Отчёт по лабораторной работе №8

Дисциплина: Комьютерные технологии и технологии программирования

Дымовой Д.Д.

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

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

2 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.

push -10 ; Поместить -10 в стек push ebx ; Поместить значение регистра ebx в стек push [buf] ; Поместить значение переменной buf в стек push word [ax] ; Поместить в стек слово по адресу в ax

Существует ещё две команды для добавления значений в стек. Это команда pusha, которая помещает в стек содержимое всех регистров общего назначения в следующем порядке: ax, cx, dx, bx, sp, bp, si, di. А также команда pushf, которая служит для перемещения в стек содержимого регистра флагов. Обе эти команды не имеют операндов.

pop eax ; Поместить значение из стека в регистр eax pop [buf] ; Поместить значение из стека в buf pop word[si] ; Поместить значение из стека в слово по адресу в si.

3 Выполнение лабораторной работы

Я создаю каталог lab08 в директории ~/work/arch-pc/. Ввожу текст программы листинга, создаю исполняемый файл и запускаю его. Этот цикл выводит числа от 5 до 0 (рис. 3.1).

Рис. 3.1: Трансляция и компановка

Вношу изменения согласно условиям лабораторной работы, создаю исполняемый файл и запускаю его (рис. 3.2).

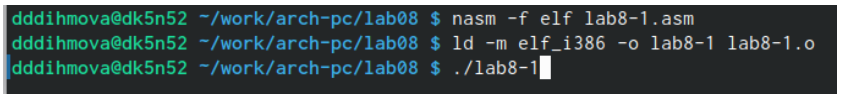

Рис. 3.2: Трансляция и компановка

Фрагмент программы выводимой при вводе с клавиатуры N=5 (рис. 3.3).

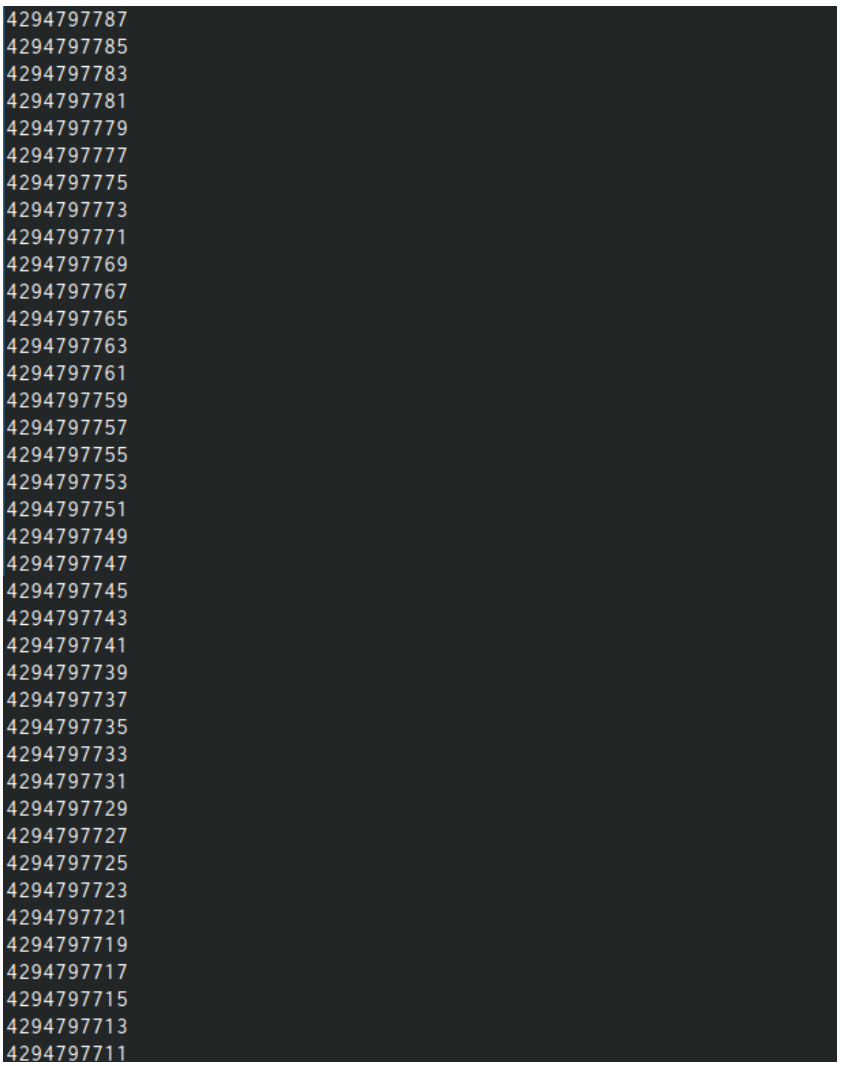

Рис. 3.3: Вывод значений

Значения регистр ecx принимает [N], введенные с клавиатуры, но число проходов цикла не заканчивается.

Снова вношу изменения согласно условиям, создаю исполныемый файл и запускаю его (рис. 3.4).

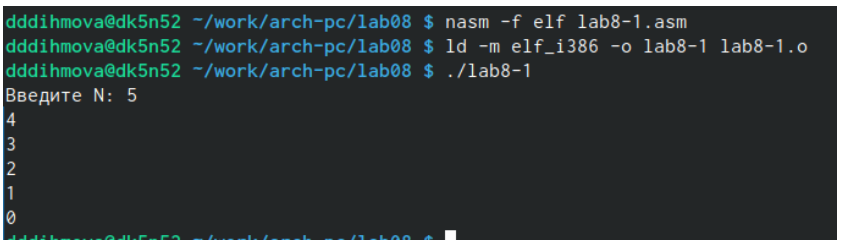

Рис. 3.4: Трансляция компановка и вывод

Да, в этом случае соответсвует.

Создаю файл в каталоге, ввожу в него текст программы листинга 8.2, создаю исполняемый файл и запускаю его, указав аргументы (рис. 3.5).

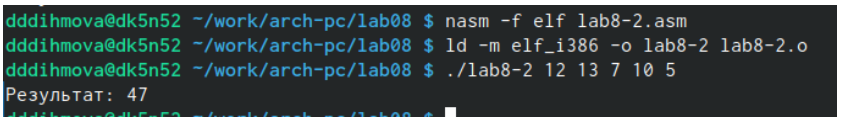

Рис. 3.5: Трансляция, компановка, запуск

Изменяю программу, чтобы вместо суммы выводилось произведение аргументов (рис. 3.6).

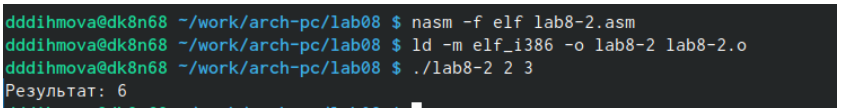

Рис. 3.6: Трансляция, компановка, запуск

add меняю на mul и сохраняю промежуточные суммы (mov esi,eax). А также изчально задаю esi значение 1, чтобы при умножении аргументов не умножалось на 0.

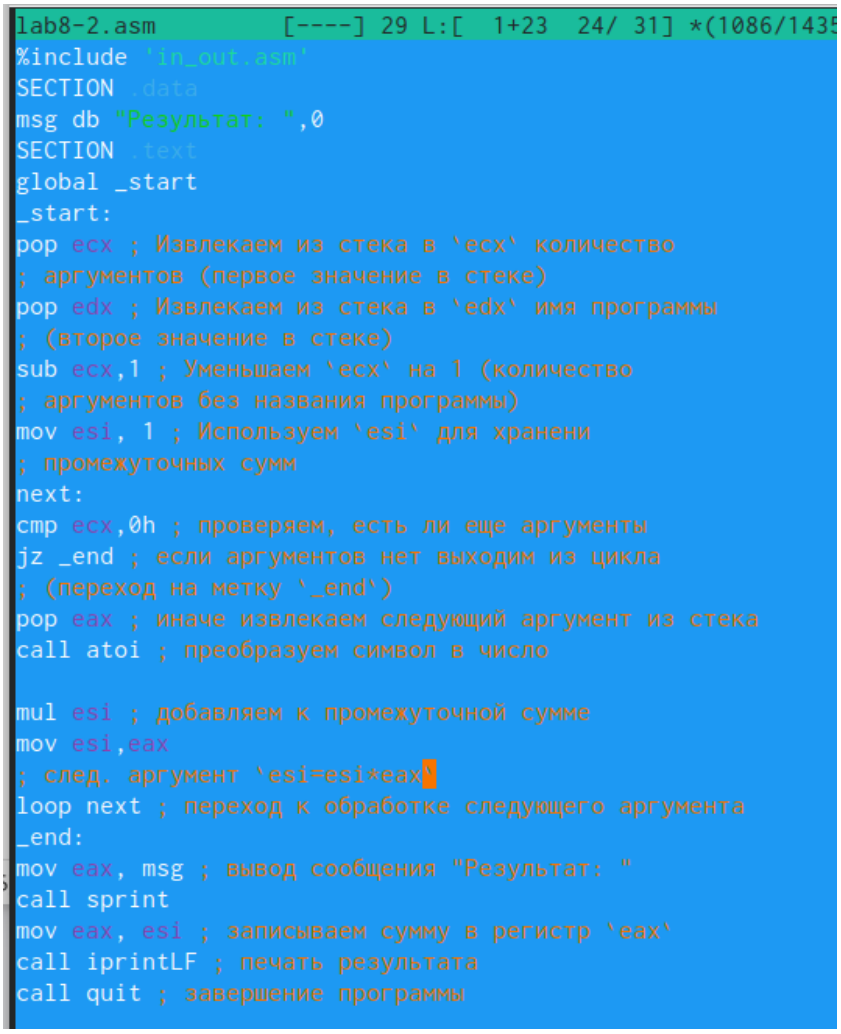

Рис. 3.7: Программа

4 Задание для самостоятельной работы

Создаю файл zadanie.asm в каталоге ~/work/arch-pc/lab08, пишу программу согласно условиям задачи и функции из 13 варианта (рис. 4.1).

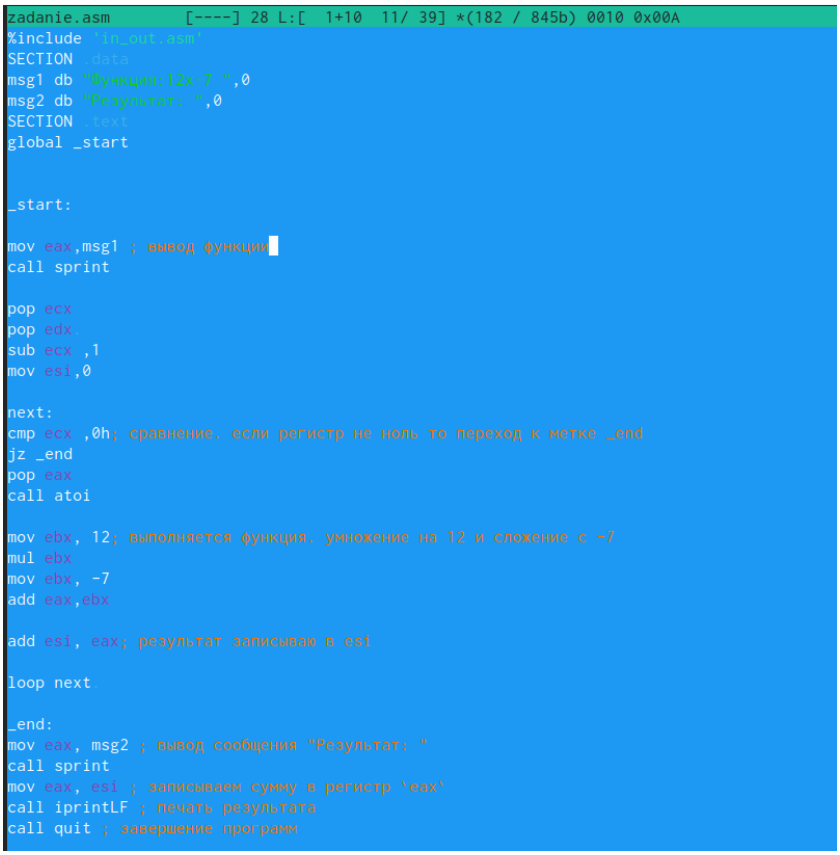

Рис. 4.1: Программа

Создаю исполняемый файл и запускаю его (рис. 4.2).

```
dddihmova@dk8n68 ~/work/arch-pc/lab08 $ nasm -f elf zadanie.asm
dddihmova@dk8n68 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o zadanie zadanie.o
dddihmova@dk8n68 ~/work/arch-pc/lab08 $ ./zadanie 1 2 3 4
Функция:12x-7 Результат: 92
```

Рис. 4.2: Проверка работы программы

5 Выводы

Я приобрела навыки написания программ с использованием циклов и обработкой аргументов командной строки.

Список литературы