

Отчёт по лабораторной работе №9

дисциплина: Архитектура компьютера

Максим Александрович Мишонков

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	7
5	Выводы	16

Список иллюстраций

4.1	Создание каталога и файла	7
4.2	Введение текста программы	8
4.3	Проверка работы исполняемого файла	8
4.4	Изменение текста программы	9
4.5	Проверка работы исполняемого файла	9
4.6	Проверка работы исполняемого файла	10
4.7	Изменение текста программы	11
4.8	Проверка работы исполняемого файла	11
4.9	Создание файла	12
4.10	Введение текста программы	12
4.11	Проверка работы исполняемого файла	12
4.12	Создание файла	13
4.13	Введение текста программы	13
4.14	Проверка работы исполняемого файла	13
4.15	Изменение текста программы	14
4.16	Проверка работы исполняемого файла	14
4.17	Текст программы	15
4.18	Проверка работы исполняемого файла	15

1 Цель работы

Целью данной лабораторной работы является приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки.

2 Задание

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

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.

Для стека существует две основные операции:

Добавление элемента в вершину стека (push)

Извлечение элемента из вершины стека (pop)

4 Выполнение лабораторной работы

1. Создал каталог lab09, перешёл в него и создал файл lab9-1.asm (рис. 4.1)

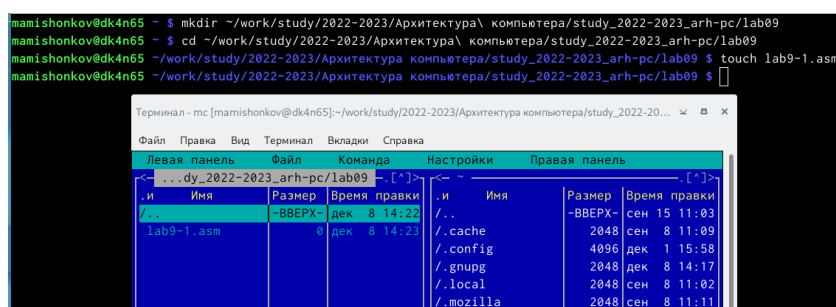


Рис. 4.1: Создание каталога и файла

2. Ввёл в файл текст программы из листинга 9.1. (рис. 4.2)

```

GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/m/a/mamishonkov
#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:
mov [N],ecx
mov eax,[N]
call iprintLF ; Вывод значения 'N'
loop label ; 'ecx=ecx-1' и если 'ecx' не '0'
; переход на 'label'
call quit

```

Рис. 4.2: Введение текста программы

3. Создал исполняемый файл и проверил его работу. (рис. 4.3)

```

mamishonkov@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ nasm -f elf lab9-1.asm
mamishonkov@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ ld -m elf_i386 -o lab9-1 lab9-1.o
mamishonkov@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ ./lab9-1
Введите N: 8
8
7
6
5
4
3
2
1
mamishonkov@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ 

```

Рис. 4.3: Проверка работы исполняемого файла

4. Изменил текст программы, добавив строку `sub eax,1` в теле цикла. (рис. 4.4)


```

GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/m/a/mamishonkov
#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:
sub ecx,1 ; 'ecx=ecx-1'
mov [N],ecx
mov eax,[N]
call iprintLF
loop label
call quit

```

Рис. 4.4: Изменение текста программы

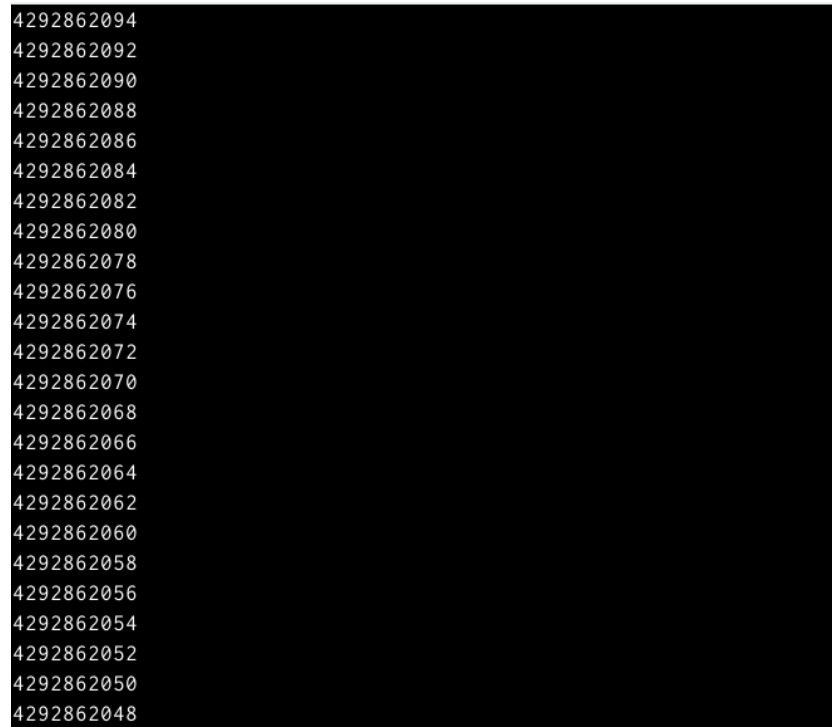
5. Создал исполняемый файл и проверил его работу. Как видно из работы программы, цикл закольцевался и стал бесконечным. (рис. 4.5, 4.6)

```

mamishonkov@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ nasm -f elf lab9-1.asm
mamishonkov@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ ld -m elf_i386 -o lab9-1 lab9-1.o
mamishonkov@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ ./lab9-1
Введите N: 5

```

Рис. 4.5: Проверка работы исполняемого файла



4292862094
4292862092
4292862090
4292862088
4292862086
4292862084
4292862082
4292862080
4292862078
4292862076
4292862074
4292862072
4292862070
4292862068
4292862066
4292862064
4292862062
4292862060
4292862058
4292862056
4292862054
4292862052
4292862050
4292862048

Рис. 4.6: Проверка работы исполняемого файла

6. Изменил текст программы так, чтобы цикл и счётчик работали правильно.
(рис. 4.7)

```

GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/m/a/mamishonkov
%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
mov [N],ecx
mov eax,[N]
call iprintLF
pop ecx ; извлечение значения ecx из стека
loop label

```

Рис. 4.7: Изменение текста программы

7. Создал исполняемый файл и проверил его работу. Число проходов цикла стало соответствовать числу N, введённому с клавиатуры. (рис. 4.8)

```

mamishonkov@dk&n53 ~ $ cd ~/work/study/2022-2023/Архитектура\ компьютера/study_2022-2023_arh-pc/lab09
mamishonkov@dk&n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ nasm -f elf lab9-1.asm
mamishonkov@dk&n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ ld -m elf_i386 -o lab9-1 lab9-1.o
mamishonkov@dk&n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ ./lab9-1
Введите N: 8
7
6
5
4
3
2
1
0

```

Рис. 4.8: Проверка работы исполняемого файла

8. Создал файл lab9-2.asm, ввёл в него текст программы из листинга 9.2. (рис. 4.9, 4.10)

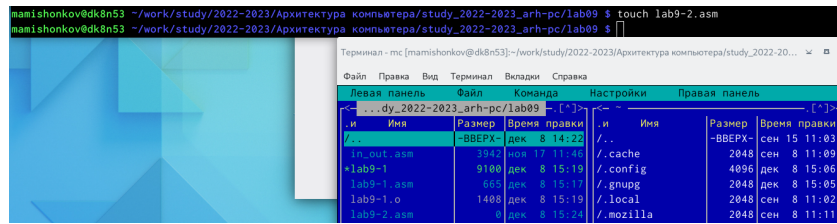


Рис. 4.9: Создание файла

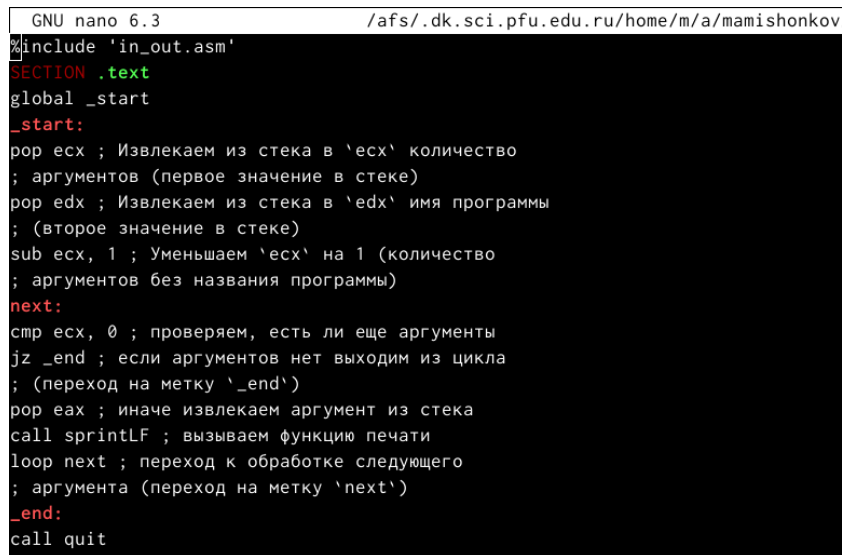


Рис. 4.10: Введение текста программы

9. Создал исполняемый файл и проверил его работу, указав аргументы. Программа выводит 3 аргумента, которые мы указали, но в разной вариации. (рис. 4.11)

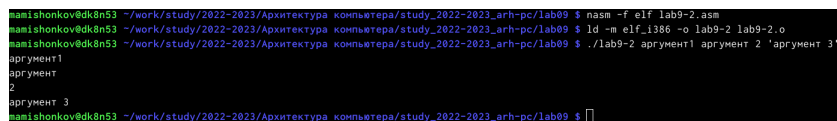


Рис. 4.11: Проверка работы исполняемого файла

10. Создал файл lab9-3.asm, ввёл в него текст программы, которая выводит сумму чисел, передающихся в программу в качестве аргументов. (рис. 4.12, 4.13)

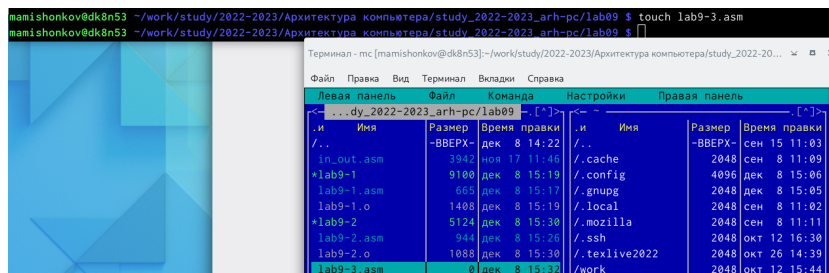


Рис. 4.12: Создание файла

```
GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/m/a/mamishonkov
#include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
    por ecx ; Извлекаем из стека в 'ecx' количество
    ; аргументов (первое значение в стеке)
    por 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:
    stp ecx,0h ; проверяем, есть ли еще аргументы
    jz _end ; если аргументов нет выходим из цикла
    ; (переход на метку '_end')
    por eax ; иначе извлекаем следующий аргумент из стека
    call atoi ; преобразуем символ в число
    add esi,eax ; добавляем к промежуточной сумме
    ; след. аргумент 'esi=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
```

Рис. 4.13: Введение текста программы

11. Создал исполняемый файл и проверил его работу, указав аргументы. (рис. 4.14)

```
mamishonkov@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ nasm -f elf lab9-3.asm
mamishonkov@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ ld -m elf_i386 -o lab9-3 lab9-3.o
mamishonkov@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ ./lab9-3 12 13 7 10 5
Результат: 47
mamishonkov@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $
```

Рис. 4.14: Проверка работы исполняемого файла

12. Изменил текст программы из листинга 9.3 таким образом, чтобы программа вычисляла произведение аргументов, передающихся в программу в качестве аргументов. (рис. 4.15)

```
GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/m/a/mamishonkov.  
%include 'in_out.asm'  
SECTION .data  
msg db "Результат: ",0  
SECTION .text  
global _start  
_start:  
pop ecx  
  
pop edx  
  
sub ecx,1  
  
mov esi,1  
mov eax,1  
  
next:  
cmp ecx,0h  
jz _end  
  
pop eax  
call atoi  
mov ebx,eax  
mov eax,esi  
mul ebx  
mov esi,eax  
loop next  
  
_end:  
mov eax,msg  
call sprint  
mov eax,esi  
call iprintLF  
  
call quit
```

Рис. 4.15: Изменение текста программы

13. Создал исполняемый файл и проверил его работу, указав аргументы. (рис. 4.16)

```
mamishonkov@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ nasm -f elf lab9-3.asm  
mamishonkov@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ ld -m elf_i386 -o lab9-3 lab9-3.o  
mamishonkov@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ ./lab9-3 12 13 7 10 5  
Результат: 54600  
mamishonkov@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $
```

Рис. 4.16: Проверка работы исполняемого файла

Самостоятельная работа

1. Написал программу, которая выводит сумму всех решений функции. Так как в лабораторной работе №7 я получил 14 вариантов, то программу я писал для функции, соответствующей 14 варианту. (рис. 4.17)

```
GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/m/a/mamishonkov
%include 'in_out.asm'

SECTION .data
ms1 db "Функция :f(x)=7(x+1) ", 0
ms2 db "Результат: ", 0

SECTION .text
global _start
_start:
mov eax,ms1
call sprintLF
pop ecx
pop edx
sub ecx,1
mov esi,0
mov ebx,7

next:
cmp ecx,0h
jz _end
pop eax
call atoi
add eax,1
mul ebx
add esi,eax
loop next

_end:
mov eax, ms2
call sprint
mov eax, esi
call iprintLF
call quit
```

Рис. 4.17: Текст программы

2. Создал исполняемый файл и проверил его работу, указав аргументы. (рис. 4.18)

```
mamishonkov@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ nasm -f elf lab9-4.asm
mamishonkov@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ ld -m elf_i386 -o lab9-4 lab9-4.o
mamishonkov@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ ./lab9-4 1 2 3
Функция :f(x)=7(x+1)
Результат: 63
mamishonkov@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ ./lab9-4 1 2 3 4
Функция :f(x)=7(x+1)
Результат: 98
mamishonkov@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $ ./lab9-4 5 6
Функция :f(x)=7(x+1)
Результат: 91
mamishonkov@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab09 $
```

Рис. 4.18: Проверка работы исполняемого файла

5 Выводы

В ходе выполнения данной лабораторной работы я освоил навыки написания программ с использованием циклов и обработкой аргументов командной строки.