

Отчёт по лабораторной работе

Дисциплина: Архитектура ЭВМ

Перегудов Александр Вадимович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	18
	Список литературы	19

Список иллюстраций

4.1	Создание каталога и файла	8
4.2	Текст программы	9
4.3	Результат работы программы	9
4.4	Изменённый текст программы	10
4.5	Результат работы программы	10
4.6	Изменённый текст программы	11
4.7	Результат работы программы	11
4.8	Создание файла	12
4.9	Текст программы	12
4.10	Результат работы программы	12
4.11	Создание файла	13
4.12	Текст программы	13
4.13	Результат работы программы	14
4.14	Изменённый текст программы	14
4.15	Результат работы программы	15
4.16	Создание файла	15
4.17	Текст программы	16
4.18	Трансляция и компоновка	16
4.19	Результат работы программы	17

Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . .	7
-----	---	---

1 Цель работы

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

2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы. Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux

Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно об Unix см. в [1–6].

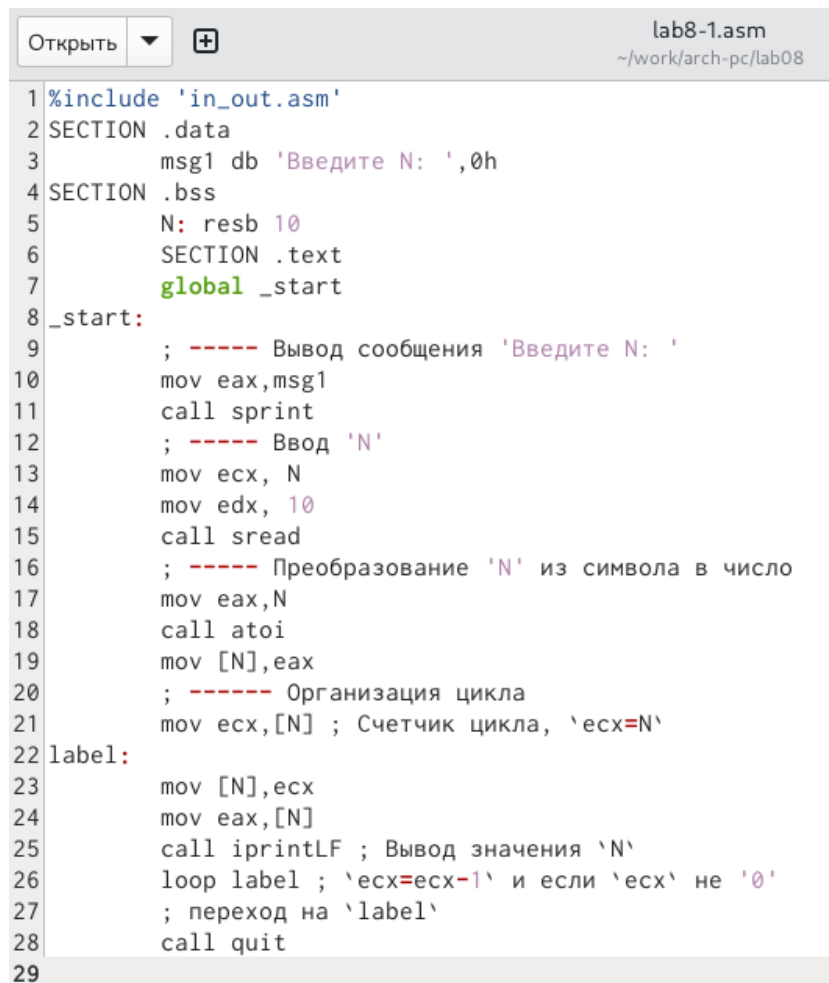
4 Выполнение лабораторной работы

1. Создал каталог для программ лабораторной работы № 8, перешёл в него и создал файл lab8-1.asm(рис. 4.1).

```
avperegudov@dk8n62 ~ $ mkdir ~/work/arch-pc/lab08  
avperegudov@dk8n62 ~ $ cd ~/work/arch-pc/lab08  
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ touch lab8-1.asm  
avperegudov@dk8n62 ~/work/arch-pc/lab08 $
```

Рис. 4.1: Создание каталога и файла

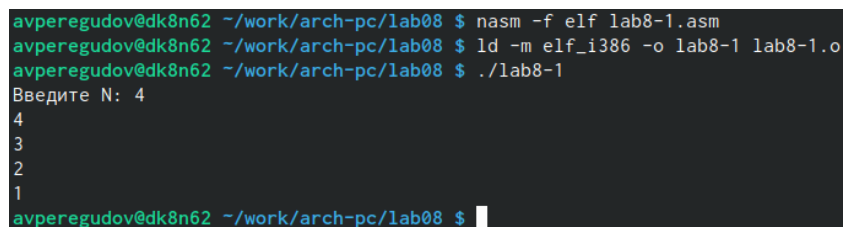
2. Ввёл в файл lab8-1.asm текст программы из листинга 8.1(рис. 4.2).



```
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
29
```

Рис. 4.2: Текст программы

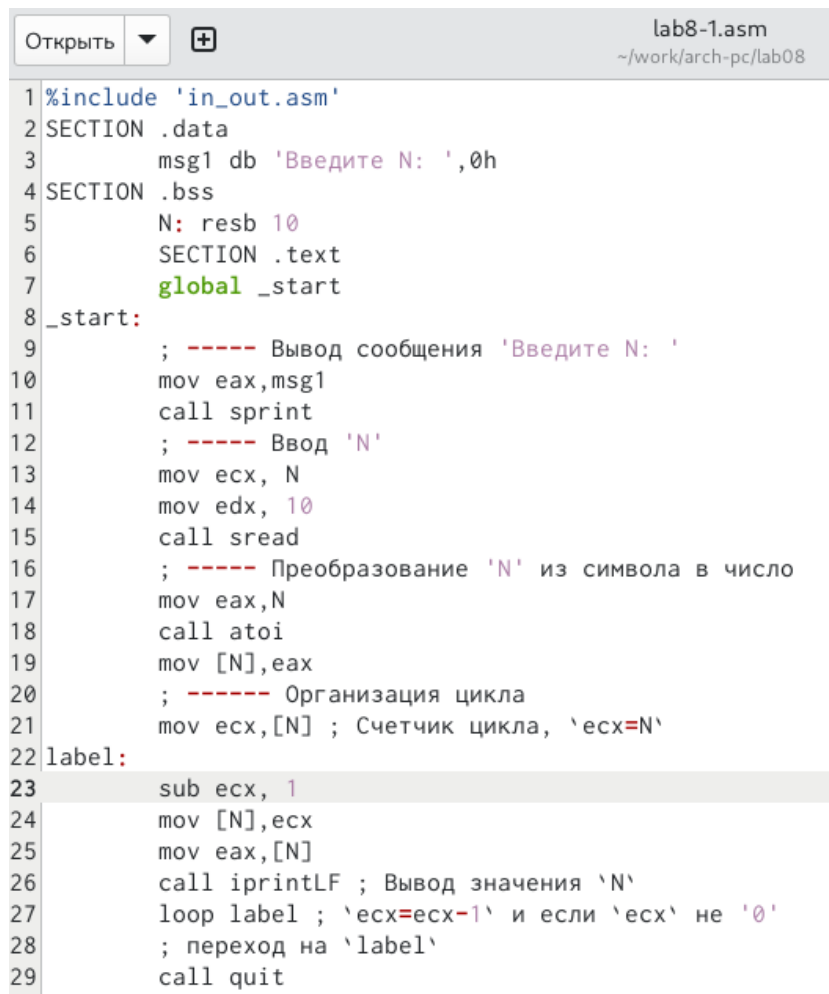
3. Создал исполняемый файл и проверил его работу(рис. 4.3).



```
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 4
4
3
2
1
avperegudov@dk8n62 ~/work/arch-pc/lab08 $
```

Рис. 4.3: Результат работы программы

4. Изменил текст программы добавив изменение значение регистра ecx в цикле(рис. 4.4).

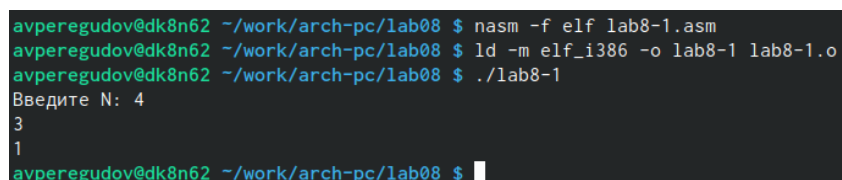


```
lab8-1.asm
~/work/arch-pc/lab08

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    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'
29    call quit
```

Рис. 4.4: Изменённый текст программы

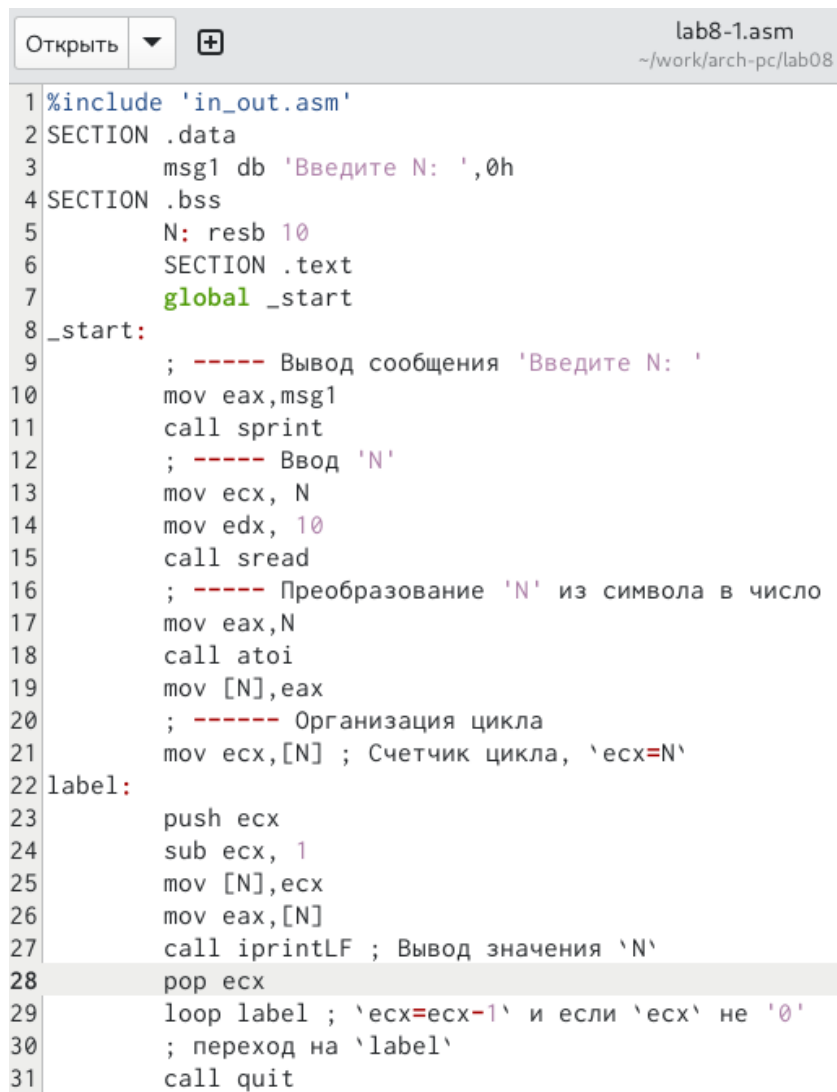
5. Создал исполняемый файл и проверил его работу(рис. 4.5).



```
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 4
3
1
avperegudov@dk8n62 ~/work/arch-pc/lab08 $
```

Рис. 4.5: Результат работы программы

6. Внёс изменения в текст программы добавив команды push и pop(рис. 4.6).

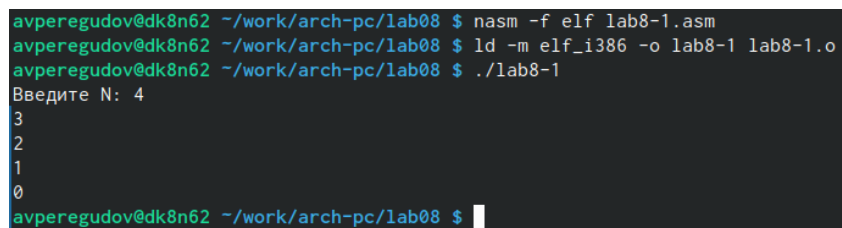


```
lab8-1.asm
~/work/arch-pc/lab08

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    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'
30    ; переход на 'label'
31    call quit
```

Рис. 4.6: Изменённый текст программы

7. Создал исполняемый файл и проверил его работу(рис. 4.7).



```
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 4
3
2
1
0
avperegudov@dk8n62 ~/work/arch-pc/lab08 $
```

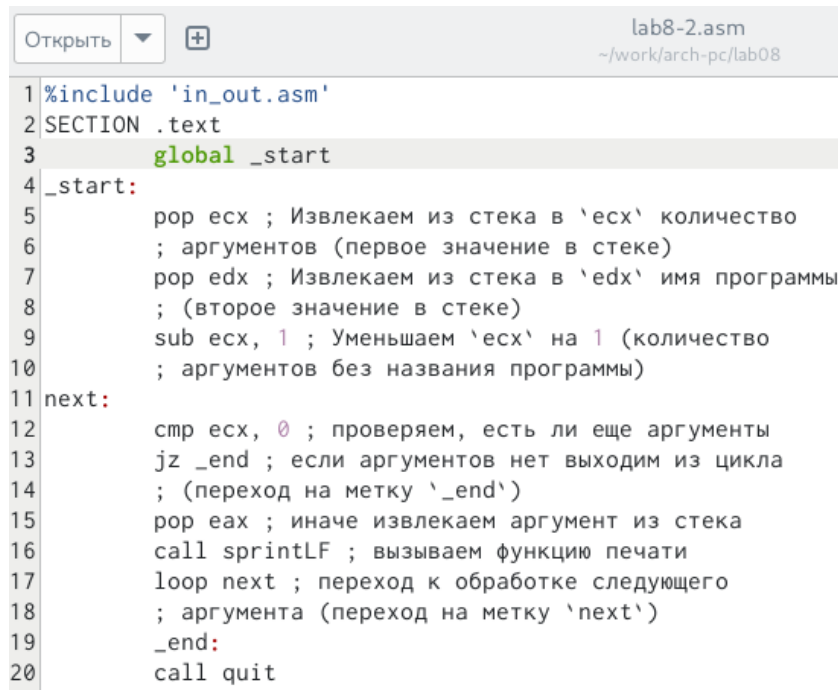
Рис. 4.7: Результат работы программы

8. Создал файл lab8-2.asm в каталоге ~/work/arch-pc/lab08(рис. 4.8).

```
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ touch lab8-2.asm
avperegudov@dk8n62 ~/work/arch-pc/lab08 $
```

Рис. 4.8: Создание файла

9. Ввёл в файл lab8-2.asm текст программы из листинга 8.2(рис. 4.9).



```
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 sprintf ; вызываем функцию печати
17     loop next ; переход к обработке следующего
18             ; аргумента (переход на метку 'next')
19     _end:
20     call quit
```

Рис. 4.9: Текст программы

10. Создал исполняемый файл и проверил его работу, указав аргументы: аргумент1, аргумент, 2, 'аргумент 3'(рис. 4.10).

```
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ nasm -f elf lab8-2.asm
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-2 lab8-2.o
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ ./lab8-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
avperegudov@dk8n62 ~/work/arch-pc/lab08 $
```

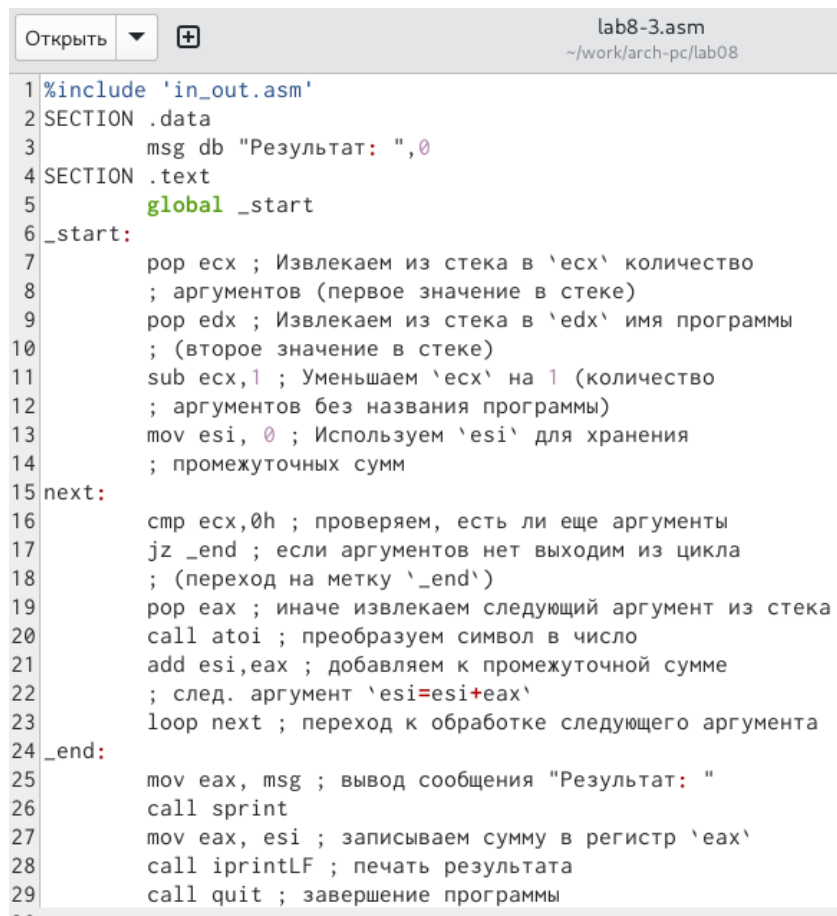
Рис. 4.10: Результат работы программы

11. Создал файл lab8-3.asm в каталоге ~/work/arch-pc/lab08(рис. 4.11).

```
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ touch lab8-3.asm
avperegudov@dk8n62 ~/work/arch-pc/lab08 $
```

Рис. 4.11: Создание файла

12. Ввёл в файл lab8-3.asm текст программы из листинга 8.3(рис. 4.12).



```
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. 4.12: Текст программы

13. Создал исполняемый файл и проверил его работу(рис. 4.13).

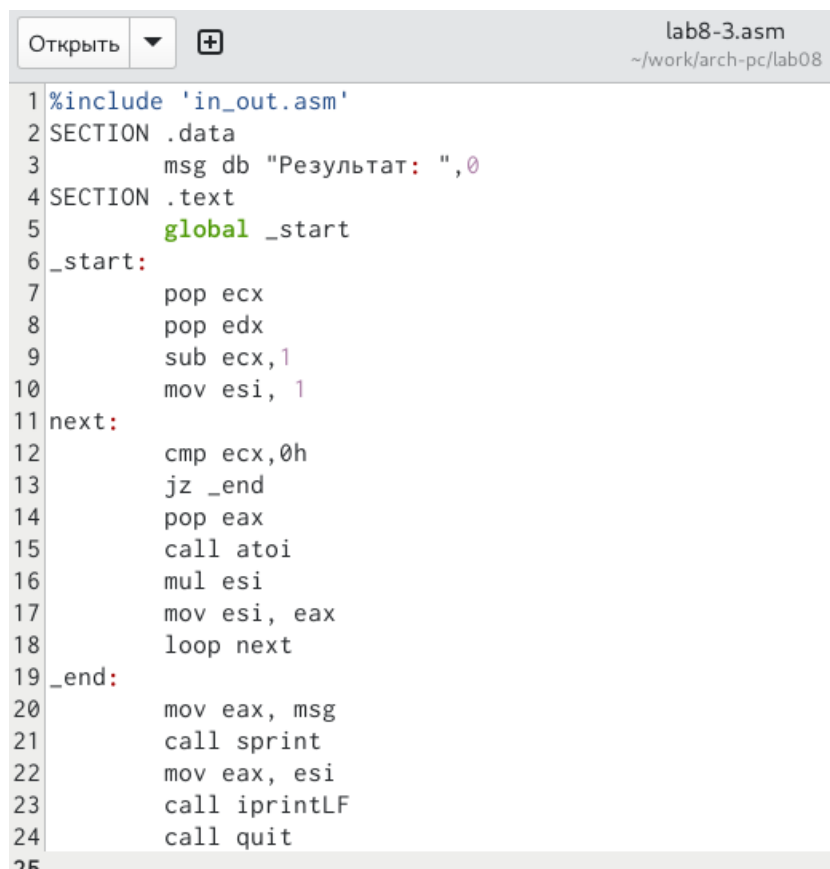
```

avperegudov@dk8n62 ~/work/arch-pc/lab08 $ nasm -f elf lab8-3.asm
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o main lab8-3.o
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ ./main 12 13 7 10 5
Результат: 47
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ █

```

Рис. 4.13: Результат работы программы

14. Изменил текст программы из листинга 8.3 для вычисления произведения аргументов командной строки(рис. 4.14).



```

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

```

Рис. 4.14: Изменённый текст программы

15. Создал исполняемый файл и проверил его работу(рис. 4.15).

```

avperegudov@dk8n62 ~/work/arch-pc/lab08 $ nasm -f elf lab8-3.asm
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o main lab8-3.o
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ ./main 3 4 5
Результат: 60
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ █

```

Рис. 4.15: Результат работы программы

16. Создал файл task.asm(рис. 4.16).

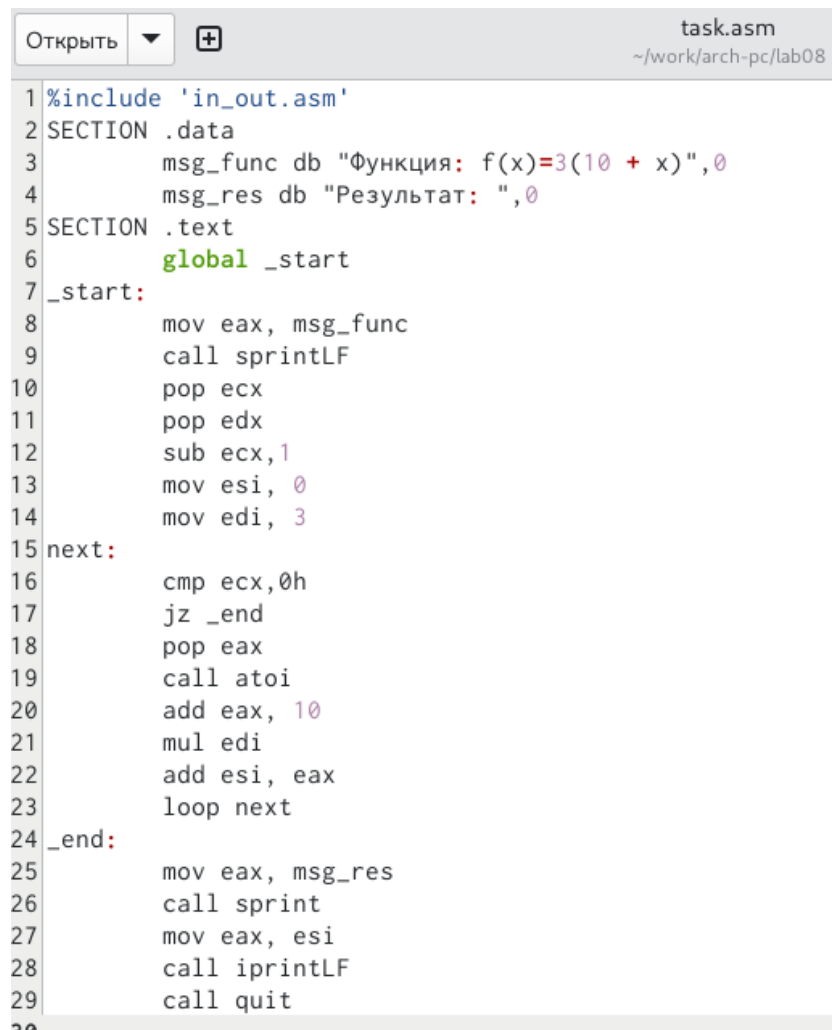
```

avperegudov@dk8n62 ~/work/arch-pc/lab08 $ touch task.asm
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ █

```

Рис. 4.16: Создание файла

17. Ввёл в файл task.asm текст программы, которая находит сумму значений функции $\text{X}(\text{X})$ 20 варианта(рис. 4.17).



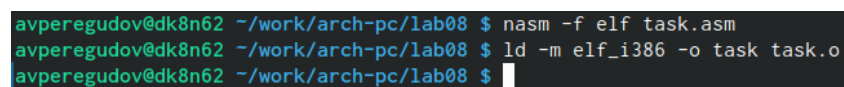
```

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

```

Рис. 4.17: Текст программы

18. Создал исполняемый файл(рис. 4.18).



```

avperegudov@dk8n62 ~/work/arch-pc/lab08 $ nasm -f elf task.asm
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o task task.o
avperegudov@dk8n62 ~/work/arch-pc/lab08 $

```

Рис. 4.18: Трансляция и компоновка

19. Проверил работу программы с разными параметрами(рис. 4.19).


```
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ ./task 1 2 3 4
Функция:  $f(x)=3(10 + x)$ 
Результат: 150
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ ./task 1 1 1 1
Функция:  $f(x)=3(10 + x)$ 
Результат: 132
avperegudov@dk8n62 ~/work/arch-pc/lab08 $ ./task 1 9 8 3 2
Функция:  $f(x)=3(10 + x)$ 
Результат: 219
avperegudov@dk8n62 ~/work/arch-pc/lab08 $
```

Рис. 4.19: Результат работы программы

5 Выводы

В этой лабораторной работе были приобретены навыки написания программ с использованием циклов и обработкой аргументов командной строки.

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.