Отчёт по лабораторной работе №8

Дисциплина: архитектура компьютера

Лысенко Маргарита Олеговна

Содержание

# 1 Цель работы

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

# 2 Задание

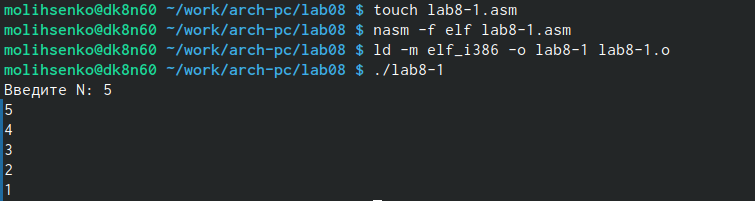
Создать программу, которая находит сумму значений функции f(x)

# 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).

# 4 Выполнение лабораторной работы

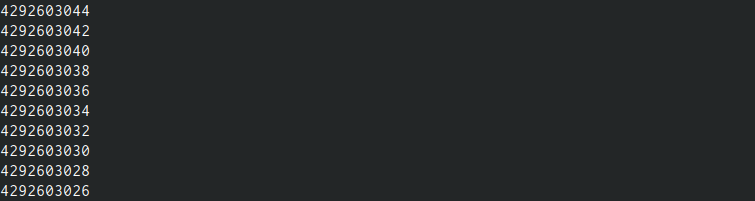
Создала каталог для программам лабораторной работы No 8, перешла в него и создала файл lab8-1.asm. Ввела в файл lab8-1.asm текст программы из листинга 8.1. Создала исполняемый файл и провертла его работу.(рис. ??).



Запуск файла

Изменила текст программы добавив изменение значение регистра ecx в цикле: label: sub ecx,1 ; ecx=ecx-1 mov [N],ecx mov eax,[N] call iprintLF loop label

программа работает некорректно(рис. ??).

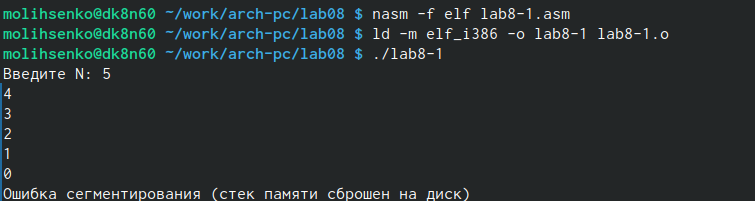


Запуск файла

Внесла изменения в текст программы добавив команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop:

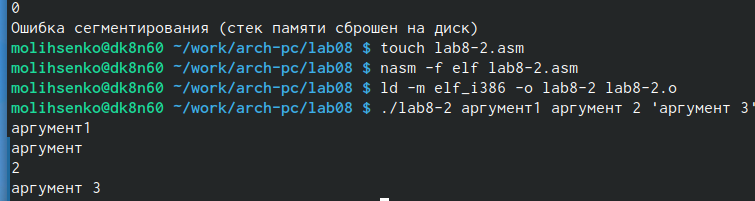
label:  
push ecx ; добавление значения ecx в стек  
sub ecx,1  
mov [N],ecx  
mov eax,[N]  
call iprintLF  
pop ecx ; извлечение значения ecx из стека  
loop label

(рис. ??).



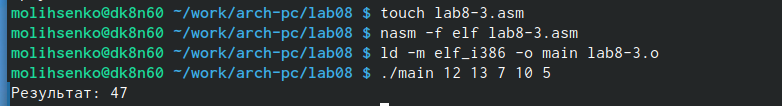
Изменение программы

Создала файл lab8-2.asm в каталоге ~/work/arch-pc/lab08 и ввела в него текст программы из листинга 8.2. Создала исполняемый файл и запустила его, указав аргументы: ./lab8-2 аргумент1 аргумент 2 ‘аргумент 3’ (рис. ??).



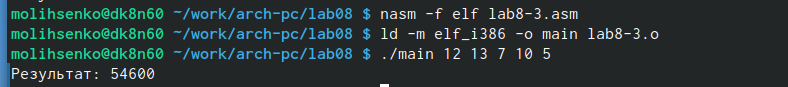
Запуск файла

Создала файл lab8-3.asm в каталоге ~/work/arch-pc/lab08 и ввела в него текст программы из листинга 8.3. (рис. ??).



Запуск программы

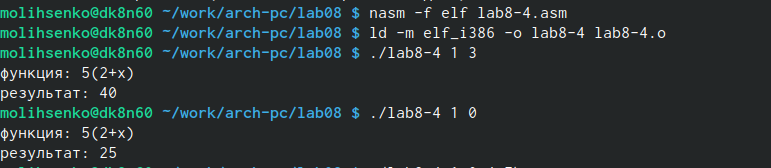
Изменила текст программы из листинга 8.3 для вычисления произведения аргументов командной строки. (рис. ??).



Запуск изменённой программы

# 5 Выполнение самостоятельной работы

Написала программу, которая находит сумму значений функции f(x) с разными значениями х (рис. ??).



Запуск программы

#Листинг

%include 'in\_out.asm'  
  
SECTION .data  
f\_x db "функция: 5(2+x)",0h  
msg db 10,13,'результат: ',0h  
  
SECTION .text  
global \_start  
  
\_start:  
pop ecx  
pop edx  
sub ecx,1  
mov esi, 0  
  
next:  
cmp ecx,0h  
jz \_end  
pop eax  
call atoi  
;dec eax  
add eax,2  
mov ebx, 5  
mul ebx  
add esi, eax  
  
loop next  
  
\_end:  
mov eax, f\_x  
call sprint  
mov eax, msg  
call sprint  
mov eax, esi  
call iprintLF  
  
call quit

# 6 Выводы

В проделанных лабораторной и самостоятельной работах мы риобрели навыки написания программ с использованием циклов и обработкой аргументов командной строки.