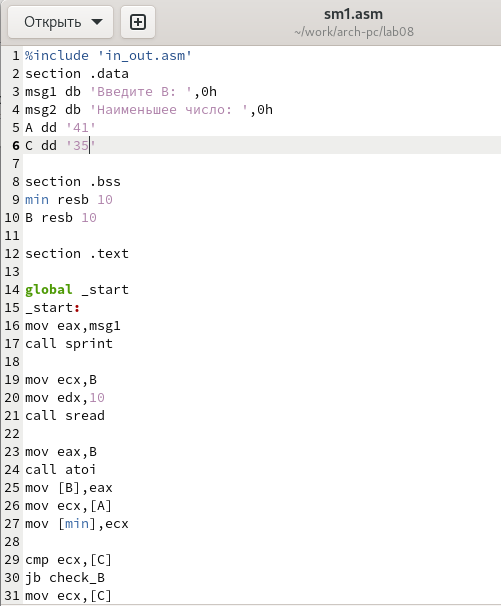


Figure 18: Программа

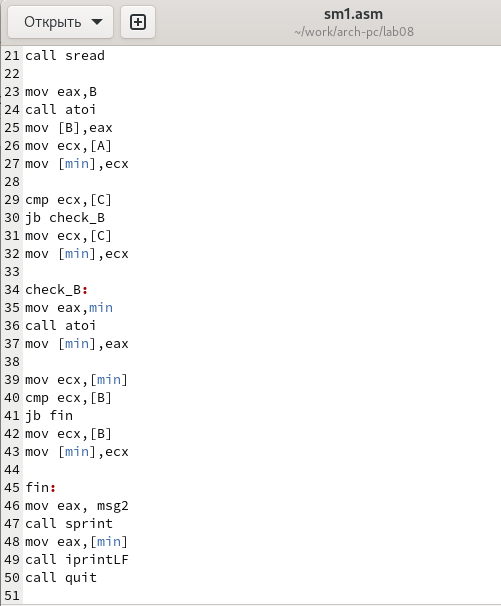


Figure 19: Программа

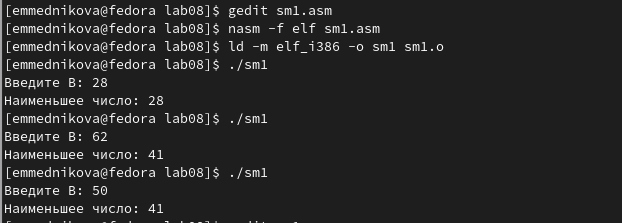


Figure 20: Результат

Программа работает не совсем корректно, выдаёт неправильное число. Однако если ввести другое число b, которое меньше заданных, то работает верно.

1. Написала программу для вычисления функции, представленная в таблице 8.6 в соответствии с вариантом. У меня вариант 10. Значения (3;0) и (1;2)

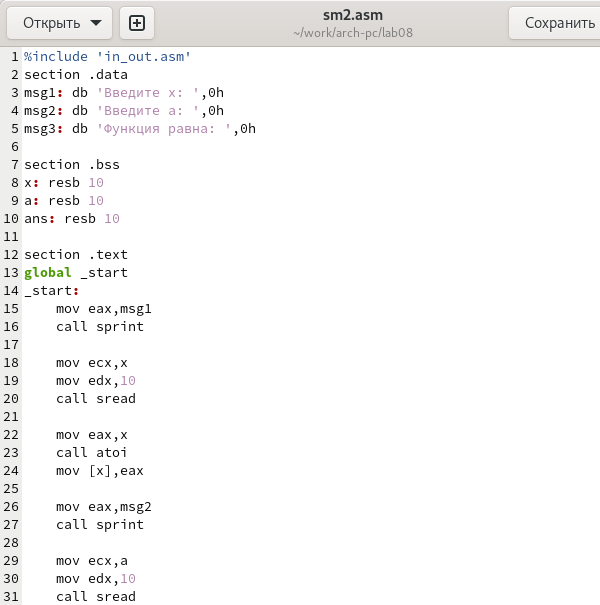


Figure 21: Программа

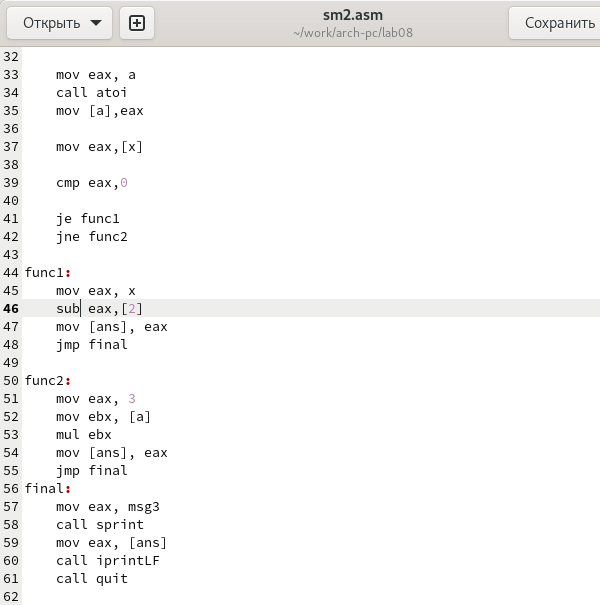


Figure 22: Программа

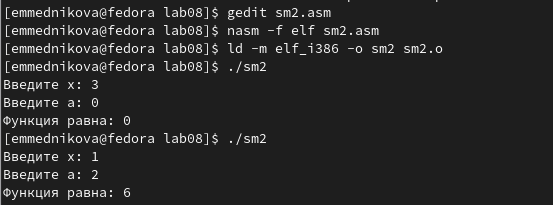


Figure 23: Результат

# 5 Выводы

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