Отчёт по лабораторной работе 7

Архитектура компьютера

Волчкова Елизавета Дмитриевна

Содержание

# Цель работы

Изучить команды условного и безусловного переходов и приобрести навыки написания программ с использованием переходов, а также знакомство с назначением и структурой файла листинга.

# Задание 1. Напишите программу нахождения наименьшей из 3 целочисленных переменных a, b и с. Значения переменных выбрать из табл. 7.5 в соответствии с вариантом, полученным при выполнении лабораторной работы №7. Создайте исполняемый файл и проверьте его работу.

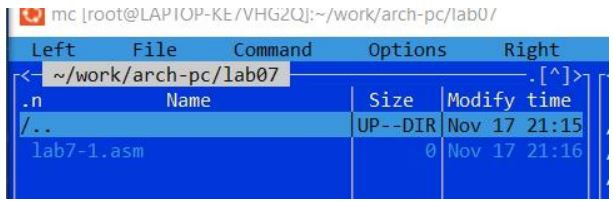
# Задание 2. Напишите программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции f(x) и выводит результат вычислений. Вид функции f(x) выбрать из таблицы 7.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу для значений x и a из 7.6.

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

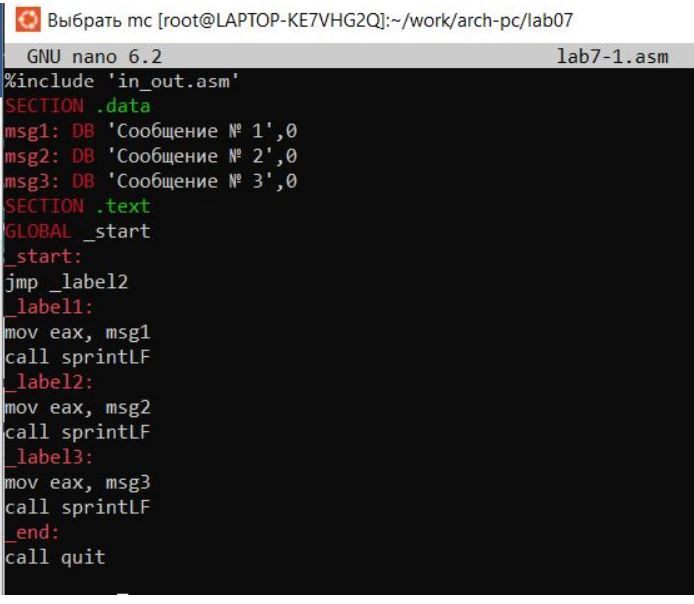
• условный переход – выполнение или не выполнение перехода в определенную точку программы в зависимости от проверки условия.

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

# Выполнение лабораторной работы. Сачала создала каталог для программам лабораторной работы №7, перешла в него и создала файл lab7-1.asm: mkdir ~/work/arch-pc/lab07 cd ~/work/arch-pc/lab07 touch lab7-1.asm



Ввела в файл lab7-1.asm текст программы из листинга 7.1.



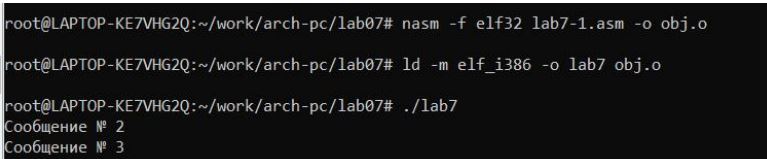
*2lab07*

Создала исполняемый файл и запустила его. Результат работы данной программы будет следующим: root@LAPTOP-KE7VHG2Q:~# ./lab7

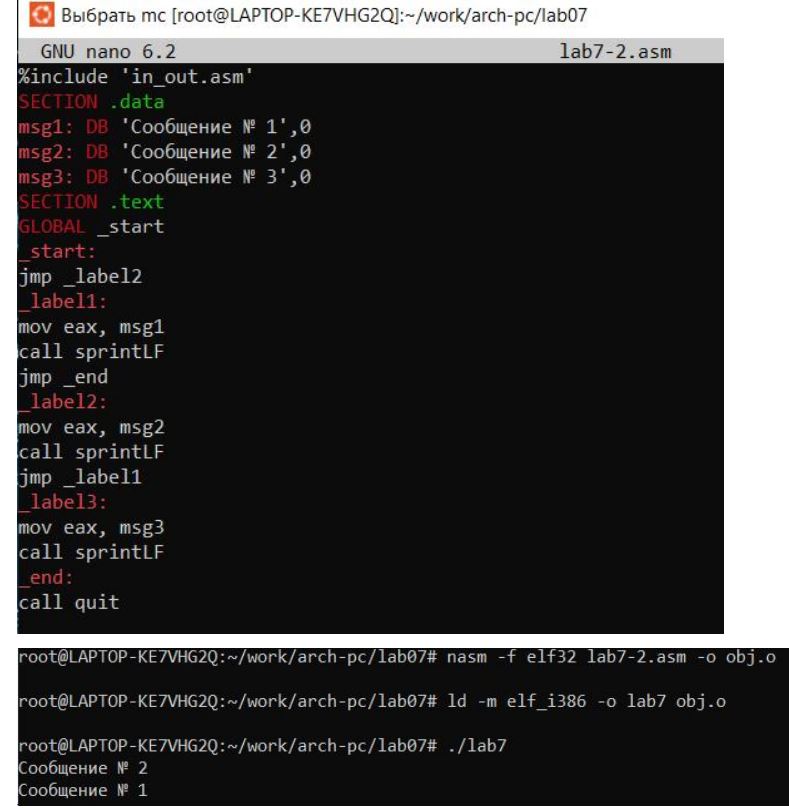
Сообщение № 2

Сообщение № 3

root@LAPTOP-KE7VHG2Q:~#



Далее в текст программы после вывода сообщения №2 добавила инструкцию jmp с меткой \_label1 и после вывода сообщения №1 добавил аинструкцию jmp с меткой \_end.



Затем создала исполняемый файл и проверила его работу. Потом изменила текст программы добавив или изменив инструкции jmp, чтобы вывод программы был следующим:

root@LAPTOP-KE7VHG2Q:~# ./lab7

Сообщение № 3

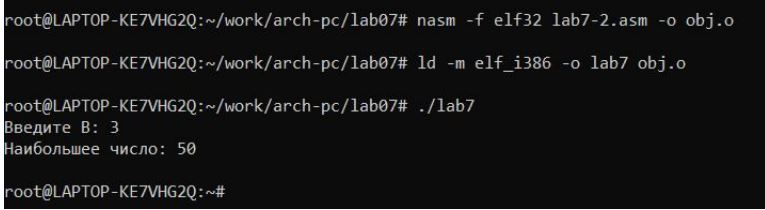
Сообщение № 2

Сообщение № 1

root@LAPTOP-KE7VHG2Q:~#



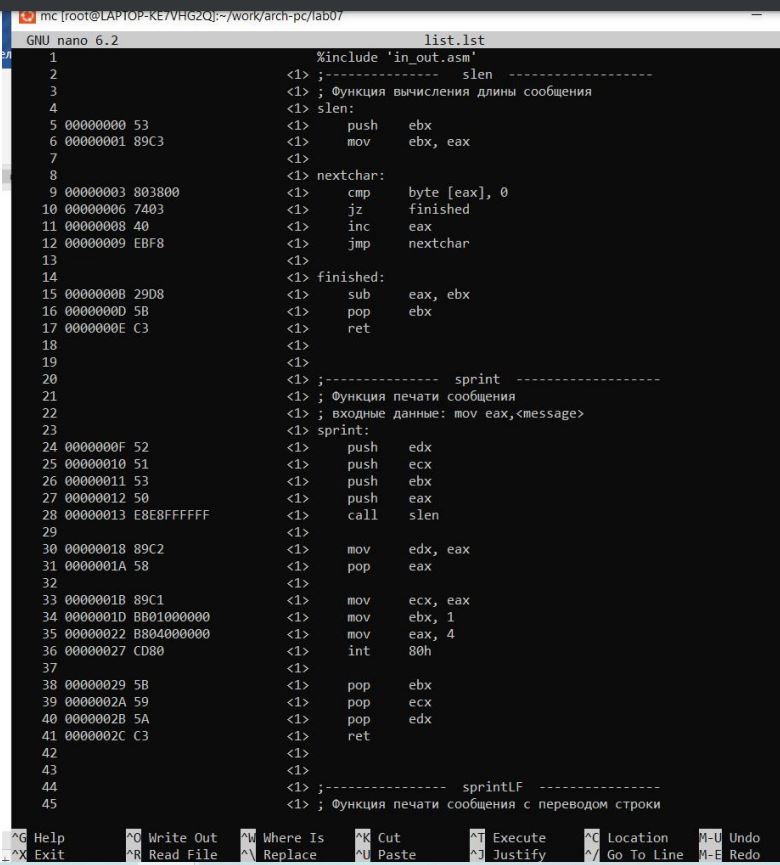
После создала файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. Я внимательно изучила текст программы из листинга 7.3 и ввела его в lab7-2.asm. Далее я создала исполняемый файл и проверила его работу для разных значений B.



Потом создала файл листинга для программы из файла lab7-2.asm

nasm -f elf -l list.lst lab7-2.asm

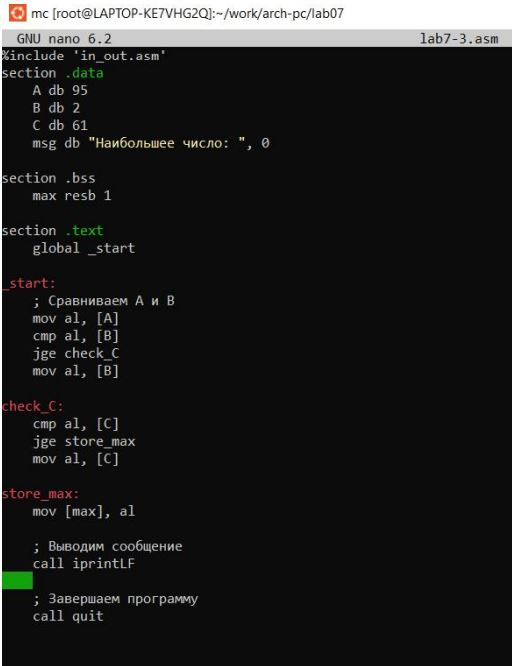
Открыла файл листинга list.lst с помощью текстового редактора nano: nano list.lst

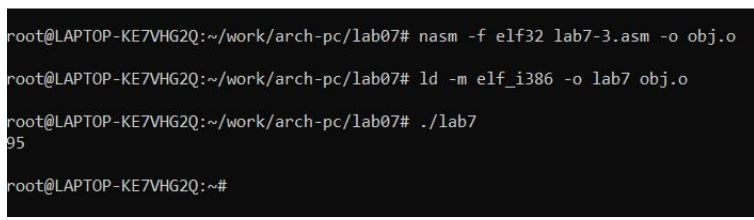


Я открыла файл с программой lab7-2.asm и в любой инструкции с двумя операндами удалить один операнд. Выполнила трансляцию с получением файла листинга: nasm -f elf -l list.lst lab7-2.asm-o obj.o.

Какие выходные файлы создаются в этом случае? list.lst obj.o

Что добавляется в листинге? Файл листинга позволяет увидеть работу компилятора FASM: что генерирует каждая строка исходного кода, сколько байт занимают машинные команды, какие значения присваиваются переменным. Листинг состоит из трех колонок: первая содержит адреса (точнее, смещения в секции), вторая —сгенерированный машинный код и третья — соответствующие строчки исходного кода программы.

Написала программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции f(x) и выводит результат вычислений. Вид функции f(x) выбрала из таблицы7.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создала исполняемый файл ипроверилаегоработу для значений x и a из 7.6. 

После создала файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 я внимательно изучила текст программы из листинга 7.3 и ввела его в lab7-2.asm. Далее я создала исполняемый файл и проверила его работу для разныхзначений B. 

# Выводы

Целью было изучить команды условного и безусловного переходов и приобрести навыки написания программ с использованием переходов, а также знакомство с назначением и структурой файла листинга. Сделав задания я смогла разобраться в данной теме.

# 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/Learningbash-Shell-Programming-Nutshell/dp/0596009658.6. Robbins A. Bash Pocket Reference. —O’Reilly Media, 2016. —156с.— ISBN 978-1491941591. 7. The NASM documentation. — 2021. —URL: https://www.nasm.us/docs.php. 8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. —502с. —ISBN9781784396879. 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, 2017. 11. Новожилов О. П. Архитектура ЭВМи систем. —М. : Юрайт, 2016.12. Расширенный ассемблер: NASM. —2021. —URL: https://www.opennet.ru/docs/RUS/nasm/. 13. Робачевский А., Немнюгин С., Стесик О. ОперационнаясистемаUNIX. — 2-е изд. — БХВПетербург, 2010. —656 с. —ISBN978-5-94157-538-1.14. Столяров А. Программирование на языке ассемблера NASMдляОСUnix. — 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-еизд. — СПб. : Питер, 2015. — 1120 с. — (Классика