Отчёт по лабораторной работе №8

Дисциплина: архитектура компьютера

Назармамадов Умед Джамшедович

Содержание

# 1 Цель работы

Приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки.

# 2 Задание

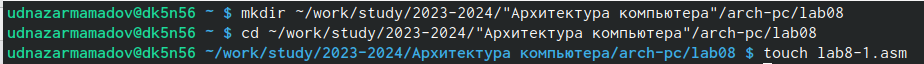
1. Реализация циклов в NASM.
2. Обработка аргументов командной строки.
3. Вопросы для самопроверки.

# 3 Теоретическое введение

Стек — это структура данных, организованная по принципу LIFO («Last In — First Out» или «последним пришёл — первым ушёл»). Стек является частью архитектуры процессора и реализован на аппаратном уровне. Для работы со стеком в процессоре есть специальные регистры (ss, bp, sp) и команды. Основной функцией стека является функция сохранения адресов возврата и передачи аргументов при вызове процедур. Кроме того, в нём выделяется память для локальных переменных и могут временно храниться значения регистров. Стек имеет вершину, адрес последнего добавленного элемента, который хранится в ре- гистре esp (указатель стека). Противоположный конец стека называется дном. Значение, помещённое в стек последним, извлекается первым. При помещении значения в стек указа- тель стека уменьшается, а при извлечении — увеличивается. Для стека существует две основные операции: • добавление элемента в вершину стека (push); • извлечение элемента из вершины стека (pop). Команда push размещает значение в стеке, т.е. помещает значение в ячейку памяти, на которую указывает регистр esp, после этого значение регистра esp увеличивается на 4. Команда pop извлекает значение из стека, т.е. извлекает значение из ячейки памяти, на которую указывает регистр esp, после этого уменьшает значение регистра esp на 4. У этой команды также один операнд, который может быть регистром или переменной в памяти.

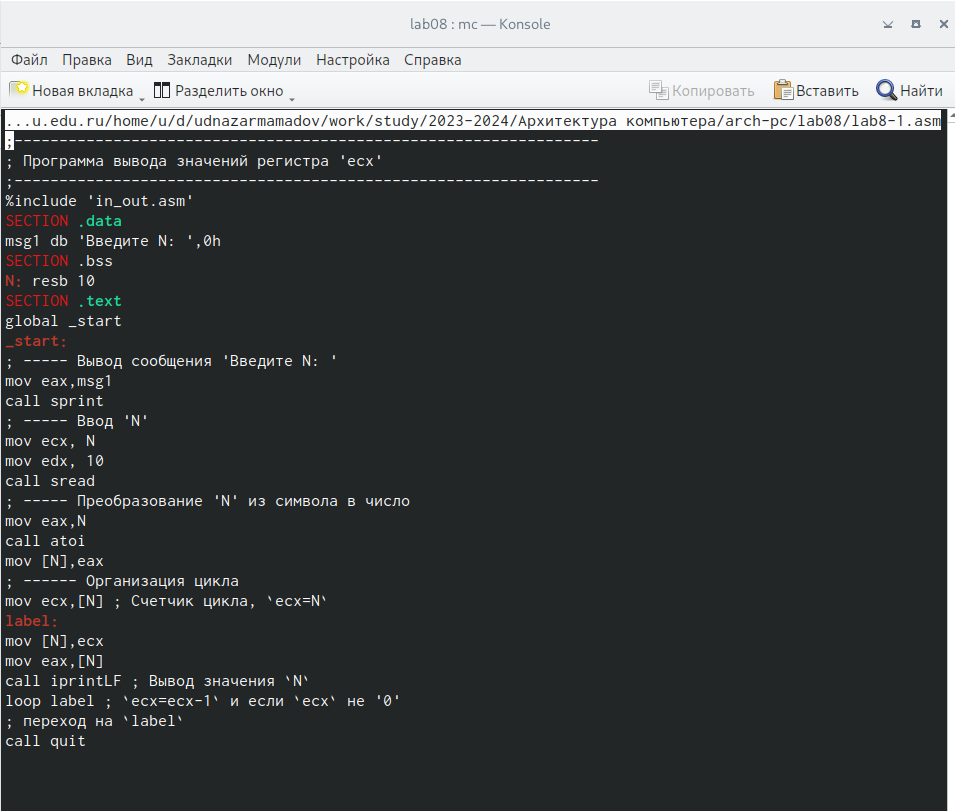
# 4 Выполнение лабораторной работы

Создаю каталог для программам лабораторной работы No 8, перехожу в него и создаю файл lab8-1.asm (рис. ??).



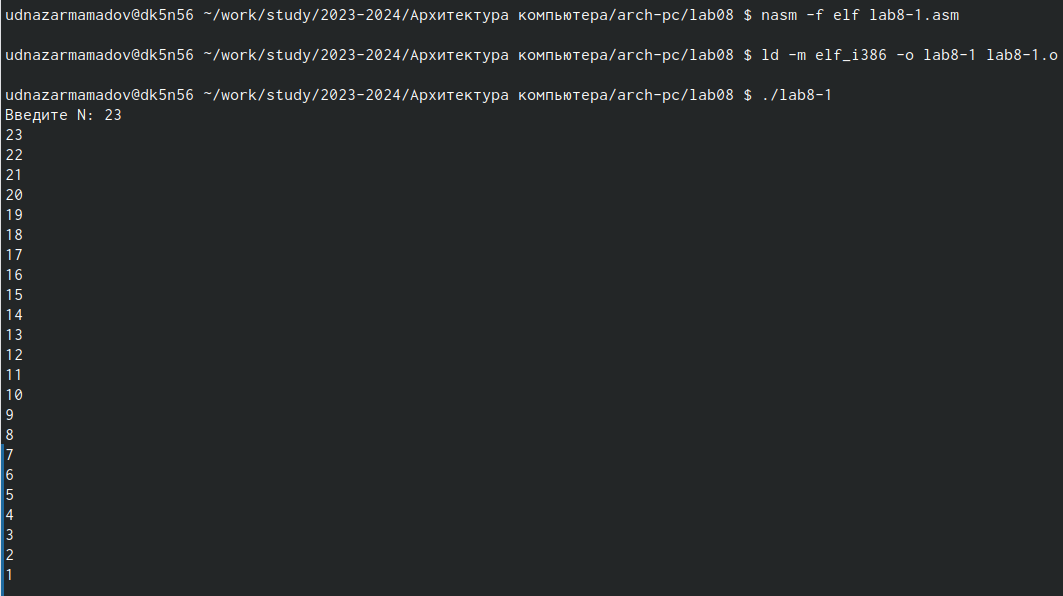
Создание каталога и файла

Захожу в файл lab8-1.asm, и ввожу туда текст программы из листинга 8.1 (рис. ??).



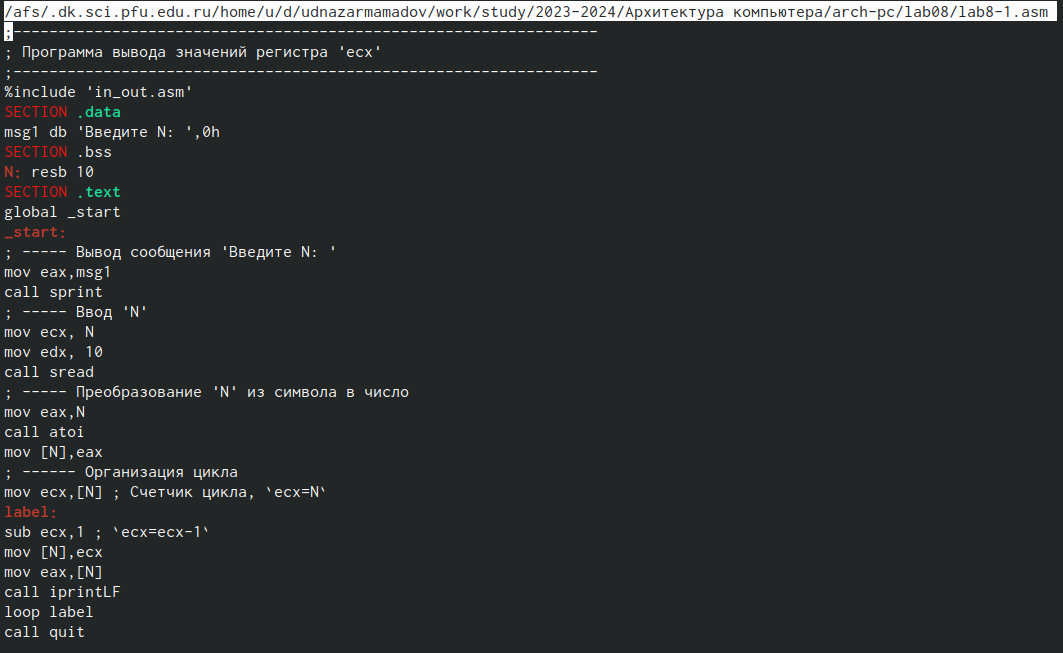
Редактирование файла

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



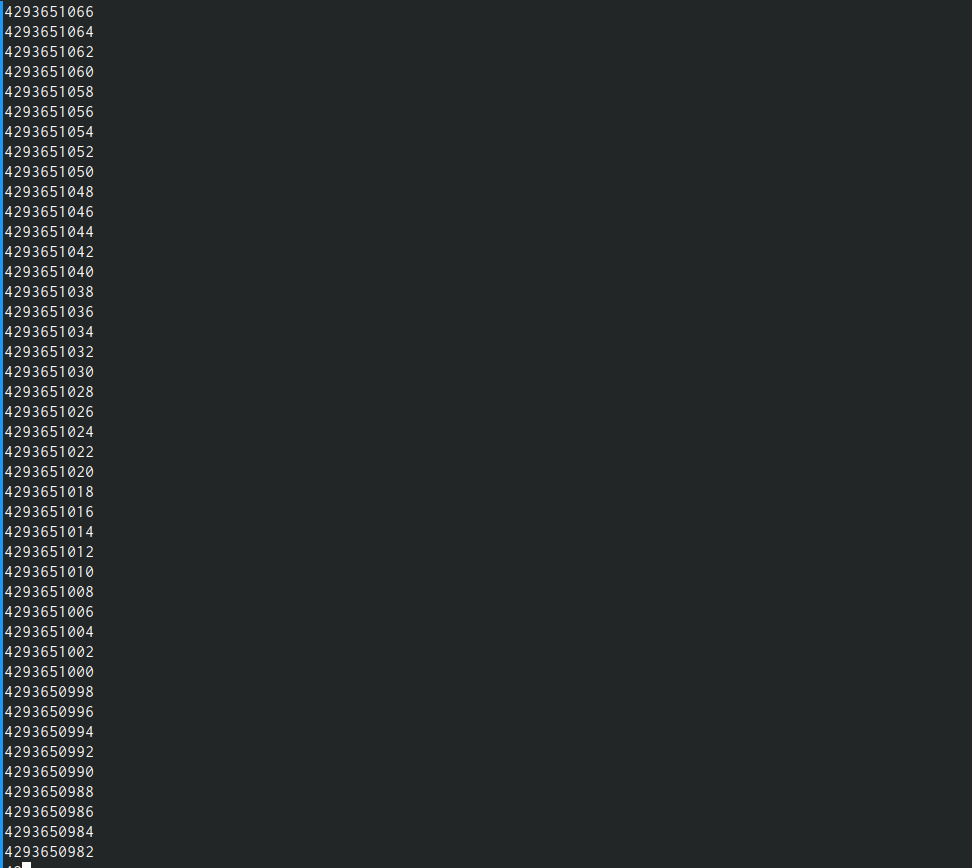
Запуск файла

Изменяю текст программы, добавив изменение значения регистра ecx в цикле (рис. ??).



Редактирование файла

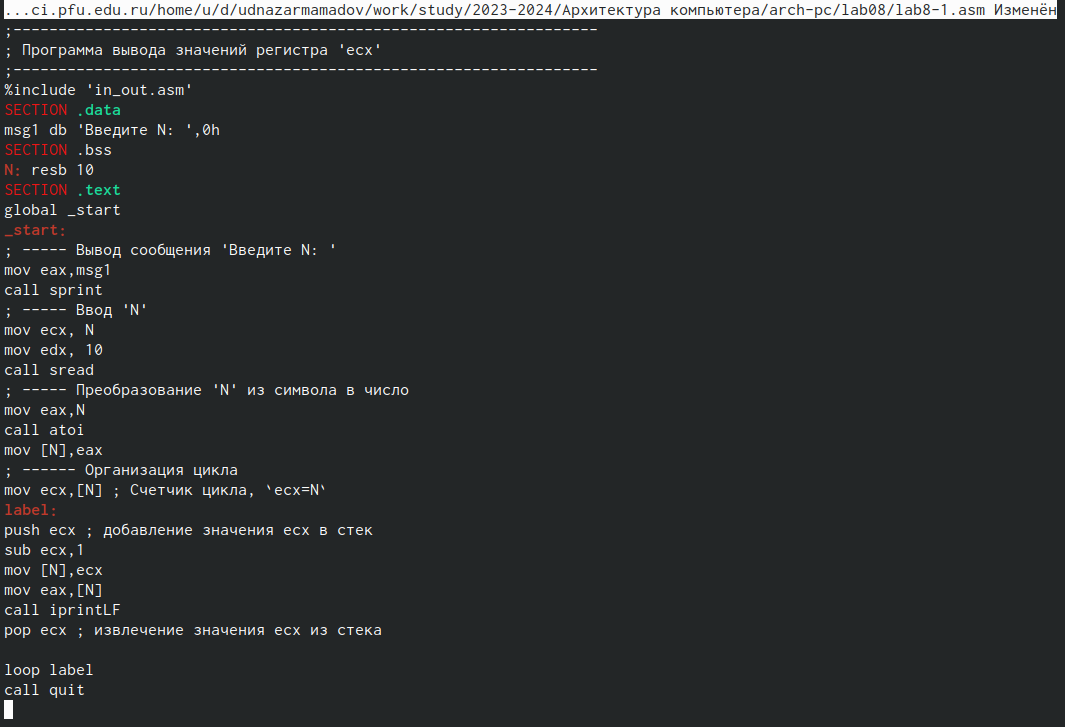
Запускаю файл для проверки (рис. ??).



Запуск файла

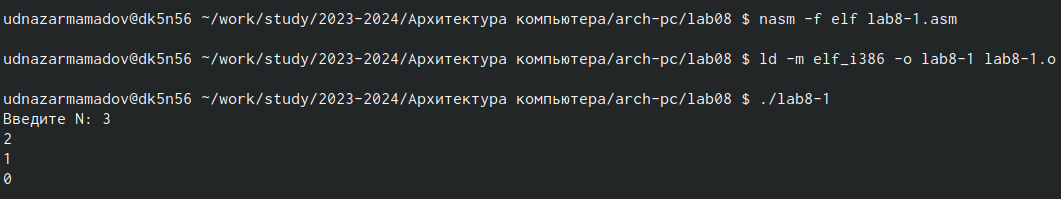
Число проходов цикла не соответствует введенному с клавиатуры значению.

Вношу измененя в программу, добавив команды push и pop для сохранения значения счетчика цикла loop (рис. ??).



Редактирование файла

Создаю исполняемый файл и проверяю его работу (рис. ??).

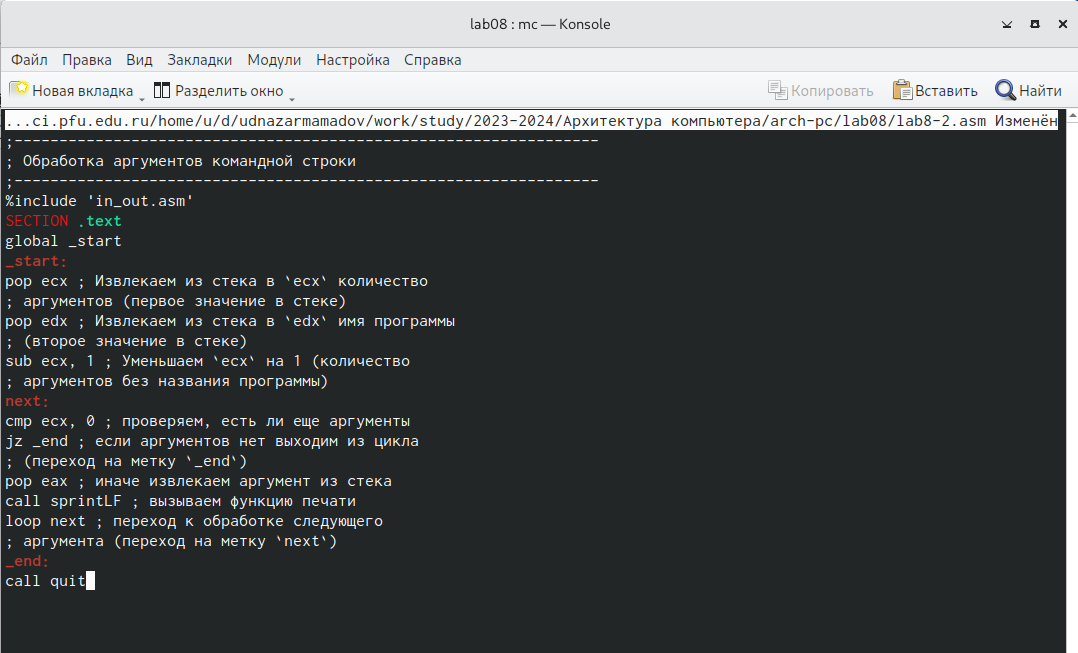


Создание файла

Создаю файл lab8-2.asm в каталоге ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/lab08 и ввожу в него текст программы из листинга 8.2 (рис. ??).

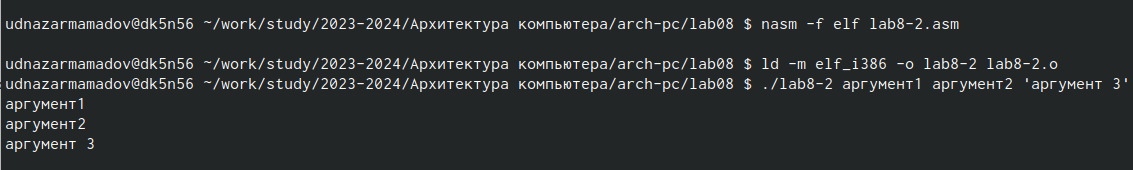
Создание файла

Создание файла



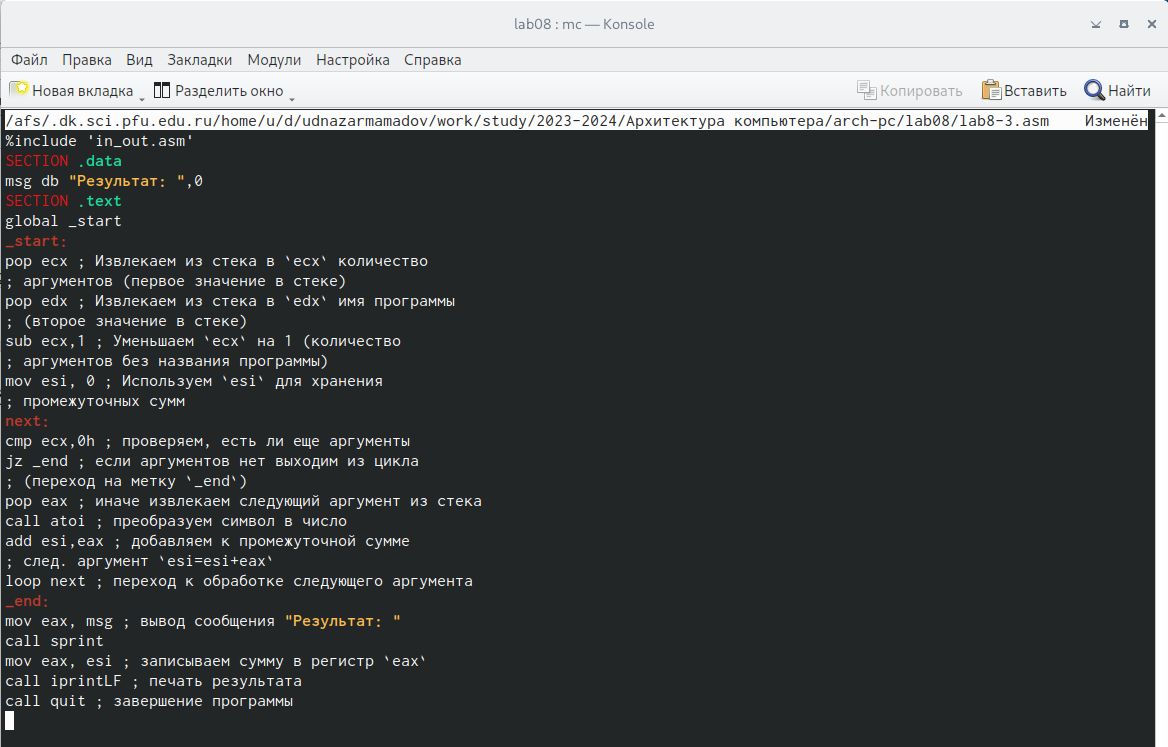
Редактирование файла

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



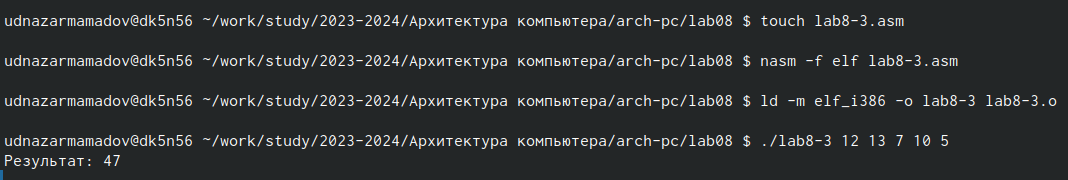
Запуск программы

Создаю еще один файл lab8-3.asm, которая выводит сумму чисел, которые передаются в программу как аргументы (рис. ??).



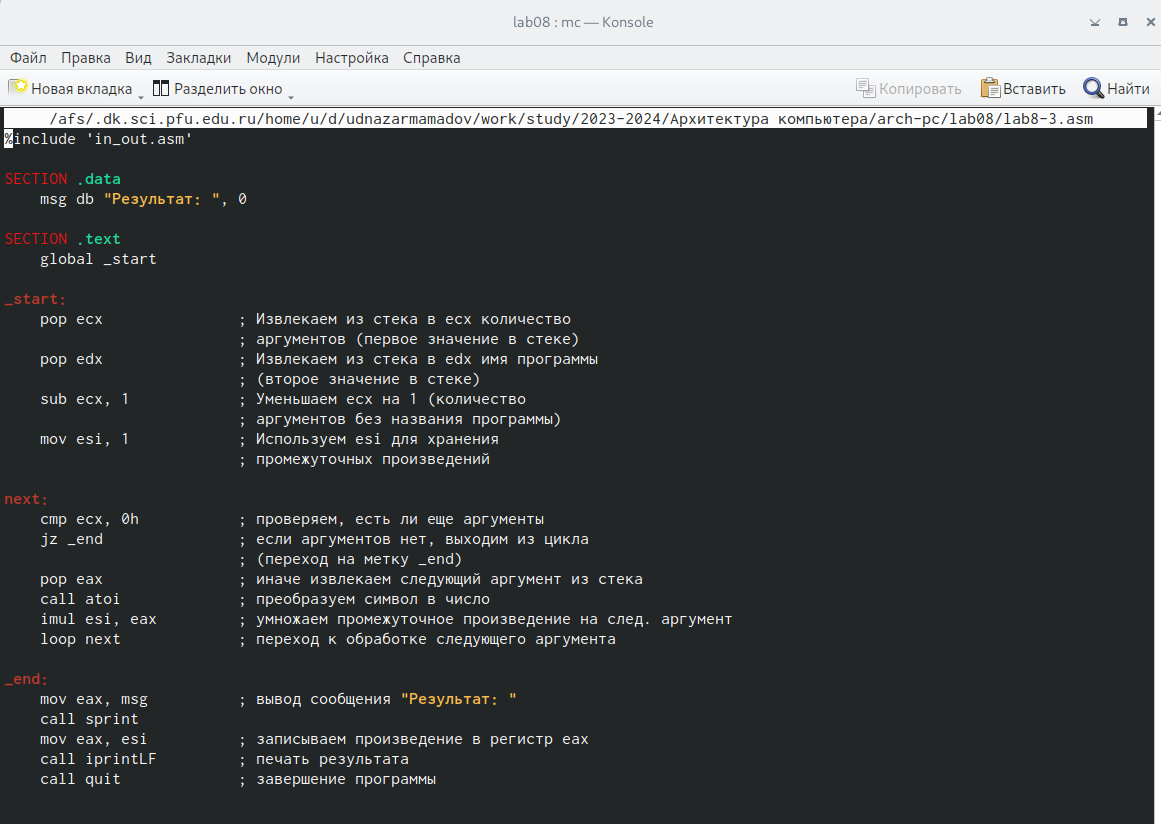
Создание файла

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



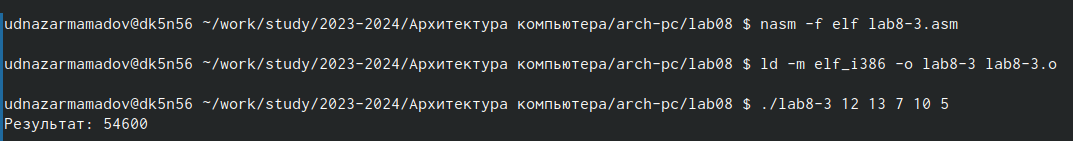
Запуск файла

Изменяю текст программы из листинга 8.3 для вычисления произведения аргументов командной строки (рис. ??).



Редактирование файла

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



Запуск файла

# 5 Вопросы для самопроверки

1. Опишите работу команды loop. Проверка условия: Происходит автоматическое сравнение ecx с нулем. Если ecx не равен нулю, то выполнение программы переходит к следующему шагу. В противном случае, цикл завершается, и выполнение программы продолжается со следующей инструкции после loop. Переход к метке: Если условие (ecx ≠ 0) выполнено, происходит безусловный переход к метке, указанной в destination. Уменьшение счетчика цикла: После выполнения перехода, регистр ecx уменьшается на 1 (ecx = ecx - 1).
2. Как организовать цикл с помощью команд условных переходов, не прибегая к специ- альным командам управления циклами? В NASM можно организовать цикл с использованием команд условных переходов, таких как jmp, je (jump if equal), jne (jump if not equal), jl (jump if less), jg (jump if greater), и других.
3. Дайте определение понятия «стек». Стек — это структура данных, организованная по принципу LIFO («Last In — First Out» или «последним пришёл — первым ушёл»). Стек является частью архитектуры процессора и реализован на аппаратном уровне.
4. Как осуществляется порядок выборки содержащихся в стеке данных? Для стека существует две основные операции: • добавление элемента в вершину стека (push); • извлечение элемента из вершины стека (pop).

# 6 Выводы

При выполнения данной лабораторной работы я приобрел навыки написания программ с использованием циклов и обработкой аргументов командной строки.

# 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).