

Отчёт по лабораторной работе №8

Дисциплина: Архитектура компьютера

Арсений Андреевич Шалин

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение самостоятельной работы	10
4	Выводы	16
	Список литературы	17

Список иллюстраций

2.1	Задание 8.3.1А	6
2.2	Задание 8.3.1Б	6
2.3	Задание 8.3.1В	6
2.4	Задание 8.3.1Г	7
2.5	Задание 8.3.1Д	7
2.6	Задание 8.3.1Е	7
2.7	Задание 8.3.2А	8
2.8	Задание 8.3.2Б	8
2.9	Задание 8.3.2В	8
2.10	Задание 8.3.2Г	9
2.11	Задание 8.3.2Д	9
3.1	Задание 8.4А	10
3.2	Задание 8.4Б	11

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. 2.1).

```
[aashalin@aashalin ~]$ cd work/study/2024-2025/Архитектура\ компьютера/arch-pc/labs/  
[aashalin@aashalin labs]$ cd lab08  
[aashalin@aashalin lab08]$ ls  
lab8-1.asm  
[aashalin@aashalin lab08]$
```

Рис. 2.1: Задание 8.3.1А

Скопировал программу из листинга 8.1 в lab8-1.asm и создал и запустил исполняемый файл (рис. 2.2).

```
[aashalin@aashalin lab08]$ nasm -f elf lab8-1.asm  
[aashalin@aashalin lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o  
[aashalin@aashalin lab08]$ ./lab8-1  
Введите N: 10  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
[aashalin@aashalin lab08]$
```

Рис. 2.2: Задание 8.3.1Б

Добавил изменение значение регистра ecx в цикле (рис. 2.3).

```
label:  
sub ecx,1 ; `ecx=ecx-1`  
mov [N],ecx  
mov eax,[N]  
call iprintLF ; Вывод значения `N`  
loop label ; `ecx=ecx-1` и если `ecx` не `0`  
-- . ; переход на `label`
```

Рис. 2.3: Задание 8.3.1В

Создал исполняемый файл и проверил его работу (рис. 2.4).

```
[aashalin@aashalin lab08]$ nasm -f elf lab8-1.asm
[aashalin@aashalin lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[aashalin@aashalin lab08]$ ./lab8-1
Введите N: 10
9
7
5
3
1
[aashalin@aashalin lab08]$ █
```

Рис. 2.4: Задание 8.3.1Г

Добавил команды push и pop для сохранения значения счетчика цикла loop (рис. 2.5).

```
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`
```

Рис. 2.5: Задание 8.3.1Д

Создал исполняемый файл и проверил его работу (рис. 2.6).

```
[aashalin@aashalin lab08]$ nasm -f elf lab8-1.asm
[aashalin@aashalin lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[aashalin@aashalin lab08]$ ./lab8-1
Введите N: 10
9
8
7
6
5
4
3
2
1
0
[aashalin@aashalin lab08]$ █
```

Рис. 2.6: Задание 8.3.1Е

Скопировал программу из листинга 8.2 в lab8-2.asm (рис. 2.7).

```

#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 sprintf ; вызываем функцию печати
    loop next ; переход к обработке следующего
             ; аргумента (переход на метку `next`)
_end:
    call quit

```

Рис. 2.7: Задание 8.3.2А

Создал исполняемый файл и проверил его работу, указав аргументы (рис. 2.8).

```

[aashalin@aashalin lab08]$ nasm -f elf lab8-2.asm
[aashalin@aashalin lab08]$ ld -m elf_i386 -o lab8-2 lab8-2.o
[aashalin@aashalin lab08]$ ./lab8-2
аргумент1
аргумент
2
аргумент 3
[aashalin@aashalin lab08]$ █

```

Рис. 2.8: Задание 8.3.2Б

Скопировал программу из листинга 8.3 в lab8-3.asm и создал и запустил исполняемый файл, указав аргументы (рис. 2.9).

```

[aashalin@aashalin lab08]$ nasm -f elf lab8-3.asm
[aashalin@aashalin lab08]$ ld -m elf_i386 -o lab8-3 lab8-3.o
[aashalin@aashalin lab08]$ ./lab8-3 12 13 7 10 5
Результат: 47
[aashalin@aashalin lab08]$ █

```

Рис. 2.9: Задание 8.3.2В

Модифицировал программу из lab8-3.asm для вычисления произведения аргументов командной строки (рис. 2.10).


```

_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

```

Рис. 2.10: Задание 8.3.2Г

Создал исполняемый файл и проверил его работу, указав аргументы (рис. 2.11).

```

[aashalin@aashalin lab08]$ ./lab8-3 2 3 4
Результат: 24
[aashalin@aashalin lab08]$ █

```

Рис. 2.11: Задание 8.3.2Д

3 Выполнение самостоятельной работы

Модифицировал программу из lab8-3.asm в main.asm для вычисления варианта 12, полученного в лабораторной №6 (рис. 3.1).

```
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
```

Рис. 3.1: Задание 8.4А

Создал исполняемый файл и проверил его работу, указав аргументы (рис. 3.2).

```

[aashalin@aashalin lab08]$ ./main 1 2 3
Функция:  $f(x)=15x-9$ 
Результат: 63
[aashalin@aashalin lab08]$ ./main 2 3 4
Функция:  $f(x)=15x-9$ 
Результат: 108
[aashalin@aashalin lab08]$ █

```

Рис. 3.2: 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 sprintf ; 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