Отчёт по лабораторной работе №7

Архитектура компьютера

Алехин Давид Андреевич

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

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

# 3 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 про- граммы без каких-либо условий.

# 4 Выполнение лабораторной работы

Создаю папку для лабораторной работы №7, в ней создаб файл lab7-1.asm (рис. 1). Переношу в эту папку файл in\_out.asm. (рис. 2).

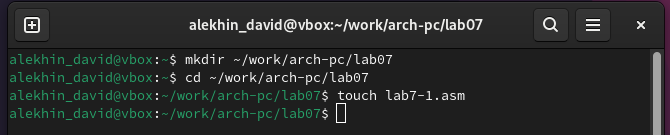


Рис. 1: Создание lab07/lab7-1.asm

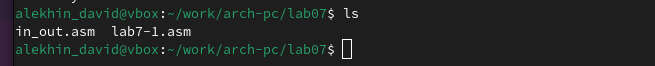


Рис. 2: in\_out.asm

Ввожу в файл lab7-1.asm текст программы 7.1. Программа с использованием инструкции jmp. (рис. 3).

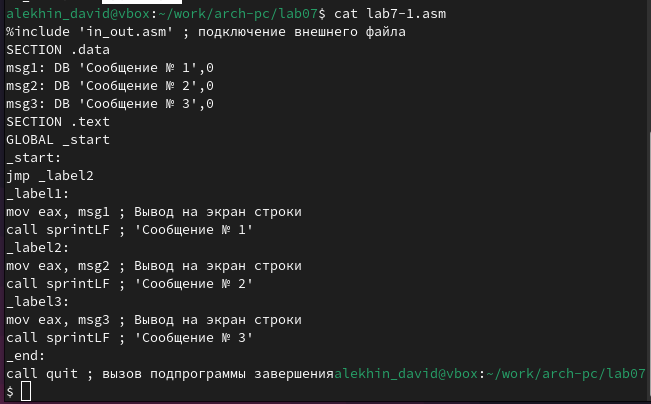


Рис. 3: Программа 7.1

Компеллирую файл и запускаю его.(рис. 4).

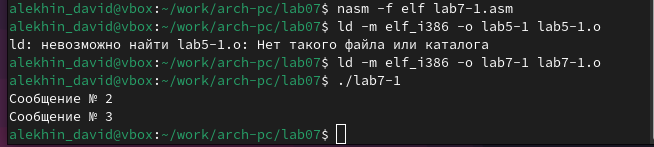


Рис. 4: ./lab7-1

Создаю файл lab7-1.2.asm с текстом команды 7.2. Программа с использованием инструкции jmp. (рис. 5). Компеллирую и запускаю его. (рис. 6).

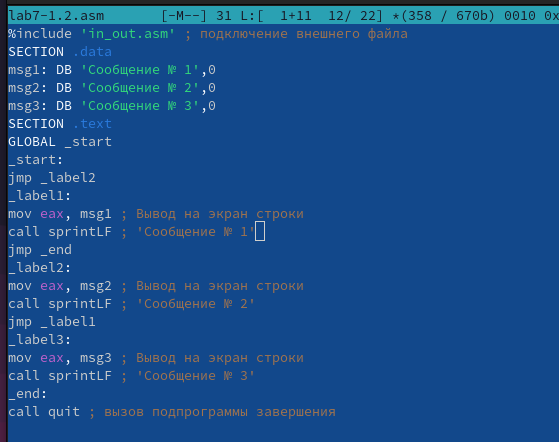


Рис. 5: lab7-1.2.asm

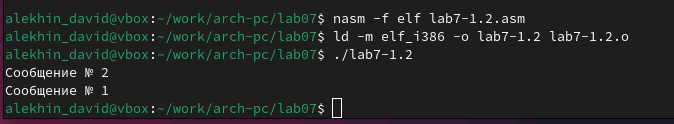


Рис. 6: Запуск lab7-1.2.asm

Корректирую текст клманды чтобы она выводила Сообщение №1, Сообщение №2, Сообщение №3 в обратном порядке. (рис. 7).Компеллирую и запускаю исправленый файл. (рис. 8).

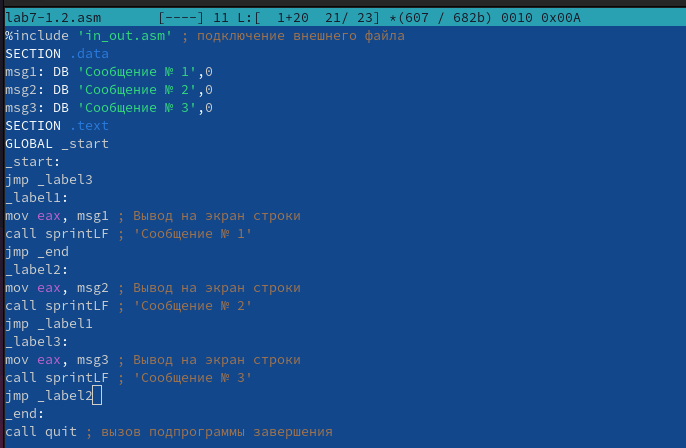


Рис. 7: Откорректрированный lab7-1.2.asm

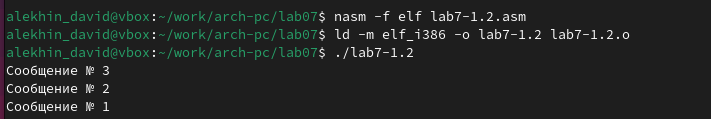


Рис. 8: Запуск откорректрированного lab7-1.2.asm

Слздаю файл lab7-2.asm с текстом команды 7.3. Программа, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: A,B и C. (рис. 9).

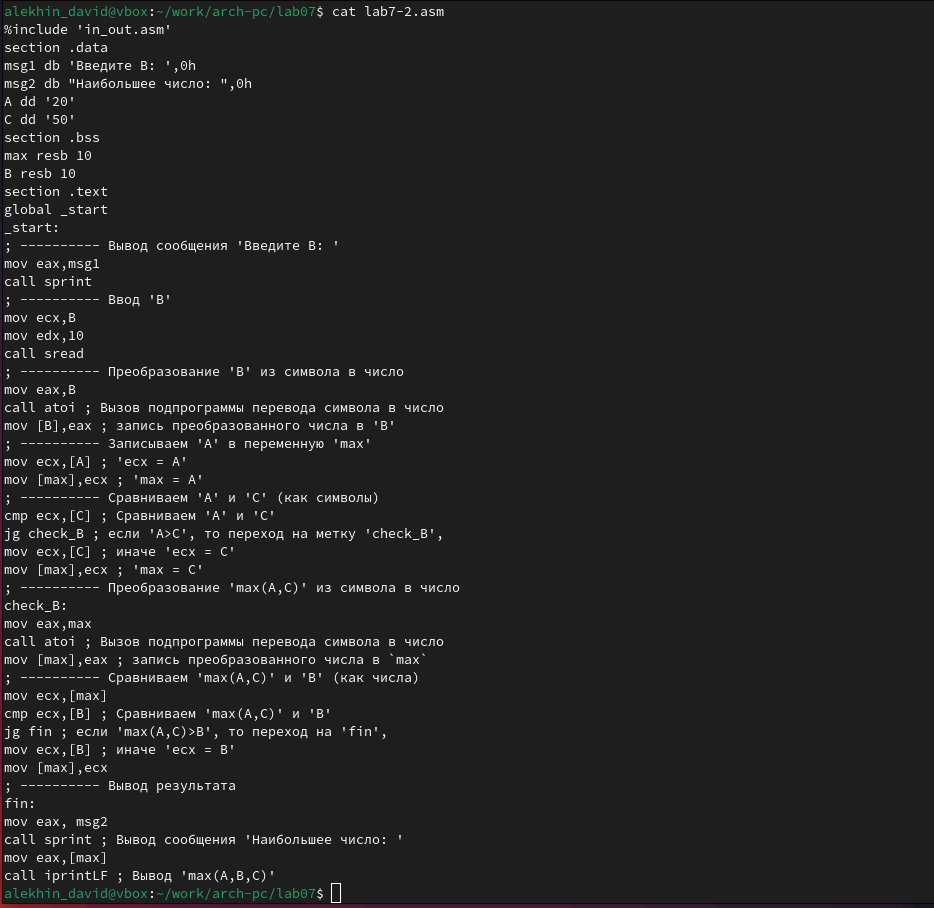


Рис. 9: lab7-2.asm

Компеллирую и запускаю lab7-2.asm. (рис. 10).

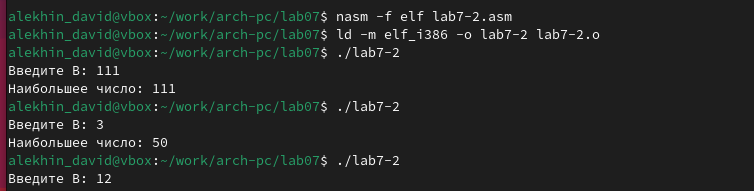


Рис. 10: Запуск lab7-2.asm

Создаю и открываю листинг lab7-2.asm. (рис. 11).

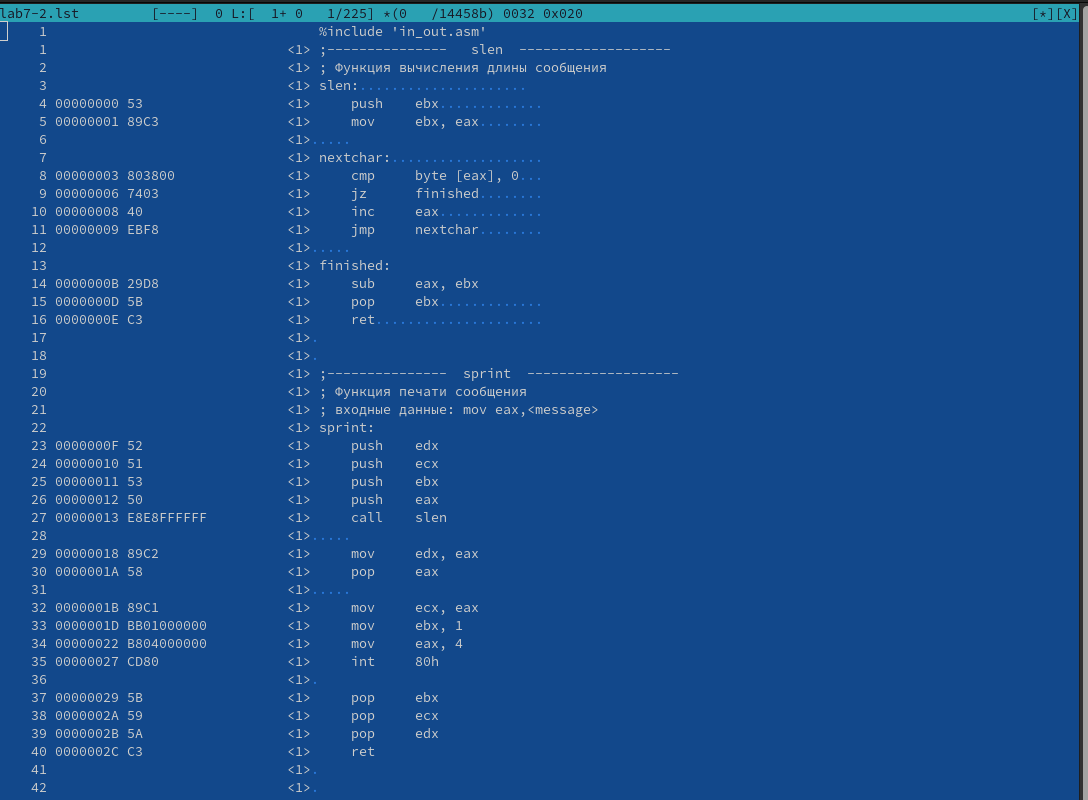


Рис. 11: lab7-2.lst

Расшифровка 3х строк листинга: 1)Строка 5: Эта строка находится на 5 месте, ее адрес “00000035”, Машинный код - [32300000], A dd ‘20’ - исходный текст программы, присваивающий переменной А значение 20. 2)Строка 6: Эта строка находится на 6 месте, ее адрес “00000039”, Машинный код - [35300000], A dd ‘50’ - исходный текст программы, присваивающий переменной B значение 50. 3)Строка 8 : Эта строка находится на 8 месте, ее адрес “00000000”, Машинный код - res Ah, max resb 10 - исходный текст программы, означающий, что максимальное резервирование может быть 10 байт. (рис. 12).

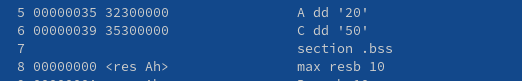


Рис. 12: Расшифровка 3х строк листинга

У даляю в строке mov eax, max операнду max.(рис. 13).

Рис. 13: mov eax

Рис. 13: mov eax

В результате получаю ошибку при создании asm файла и lst файла (рис. 14)., а в фай с листингом получаю ошибку в 34 строке.(рис. 15).

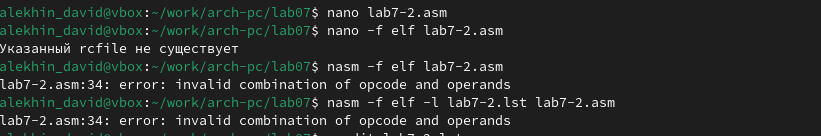


Рис. 14: Ошибка при создании

Рис. 15: Ошибка в листинге

Рис. 15: Ошибка в листинге

Приступаю к самостоятельной работе (1 вариант). Создаю файл lab7-1s.asm и пишу код для первого задания. (рис. 16).

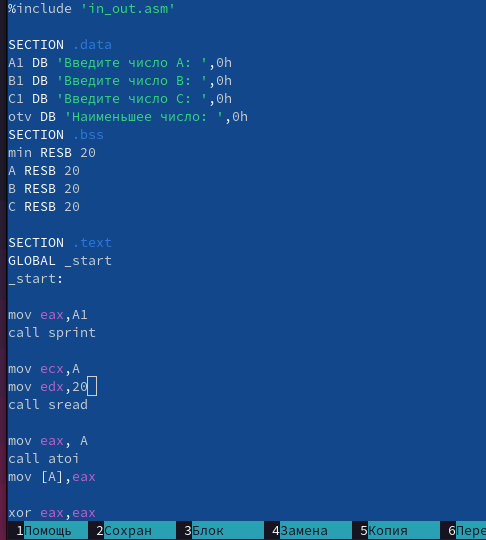


Рис. 16: Код 1 задания

Компеллирую и запускаю lab7-1s.asm, проверяю работоспособность.(рис. 17).

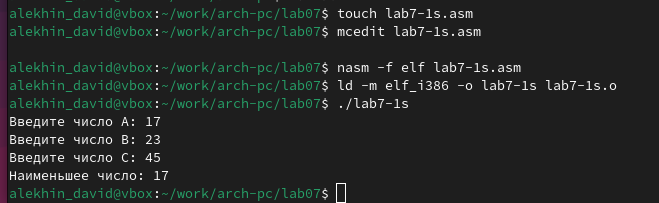


Рис. 17: ./lab7-1s.asm

Создаю файл lab7-2s.asm и пишу код для второго задания. (рис. 18).

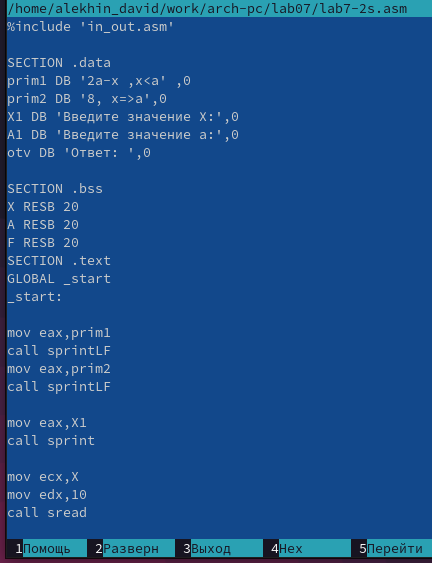


Рис. 18: Код 2 задания

Компеллирую и запускаю lab7-2s.asm, проверяю работоспособность. (рис. 19).

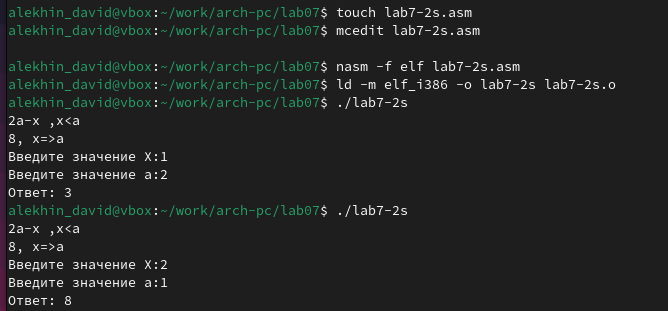


Рис. 19: ./lab7-2s.asm

# 5 Выводы

Сделав лабораторную работу я выполнил её главную цель: “Изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.”

# Список литературы

1. GDB: The GNU Project Debugger. — URL: https://www.gnu.org/software/gdb/.
2. GNU Bash Manual. — 2016. — URL: https://www.gnu.org/software/bash/manual/.
3. Midnight Commander Development Center. — 2021. — URL: https://midnight-commander. org/.
4. NASM Assembly Language Tutorials. — 2021. — URL: https://asmtutor.com/.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O’Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: http://www.amazon.com/Learning- bash-Shell-Programming-Nutshell/dp/0596009658.
6. Robbins A. Bash Pocket Reference. — O’Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: https://www.nasm.us/docs.php.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс,
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: https://www.opennet.ru/docs/RUS/nasm/.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВ- Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm\_unix.
15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер,
17. — 1120 с. — (Классика Computer Science).