Отчёт по лабораторной работе №8

Дисциплина: Архитектура компьютера

Арсений Андреевич Шалин

Содержание

Список иллюстраций

Список таблиц

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

Создал файл lab8-1.asm в каталоге для программ лабораторной работы №8 (рис. 1).

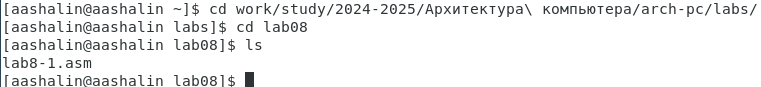


Рис. 1: Задание 8.3.1A

Скопировал программу из листинга 8.1 в lab8-1.asm и создал и запустил исполняемый файл (рис. 2).

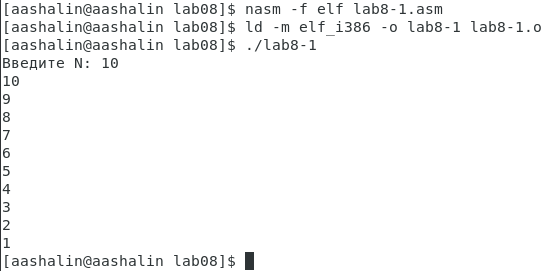


Рис. 2: Задание 8.3.1Б

Добавил изменение значение регистра ecx в цикле (рис. 3).

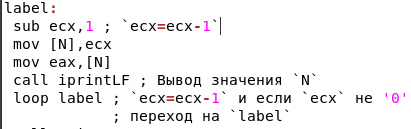


Рис. 3: Задание 8.3.1В

Создал исполняемый файл и проверил его работу (рис. 4).

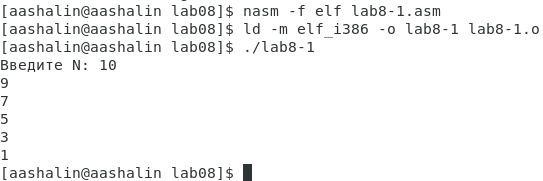


Рис. 4: Задание 8.3.1Г

Добавил команды push и pop для сохранения значения счетчика цикла loop (рис. 5).

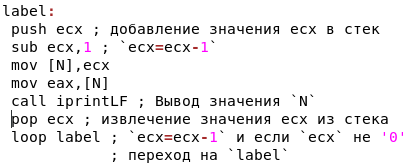


Рис. 5: Задание 8.3.1Д

Создал исполняемый файл и проверил его работу (рис. 6).

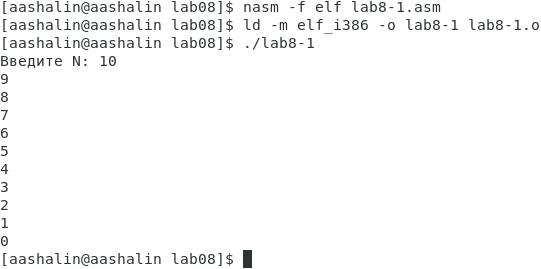


Рис. 6: Задание 8.3.1Е

Скопировал программу из листинга 8.2 в lab8-2.asm (рис. 7).

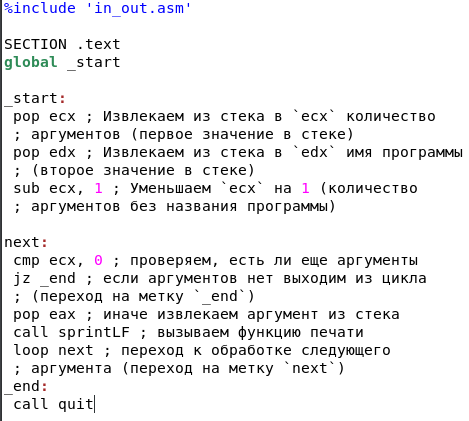


Рис. 7: Задание 8.3.2A

Создал исполняемый файл и проверил его работу, указав аргументы (рис. 8).

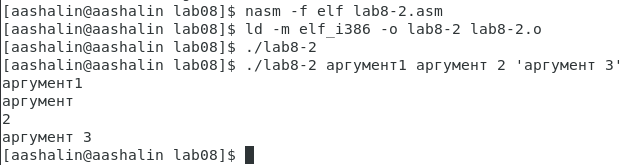


Рис. 8: Задание 8.3.2Б

Скопировал программу из листинга 8.3 в lab8-3.asm и создал и запустил исполняемый файл, указав аргументы (рис. 9).

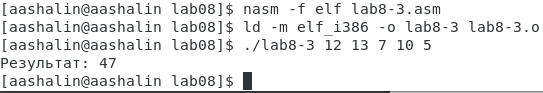


Рис. 9: Задание 8.3.2В

Модифицировал программу из lab8-3.asm для вычисления произведения аргументов командной строки (рис. 10).

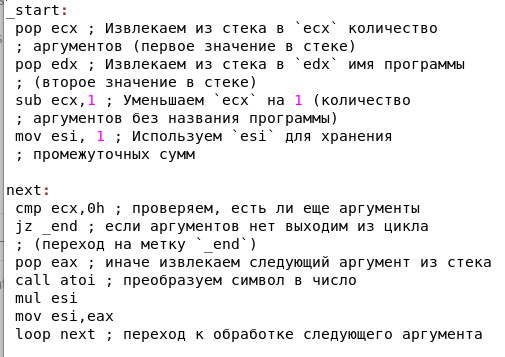


Рис. 10: Задание 8.3.2Г

Создал исполняемый файл и проверил его работу, указав аргументы (рис. 11).

Рис. 11: Задание 8.3.2Д

Рис. 11: Задание 8.3.2Д

# 3 Выполнение cамостоятельной работы

Модифицировал программу из lab8-3.asm в main.asm для вычисления варианта 12, полученного в лабораторной №6 (рис. 12).

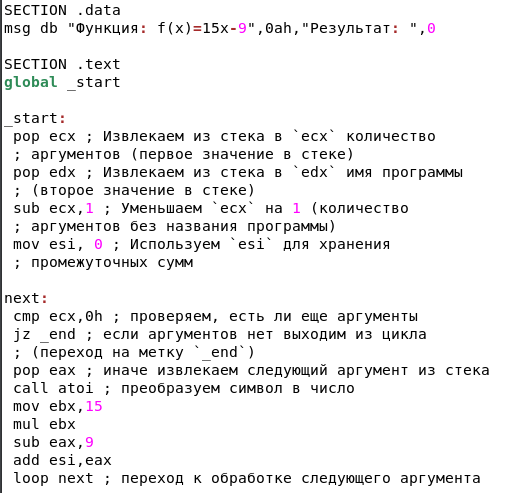


Рис. 12: Задание 8.4A

Создал исполняемый файл и проверил его работу, указав аргументы (рис. 13).

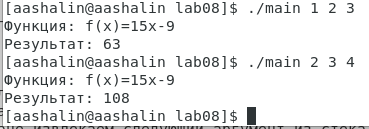


Рис. 13: Задание 8.4Б

Листинг 1., lab8-1.asm

%include 'in\_out.asm'  
  
SECTION .data  
 msg1 db 'Введите N: ',0h  
  
SECTION .bss  
 N: resb 10  
  
SECTION .text  
 global \_start  
\_start:  
  
; ----- Вывод сообщения 'Введите N: '  
 mov eax,msg1  
 call sprint  
  
; ----- Ввод 'N'  
 mov ecx, N  
 mov edx, 10  
 call sread  
  
; ----- Преобразование 'N' из символа в число  
 mov eax,N  
 call atoi  
 mov [N],eax  
  
; ------ Организация цикла  
 mov ecx,[N] ; Счетчик цикла, `ecx=N`  
label:  
 push ecx ; добавление значения ecx в стек  
 sub ecx,1 ; `ecx=ecx-1`  
 mov [N],ecx  
 mov eax,[N]  
 call iprintLF ; Вывод значения `N`  
 pop ecx ; извлечение значения ecx из стека  
 loop label ; `ecx=ecx-1` и если `ecx` не '0'  
 ; переход на `label`  
 call quit

Листинг 2., lab8-2.asm

%include 'in\_out.asm'  
  
SECTION .text  
global \_start  
  
\_start:  
 pop ecx ; Извлекаем из стека в `ecx` количество  
 ; аргументов (первое значение в стеке)  
 pop edx ; Извлекаем из стека в `edx` имя программы  
 ; (второе значение в стеке)  
 sub ecx, 1 ; Уменьшаем `ecx` на 1 (количество  
 ; аргументов без названия программы)  
  
next:  
 cmp ecx, 0 ; проверяем, есть ли еще аргументы  
 jz \_end ; если аргументов нет выходим из цикла  
 ; (переход на метку `\_end`)  
 pop eax ; иначе извлекаем аргумент из стека  
 call sprintLF ; вызываем функцию печати  
 loop next ; переход к обработке следующего  
 ; аргумента (переход на метку `next`)  
\_end:  
 call quit

Листинг 3., lab8-3.asm

%include 'in\_out.asm'  
  
SECTION .data  
msg db "Результат: ",0  
  
SECTION .text  
global \_start  
  
\_start:  
 pop ecx ; Извлекаем из стека в `ecx` количество  
 ; аргументов (первое значение в стеке)  
 pop edx ; Извлекаем из стека в `edx` имя программы  
 ; (второе значение в стеке)  
 sub ecx,1 ; Уменьшаем `ecx` на 1 (количество  
 ; аргументов без названия программы)  
 mov esi, 1 ; Используем `esi` для хранения  
 ; промежуточных сумм  
  
next:  
 cmp ecx,0h ; проверяем, есть ли еще аргументы  
 jz \_end ; если аргументов нет выходим из цикла  
 ; (переход на метку `\_end`)  
 pop eax ; иначе извлекаем следующий аргумент из стека  
 call atoi ; преобразуем символ в число  
 mul esi  
 mov esi,eax  
 loop next ; переход к обработке следующего аргумента  
  
\_end:  
 mov eax, msg ; вывод сообщения "Результат: "  
 call sprint  
 mov eax, esi ; записываем сумму в регистр `eax`  
 call iprintLF ; печать результата  
 call quit ; завершение программы

Листинг 4., main.asm

%include 'in\_out.asm'  
  
SECTION .data  
msg db "Функция: f(x)=15x-9",0ah,"Результат: ",0  
  
SECTION .text  
global \_start  
  
\_start:  
 pop ecx ; Извлекаем из стека в `ecx` количество  
 ; аргументов (первое значение в стеке)  
 pop edx ; Извлекаем из стека в `edx` имя программы  
 ; (второе значение в стеке)  
 sub ecx,1 ; Уменьшаем `ecx` на 1 (количество  
 ; аргументов без названия программы)  
 mov esi, 0 ; Используем `esi` для хранения  
 ; промежуточных сумм  
  
next:  
 cmp ecx,0h ; проверяем, есть ли еще аргументы  
 jz \_end ; если аргументов нет выходим из цикла  
 ; (переход на метку `\_end`)  
 pop eax ; иначе извлекаем следующий аргумент из стека  
 call atoi ; преобразуем символ в число  
 mov ebx,15  
 mul ebx  
 sub eax,9  
 add esi,eax   
 loop next ; переход к обработке следующего аргумента  
  
\_end:  
 mov eax, msg ; вывод сообщения "Результат: "  
 call sprint  
 mov eax, esi ; записываем сумму в регистр `eax`  
 call iprintLF ; печать результата  
 call quit ; завершение программы

Ссылка на отчёт лабораторной №8, сделанный в Markdown. <https://github.com/arsenyshalin/study_2024-2025_arh-pc/tree/master/labs/lab08/report>

# 4 Выводы

* Приобретены навыки написания программ с использованием циклов и обработкой аргументов командной строки.
* Создана программа для вычисления выражения из варианта, полученного из номера студенческого билета в лабораторной №6.

# Список литературы