

Шаблон отчёта по лабораторной работе

Арифметические операции в NASM

Комкова Виктория Руслановна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	15
	Список литературы	16

Список иллюстраций

4.1	Запуск lab6-1	8
4.2	Исправление lab6-1	8
4.3	Запуск исправленного lab6-1	9
4.4	Заполнение созданного lab6-2	9
4.5	Запуск lab6-2	9
4.6	Изменение в lab6-2	10
4.7	Запуск исправленной lab6-2	10
4.8	Изменение функции iprintLF в lab6-2	11
4.9	Запуск измененной lab6-2	11
4.10	Создание lab6-3	11
4.11	Запуск lab6-3	12
4.12	Запуск измененной lab6-3	12
4.13	Заполнение variant.asm	13
4.14	Запуск variant.asm	13

Список таблиц

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

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

1. Создать каталог для программ лабораторной работы № 6.
2. Создать файл lab6-1.asm и ввести в файл lab6-1.asm текст программы из листинга 6.1. Создать исполняемый файл и запустить его.
3. Исправить текста программы (Листинг 6.1), создать исполняемый файл и запустить его.
4. Создать файл lab6-2.asm и ввести в него текст программы из листинга 6.2. Исправить текст программы (Листинг 6.2). Создать исполняемый файл и запустите его.
5. Заменить функцию `iprintLF` на `iprint`. Создать исполняемый файл и запустить его.
6. Создать файл lab6-3.asm. Ввести текст программы из листинга 6.3.
7. Изменить текст программы для вычисления выражения $\pi(\pi) = (4 \cdot 6 + 2)/5$. Создать исполняемый файл и проверить его работу.
8. Создать файл variant.asm в каталоге. Ввести в файл variant.asm текст программы из 6.4 листинга. Создать исполняемый файл и запустить его.
9. Самостоятельная работа.
10. Ответы на вопросы.

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 систему
/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 про Unix см. в [1–4].

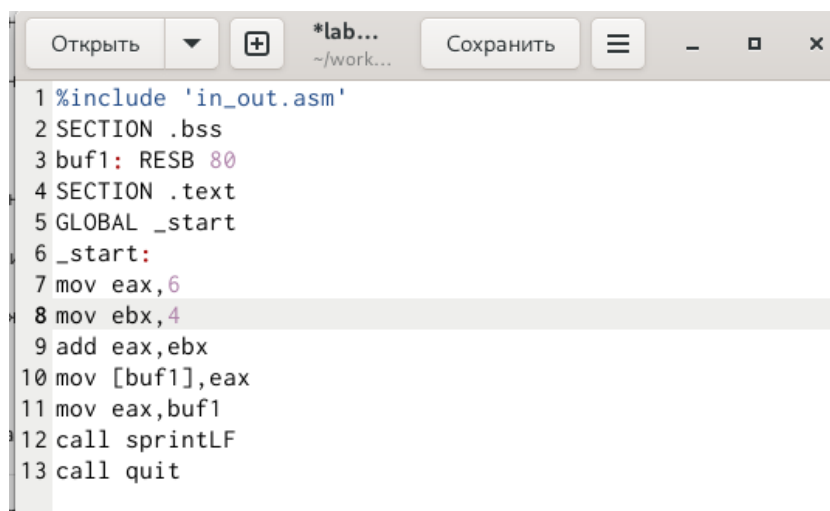
4 Выполнение лабораторной работы

1. Создала каталог для программ лабораторной работы №, перешла в него, создала файл lab6-1.asm. Создала исполняемый файл и запустила его

```
vrkomkova@dk3n35 ~ $ cd ~/work/arch-pc/lab06
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ ./lab6-1
j
```

Рис. 4.1: Запуск lab6-1

2. Изменила текст программы и вместо символов, записала в регистры числа



```
Открыть  *lab...  Сохранить  -  □  ×
~/work...

1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7 mov eax,6
8 mov ebx,4
9 add eax,ebx
10 mov [buf1],eax
11 mov eax,buf1
12 call sprintf
13 call quit
```

Рис. 4.2: Исправление lab6-1

3. Создала исполняемый файл и запустила его.


```

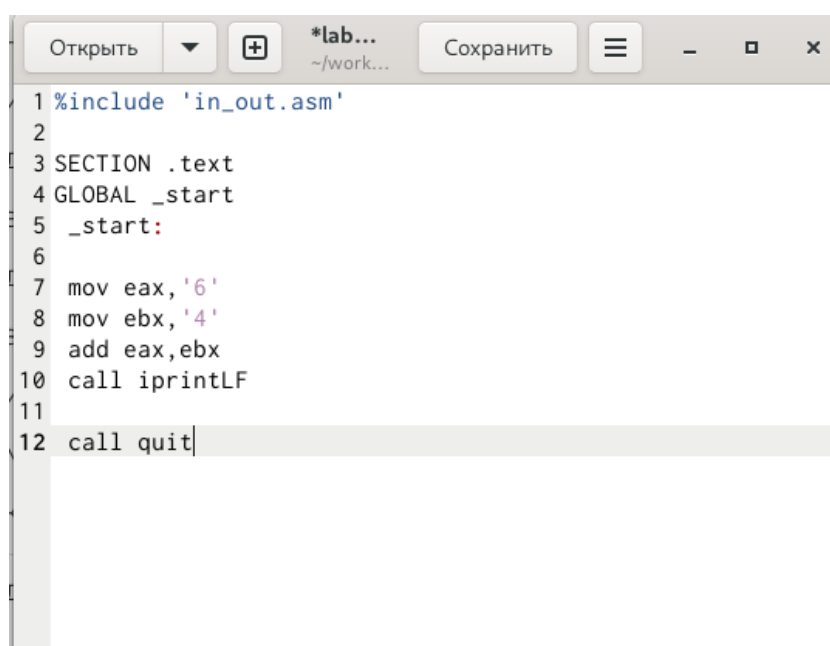
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ ./lab6-1

vrkomkova@dk3n35 ~/work/arch-pc/lab06 $

```

Рис. 4.3: Запуск исправленного lab6-1

4. Создала файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и ввела в него текст программы.



```

1 %include 'in_out.asm'
2
3 SECTION .text
4 GLOBAL _start
5 _start:
6
7 mov eax, '6'
8 mov ebx, '4'
9 add eax, ebx
10 call iprintLF
11
12 call quit

```

Рис. 4.4: Заполнение созданного lab6-2

5. Создала исполняемый файл и запустила его.

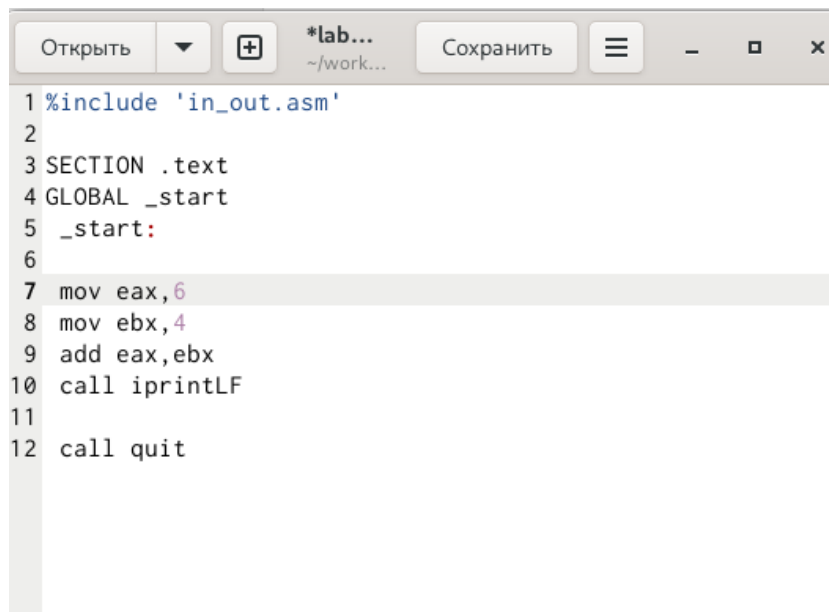
```

vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ ./lab6-2
106

```

Рис. 4.5: Запуск lab6-2

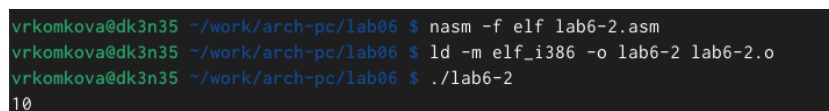
6. Изменила файл lab6-2.asm.



```
1 %include 'in_out.asm'
2
3 SECTION .text
4 GLOBAL _start
5 _start:
6
7 mov eax, 6
8 mov ebx, 4
9 add eax, ebx
10 call iprintLF
11
12 call quit
```

Рис. 4.6: Изменение в lab6-2

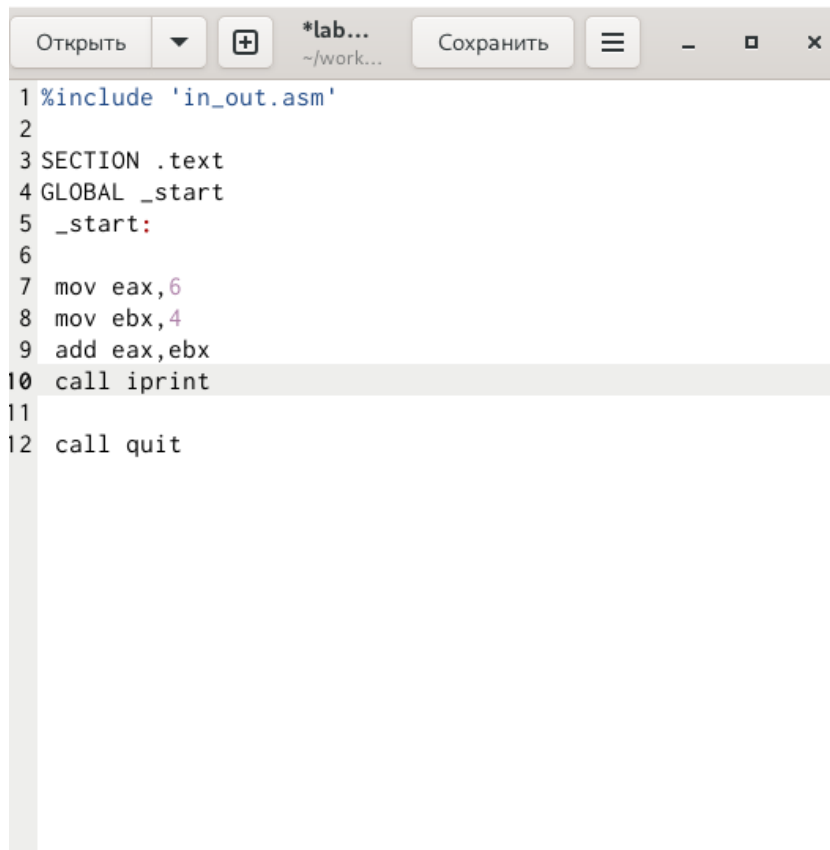
7. Создала исполняемый файл и запустила его.



```
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ ./lab6-2
10
```

Рис. 4.7: Запуск исправленной lab6-2

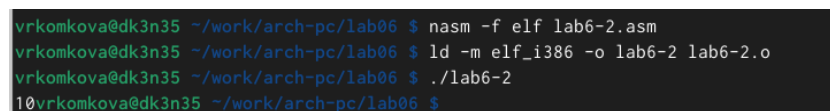
8. Заменяла функцию iprintLF на iprint.



```
Открыть ▼ *lab... ~/work... Сохранить ≡ - □ ×
1 %include 'in_out.asm'
2
3 SECTION .text
4 GLOBAL _start
5 _start:
6
7 mov eax,6
8 mov ebx,4
9 add eax,ebx
10 call iprint
11
12 call quit
```

Рис. 4.8: Изменение функции iprintLF в lab6-2

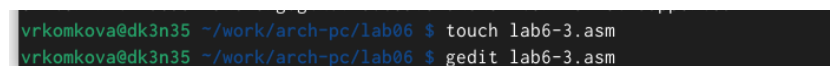
9. Создала исполняемый файл и запустила его.



```
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ ./lab6-2
10vrkomkova@dk3n35 ~/work/arch-pc/lab06 $
```

Рис. 4.9: Запуск измененной lab6-2

10. Создала файл lab6-3.asm в каталоге ~/work/arch-pc/lab06



```
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ touch lab6-3.asm
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ gedit lab6-3.asm
```

Рис. 4.10: Создание lab6-3

11. Заполнила lab6-3.asm,создала исполняемый файл и запустила его.

```
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
```

Рис. 4.11: Запуск lab6-3

12. Изменил текст программы для вычисления выражения $f(x) = (4*6+2)/5$. Создала исполняемый файл и запустила его.

```
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
```

Рис. 4.12: Запуск измененной lab6-3

13. Создала файл `variant.asm` в каталоге `~/work/arch-pc/lab06`. Заполнила `variant.asm` и создала исполняемый файл.

```

1 %include 'in_out.asm'
2
3 SECTION .data
4 func: DB "Формула (5+X)^2 - 3", 0
5 msg: DB "Введите x: ", 0
6 res: DB "Результат: ", 0
7
8 SECTION .bss
9 x: RESB 80 ; Буфер для ввода x
10 result: RESB 80 ; Буфер для хранения результата
11
12 SECTION .text
13 GLOBAL _start
14
15 _start:
16 ; Выводим название формулы
17 mov eax, func
18 call sprintLF
19
20 ; Запрашиваем ввод x
21 mov eax, msg
22 call sprintLF
23
24 ; Читаем значение x
25 mov ecx, x
26 mov edx, 80
27 call sread
28
29 ; Преобразуем строку в число
30 mov eax, x
31 call atoi
32
33 ; Выполняем вычисления: (5 + x)^2 - 3
34 add eax, 5 ; x + 5
35 imul eax, eax ; (x + 5)^2
36 sub eax, 3 ; (x + 5)^2 - 3
37
38 ; Сохраняем результат для вывода
39 mov [result], eax
40
41 ; Выводим результат

```

Рис. 4.13: Заполнение variant.asm

14. Запустила исполняемый файл variant.asm.

```

vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ ./variant
Формула (5+X)^2 - 3
Введите x:
5
Результат: 97
vrkomkova@dk3n35 ~/work/arch-pc/lab06 $ ./variant
Формула (5+X)^2 - 3
Введите x:
1
Результат: 33

```

Рис. 4.14: Запуск variant.asm

Ответы на вопросы: 1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения 'Ваш вариант: '? `mov eax,rem call sprint` 2. Для чего используется следующие инструкции? `mov ecx, x mov edx, 80 call sread` для получения данных с клавиатуры 3. Для чего используется инструкция "call atoi"? для преобразования ASCII кода в число 4. Какие строки листинга 6.4 отвечают за вычисления варианта? `xor edx,edx mov ebx,20 div ebx inc edx` 5. В какой регистр записывается остаток от деления при выполнении инструкции "div ebx"? В `edx` 6. Для чего используется инструкция "inc edx"? Увеличение `edx` на 1. 7. Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений? `mov eax,edx call iprintLF`

5 Выводы

Мне удалось освоить арифметические инструкции языка ассемблера в NASM.

Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.
2. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.