

Отчёт по лабораторной работе №8

Дисциплина: архитектура компьютера

Логинов Георгий Евгеньевич

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Реализация циклов в NASM	8
4.2	Обработка аргументов командной строки	11
4.3	Задание для самостоятельной работы (Вариант 20)	15
5	Выводы	19
	Список литературы	20

Список иллюстраций

4.1	Создание каталога и файла в ней, редактирование файла	8
4.2	Исполнение программы	9
4.3	Изменение текста программы	9
4.4	Исполнение программы	10
4.5	Изменение программы	10
4.6	Исполнение программы	10
4.7	Создание и редактирование файла	11
4.8	Запуск исполняемого файла	11
4.9	Создание файла и его редактирование	12
4.10	Запуск исполняемого файла	13
4.11	Создание файла и его редактирование	13
4.12	Запуск исполняемого файла	14
4.13	Создание и редактирование файла	16
4.14	Запуск исполняемого файла	18

Список таблиц

1 Цель работы

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

2 Задание

1. Реализация циклов в NASM
2. Обработка аргументов командной строки
3. Задание для самостоятельной работы

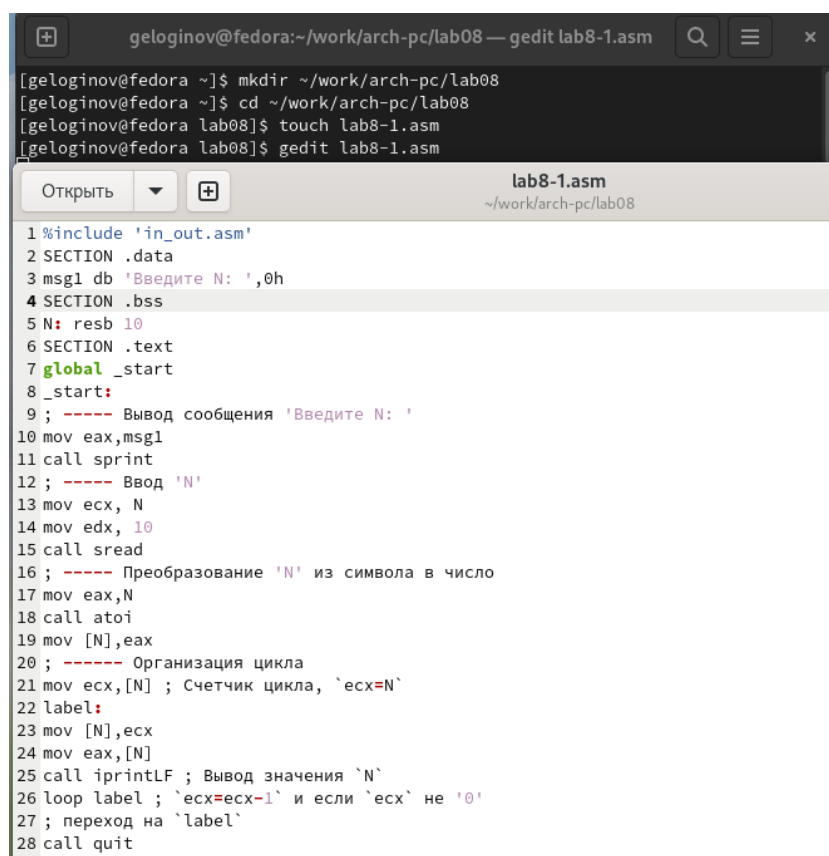
3 Теоретическое введение

Цикл в программировании — это управляющая конструкция, которая заставляет какой-то блок кода выполняться несколько раз.

4 Выполнение лабораторной работы

4.1 Реализация циклов в NASM

Создаю каталог для программ для лабораторной работы №8, перехожу в него и создаю файл lab8-1.asm, в который ввожу программу вывода значений регистра ecx (рис. 4.1).



The image shows a terminal window and a text editor window. The terminal window displays the following commands and output:

```
[geloginov@fedora ~]$ mkdir ~/work/arch-pc/lab08
[geloginov@fedora ~]$ cd ~/work/arch-pc/lab08
[geloginov@fedora lab08]$ touch lab8-1.asm
[geloginov@fedora lab08]$ gedit lab8-1.asm
```

The text editor window, titled "lab8-1.asm", shows the following assembly code:

```
1 %include 'in_out.asm'
2 SECTION .data
3 msg1 db 'Введите N: ',0h
4 SECTION .bss
5 N: resb 10
6 SECTION .text
7 global _start
8 _start:
9 ; ----- Вывод сообщения 'Введите N: '
10 mov eax,msg1
11 call sprint
12 ; ----- Ввод 'N'
13 mov ecx, N
14 mov edx, 10
15 call sread
16 ; ----- Преобразование 'N' из символа в число
17 mov eax,N
18 call atoi
19 mov [N],eax
20 ; ----- Организация цикла
21 mov ecx,[N] ; Счетчик цикла, `ecx=N`
22 label:
23 mov [N],ecx
24 mov eax,[N]
25 call iprintLF ; Вывод значения `N`
26 loop label ; `ecx=ecx-1` и если `ecx` не `0`
27 ; переход на `label`
28 call quit
```

Рис. 4.1: Создание каталога и файла в ней, редактирование файла

Создаю исполняемый файл и запускаю его (рис. 4.2). Программа отработала корректно.

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

Рис. 4.2: Исполнение программы

Изменяю текст программы добавив изменение значения регистра есх в цикле (рис. 4.3).

```
[geloginov@fedora lab08]$ gedit lab8-1.asm
Открыть *lab8-1.asm ~/work/arch-pc/lab08
20 ; ----- Организация цикла
21 mov ecx,[N] ; Счетчик цикла, `ecx=N`
22 label:
23 sub ecx, 1
24 mov [N],ecx
25 mov eax,[N]
26 call iprintLF ; Вывод значения `N`
27 loop label ; `ecx=ecx-1` и если `ecx` не `0`
28 ; переход на `label`
```

Рис. 4.3: Изменение текста программы

Создаю исполняемый файл и запускаю его (рис. 4.4). Число проходов цикла равно $N/2$.

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

Рис. 4.4: Исполнение программы

Вношу изменения в текст программы добавив команды push и pop для сохранения значения счётчика loop(рис. 4.5).

```
[geloginov@fedora lab08]$ gedit lab8-1.asm
Открыть *lab8-1.asm ~/work/arch-pc/lab08
21 mov ecx,[N] ; Счетчик цикла, `ecx=N`
22 label:
23 push ecx
24 sub ecx, 1
25 mov [N],ecx
26 mov eax,[N]
27 call iprintLF ; Вывод значения `N`
28 pop ecx
29 loop label ; `ecx=ecx-1` и если `ecx` не `0`
```

Рис. 4.5: Изменение программы

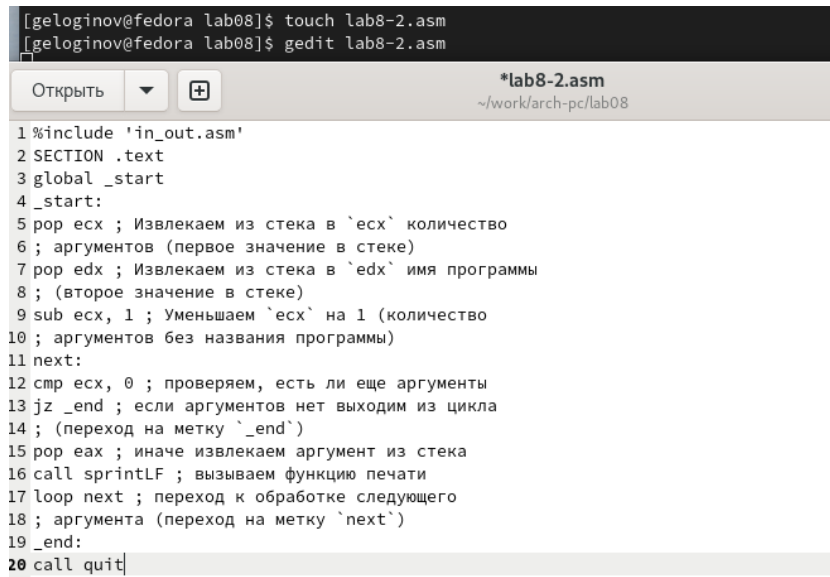
Создаю исполняемый файл и проверяю его работу (рис. 4.6). Число проходов соответствует значению N введённому с клавиатуры.

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

Рис. 4.6: Исполнение программы

4.2 Обработка аргументов командной строки

Создаю файл lab8-2.asm и ввожу в него программу обработки аргументов командной строки (рис. 4.7).



```
[geloginov@fedora lab08]$ touch lab8-2.asm
[geloginov@fedora lab08]$ gedit lab8-2.asm
```

Открыть [иконка] *lab8-2.asm
~/work/arch-pc/lab08

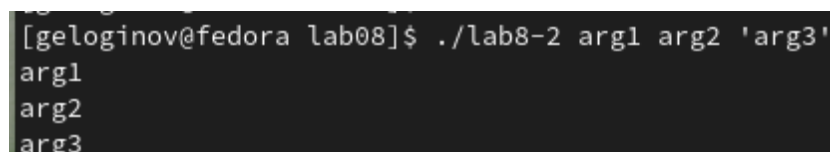
```
1 %include 'in_out.asm'
2 SECTION .text
3 global _start
4 _start:
5 pop ecx ; Извлекаем из стека в `ecx` количество
6 ; аргументов (первое значение в стеке)
7 pop edx ; Извлекаем из стека в `edx` имя программы
8 ; (второе значение в стеке)
9 sub ecx, 1 ; Уменьшаем `ecx` на 1 (количество
10 ; аргументов без названия программы)
11 next:
12 cmp ecx, 0 ; проверяем, есть ли еще аргументы
13 jz _end ; если аргументов нет выходим из цикла
14 ; (переход на метку `_end`)
15 pop eax ; иначе извлекаем аргумент из стека
16 call printf ; вызываем функцию печати
17 loop next ; переход к обработке следующего
18 ; аргумента (переход на метку `next`)
19 _end:
20 call quit
```

Рис. 4.7: Создание и редактирование файла

Создаю исполняемый файл (рис. ??).

```
[geloginov@fedora lab08]$ nasm -f elf lab8-2.asm
[geloginov@fedora lab08]$ ld -m elf_i386 -o lab8-2 lab8-2.o
```

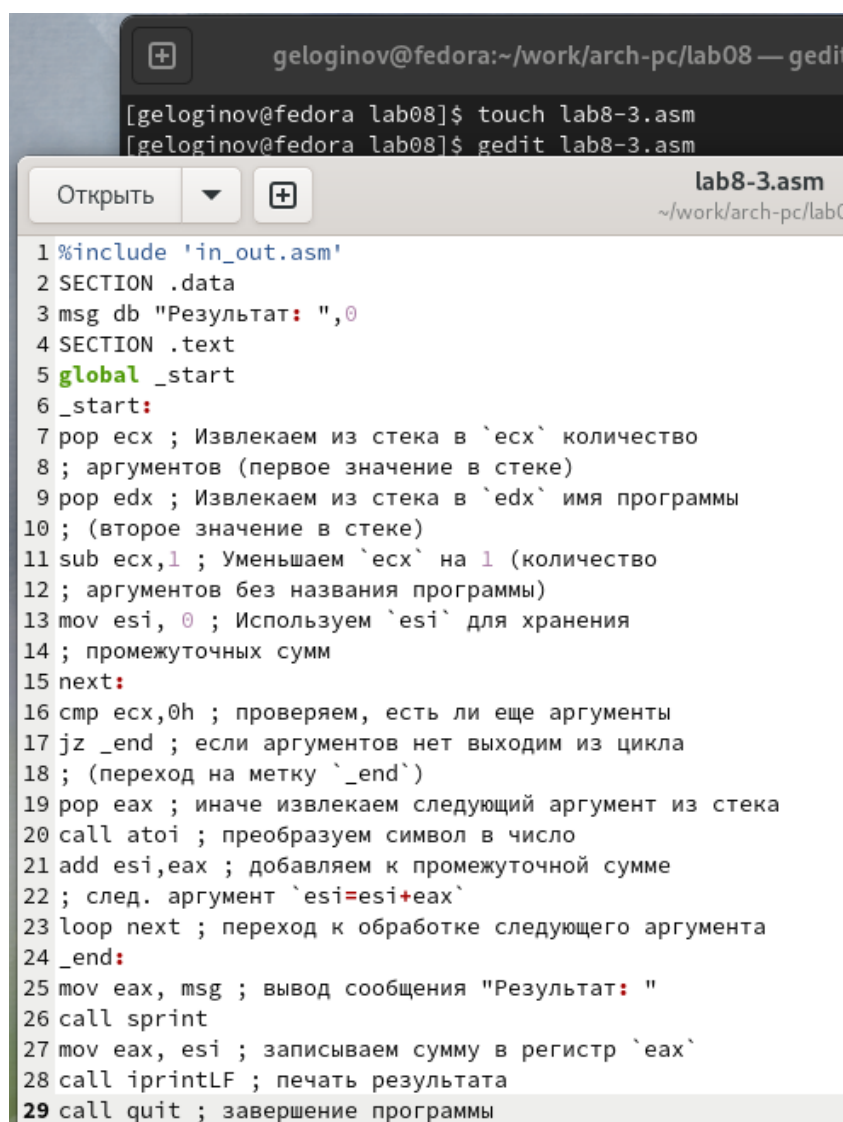
Запускаю исполняемый файл (рис. 4.8). Программой были обработаны все аргументы.



```
[geloginov@fedora lab08]$ ./lab8-2 arg1 arg2 'arg3'
arg1
arg2
arg3
```

Рис. 4.8: Запуск исполняемого файла

Создаю файл lab8-3.asm и ввожу в него программу вычисления суммы аргументов командной строки (рис. 4.9).



The image shows a terminal window at the top with the following commands and output:

```
geloginov@fedora:~/work/arch-pc/lab08 — gedit
[geloginov@fedora lab08]$ touch lab8-3.asm
[geloginov@fedora lab08]$ gedit lab8-3.asm
```

Below the terminal is a text editor window titled "lab8-3.asm". The editor contains the following assembly code:

```
1 %include 'in_out.asm'
2 SECTION .data
3 msg db "Результат: ",0
4 SECTION .text
5 global _start
6 _start:
7 pop ecx ; Извлекаем из стека в `ecx` количество
8 ; аргументов (первое значение в стеке)
9 pop edx ; Извлекаем из стека в `edx` имя программы
10 ; (второе значение в стеке)
11 sub ecx,1 ; Уменьшаем `ecx` на 1 (количество
12 ; аргументов без названия программы)
13 mov esi, 0 ; Используем `esi` для хранения
14 ; промежуточных сумм
15 next:
16 cmp ecx,0h ; проверяем, есть ли еще аргументы
17 jz _end ; если аргументов нет выходим из цикла
18 ; (переход на метку `_end`)
19 pop eax ; иначе извлекаем следующий аргумент из стека
20 call atoi ; преобразуем символ в число
21 add esi,eax ; добавляем к промежуточной сумме
22 ; след. аргумент `esi=esi+eax`
23 loop next ; переход к обработке следующего аргумента
24 _end:
25 mov eax, msg ; вывод сообщения "Результат: "
26 call sprint
27 mov eax, esi ; записываем сумму в регистр `eax`
28 call iprintLF ; печать результата
29 call quit ; завершение программы
```

Рис. 4.9: Создание файла и его редактирование

Создаю исполняемый файл и запускаю его, указав аргументы 12, 13, 7, 10, 5 (рис. 4.10). Программа отработала корректно.

```
geloginov@fedora:~/work/arch-pc/lab08
[geloginov@fedora lab08]$ nasm -f elf lab8-3.asm
[geloginov@fedora lab08]$ ld -m elf_i386 -o lab8-3 lab8-3.o
[geloginov@fedora lab08]$ ./lab8-3 12 13 7 10 5
Результат: 47
[geloginov@fedora lab08]$
```

Рис. 4.10: Запуск исполняемого файла

Создаю файл lab8-3-1.asm и ввожу в него программу вычисления произведения аргументов командной строки (рис. 4.11).

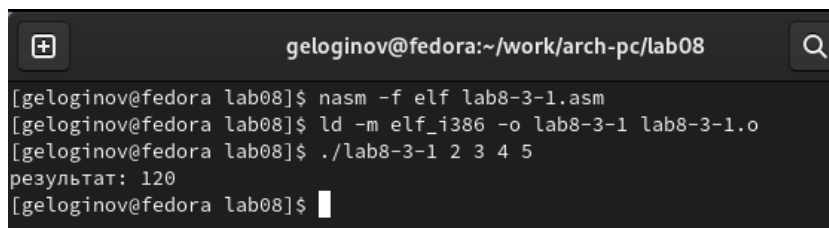
```
geloginov@fedora:~/work/arch-pc/lab08 — gedit lab8-3-1.asm
[geloginov@fedora lab08]$ touch lab8-3-1.asm
[geloginov@fedora lab08]$ gedit lab8-3-1.asm

lab8-3-1.asm
~/work/arch-pc/lab08

1 %include "in_out.asm"
2 SECTION .data
3 msg db 'результат: '
4 SECTION .text
5 GLOBAL _start
6
7 _start:
8 pop ecx
9 pop edx
10 sub ecx,1
11 mov esi,1
12
13 next:
14 cmp ecx,0
15 jz _end
16
17 pop eax
18 call atoi
19 mul esi
20 mov esi, eax
21
22 loop next
23
24 _end:
25 mov eax, msg
26 call sprint
27 mov eax, esi
28 call iprintLF
29 call quit
30
```

Рис. 4.11: Создание файла и его редактирование

Создаю исполняемый файл и запускаю его, указав аргументы 2, 3, 4, 5 (рис. 4.12). Программа отработала корректно.

A terminal window with a dark background. The title bar shows 'geloginov@fedora:~/work/arch-pc/lab08'. The terminal contains the following text:

```
[geloginov@fedora lab08]$ nasm -f elf lab8-3-1.asm
[geloginov@fedora lab08]$ ld -m elf_i386 -o lab8-3-1 lab8-3-1.o
[geloginov@fedora lab08]$ ./lab8-3-1 2 3 4 5
результат: 120
[geloginov@fedora lab08]$
```

Рис. 4.12: Запуск исполняемого файла

Текст программы из файла lab8-3-1.asm:

```
%include "in_out.asm"
SECTION .data
msg db 'результат: '
SECTION .text
GLOBAL _start

_start:
pop ecx
pop edx
sub ecx,1
mov esi,1

next:
cmp ecx,0
jz _end

pop eax
call atoi
mul esi
```

```
mov esi, eax
```

```
loop next
```

```
_end:
```

```
mov eax, msg
```

```
call sprint
```

```
mov eax, esi
```

```
call iprintLF
```

```
call quit
```

4.3 Задание для самостоятельной работы (Вариант 20)

Создаю файл lab8-4.asm и ввожу текст программы для вычисления суммы значений функции $f(x)=3(10+x)$ (рис. 4.13).

```
[geloginov@fedora lab08]$ touch lab8-4.asm
[geloginov@fedora lab08]$ gedit lab8-4.asm

lab8-4.asm
~/work/arch-pc/lab08

1 %include 'in_out.asm'
2
3 SECTION .data
4 f_x db "функция: f(x)=3(10 + x)",0h
5 msg db 10,13,'результат: ',0h
6
7 SECTION .text
8 global _start
9
10 _start:
11 pop ecx
12 pop edx
13 sub ecx, 1
14 mov esi, 0
15
16 next:
17 cmp ecx,0h
18 jz _end
19 pop eax
20 call atoi
21 add eax, 10
22 mov ebx, 3
23 mul ebx
24 add esi, eax
25
26 loop next
27
28 _end:
29 mov eax, f_x
30 call sprint
31 mov eax, msg
32 call sprint
33 mov eax, esi
34 call iprintLF
```

Рис. 4.13: Создание и редактирование файла

Текст программы из файла lab8-3-1.asm:

```
%include 'in_out.asm'

SECTION .data
f_x db "функция: f(x)=3(10 + 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
add eax, 10
mov ebx, 3
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

```

Создаю исполняемый файл (рис. ??).

```
[geloginov@fedora lab08]$ nasm -f elf lab8-4.asm
[geloginov@fedora lab08]$ ld -m elf_i386 -o lab8-4 lab8-4.o
```

Запускаю исполняемый файл с разными аргументами (рис. 4.14). Программа отработала корректно.

```
bash: ./lab8-4: cannot execute: Permission denied
[geologinov@fedora lab08]$ ./lab8-4 1 2 3
функция:  $f(x)=3(10 + x)$ 
результат: 108
[geologinov@fedora lab08]$ ./lab8-4 10 20 30
функция:  $f(x)=3(10 + x)$ 
результат: 270
[geologinov@fedora lab08]$
```

Рис. 4.14: Запуск исполняемого файла

5 Выводы

Были получены навыки по организации циклов и работе со стеком на языке NASM.

Список литературы

1. Лабораторная работа №8