

Лабораторная работа № 6

Архитектура вычислительных систем

Касымова Элина

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	8
4.2	Изменение	8
4.3	Копирование	9
4.4	Вывод	9
4.5	Повтор изменения	10
4.6	lab6-2.asm	10
4.7	Изменение	11
4.8	Вывод	11
4.9	Замена	12
4.10	Вывод	12
4.11	lab6-3.asm	12
4.12	Изменение	13
4.13	Вывод	13
4.14	Изменение	14
4.15	Вывод	14
4.16	Создание	14
4.17	Изменение	15
4.18	Вывод	15
4.19	Создание	16
4.20	Изменение	16
4.21	Вывод	17

Список таблиц

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

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

Цель данного шаблона — максимально упростить подготовку отчётов по лабораторным работам. Модифицируя данный шаблон, студенты смогут без труда подготовить отчёт по лабораторным работам, а также познакомиться с основными возможностями разметки Markdown.

2 Задание

Написать программу вычисления выражения $y = f(x)$. Программа должна выводить выражение для вычисления, выводить запрос на ввод значения x , вычислять заданное выражение в зависимости от введенного x , выводить результат вычислений. Вид функции $f(x)$ выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создайте исполняемый файл и проверьте его работу для значений x_1 и x_2 из 6.3.

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 об Unix см. в [1–6].

4 Выполнение лабораторной работы

1)Создаю каталог для программ лабораторной работы № 6, перехожу в него и создаю файл lab6-1.asm.

```
ekasihmova@dk3n38 ~/work/arch-pc $ mkdir ~/work/arch-pc/lab06
ekasihmova@dk3n38 ~/work/arch-pc $ cd lab06
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ touch lab6-1.asm
```

Рис. 4.1: Создание

2)Ввожу в файл lab6-1.asm текст программы из листинга.

```
%include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, '6'
mov ebx, '4'
add eax, ebx
mov [buf1], eax
mov eax, buf1
call sprintLF
call quit
```

Рис. 4.2: Изменение

3)Копирую файл in_out.

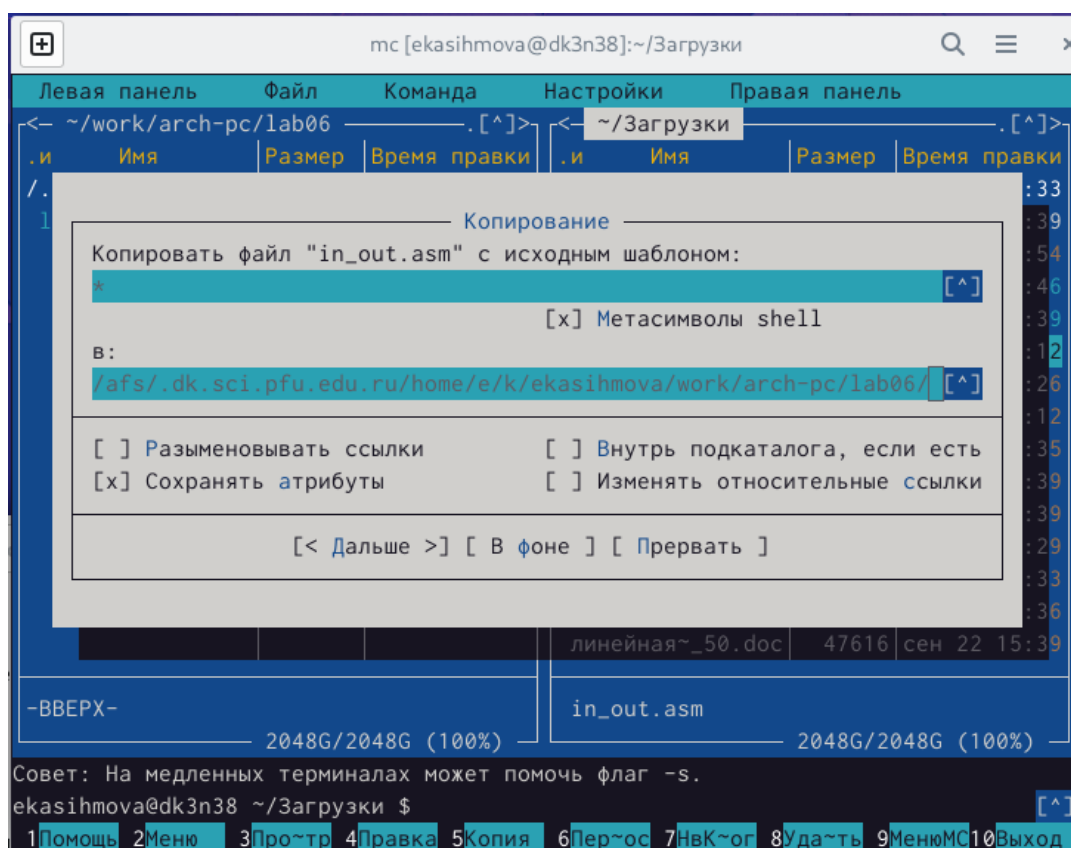


Рис. 4.3: Копирование

4) Создаю исполняемый файл и запускаю его.

```
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ ./lab6-1
j
```

Рис. 4.4: Вывод

5) Далее изменяю текст программы и вместо символов, записываю в регистры числа. Исправляю текст программы.

```
%include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintf
call quit
```

Сохранить изменённый буфер? ☐ Да ☒ Нет

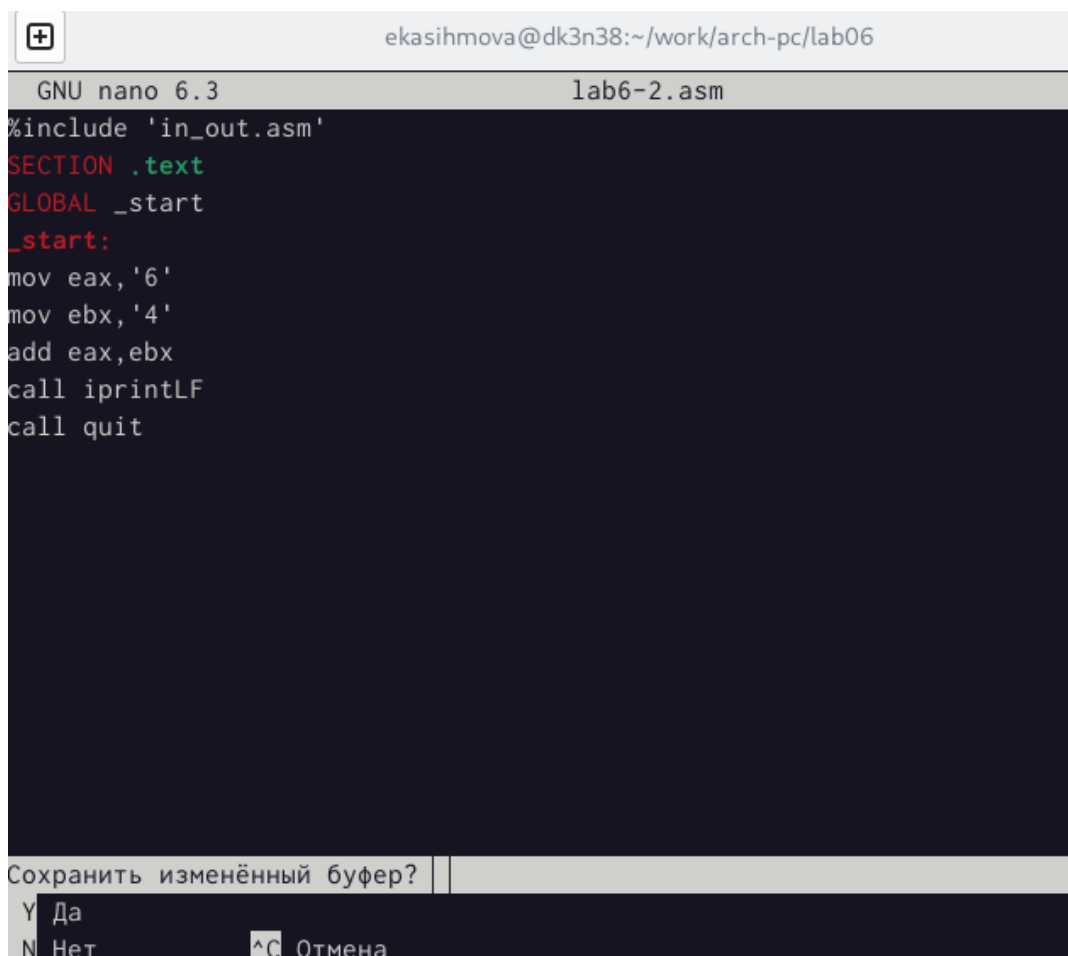
Рис. 4.5: Повтор изменения

6) Создаю файл lab6-2.asm в каталоге ~/work/arch-pc/lab06.

```
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ touch lab6-2.asm
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ nano lab6-2.asm
```

Рис. 4.6: lab6-2.asm

7) Ввожу в него текст программы из листинга.

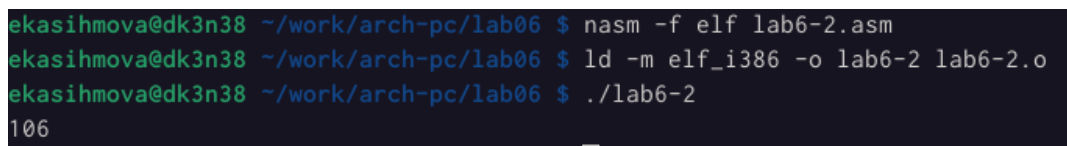


```
ekasihmova@dk3n38:~/work/arch-pc/lab06
GNU nano 6.3 lab6-2.asm
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax, '6'
mov ebx, '4'
add eax, ebx
call iprintLF
call quit

Сохранить изменённый буфер? |
Y Да
N Нет ^C Отмена
```

Рис. 4.7: Изменение

8)Создаю исполняемый файл и запускаю его.



```
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ ./lab6-2
106
```

Рис. 4.8: Вывод

9)Аналогично предыдущему примеру изменяю символы на числа. Заменяю строки.

```

%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit

```

Рис. 4.9: Замена

10) Создаю исполняемый файл и запускаю его. При исполнении программы выводит результат 10.

```

ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ nano lab6-2.asm
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ ./lab6-2
10

```

Рис. 4.10: Вывод

11) В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения $\text{X}(\text{X}) = (5 \times 2 + 3)/3$. Создаю файл lab6-3.asm в каталоге ~/work/arch-pc/lab06.

```

ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ touch lab6-3.asm

```

Рис. 4.11: lab6-3.asm

12) Внимательно изучив текст программы из листинга, ввожу в lab6-3.asm.

```

; ---- Вычисление выражения
mov eax,5 ; EAX=5
mov ebx,2 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,3 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,3 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,edx ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения

```

Рис. 4.12: Изменение

13) Создаю исполняемый файл и запускаю его. Результат работы программы 4 и остаток от деления 1.

```

ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1

```

Рис. 4.13: Вывод

14) Изменяю текст программы для вычисления выражения $\boxed{x}(\boxed{x}) = (4 \times 6 + 2)/5$.

```

#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,4 ; EAX=5
mov ebx,6 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения

```

Рис. 4.14: Изменение

15)Создаю исполняемый файл и запускаю его. Результат работы программы 5 и остаток от деления 1.

```

ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1

```

Рис. 4.15: Вывод

16)Создаю файл variant.asm в каталоге ~/work/arch-pc/lab06.

```

ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/variant.asm
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ nano variant.asm

```

Рис. 4.16: Создание

17)Внимательно изучив текст программы из листинга, ввожу в файл

variant.asm.

```
mov eax,x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, 'eax=x'
xor edx,edx
mov ebx,20
div ebx
inc edx
mov eax,rem
call sprint
mov eax,edx
call iprintLF
call quit
```

Рис. 4.17: Изменение

18)Создаю исполняемый файл и запускаю его.

```
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ nano variant.asm
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ ./variant
Введите No студенческого билета:
1032224390
Ваш вариант: 11
```

Рис. 4.18: Вывод

Ответы на вопросы:

- 1)mov eax,rem отвечают за вывод на экран сообщения 'Ваш вариант:'.
- 2)nasm mov ecx,x - адрес строки "x" в "ecx", mov edx, 80 - размер , call sread-ввод сообщения с клавиатуры.
- 3)call atoi-функция преобразует ascii-код символа в целое число и запишет результат в регистр eax.
- 4)xor edx,edx- вычисление варианта.
- 5)div ebx-регистр AH.
- 6)inc edx- увеличивает регистр на 1.

7)call iprintLF-вывод на экран.

Самостоятельная работа.

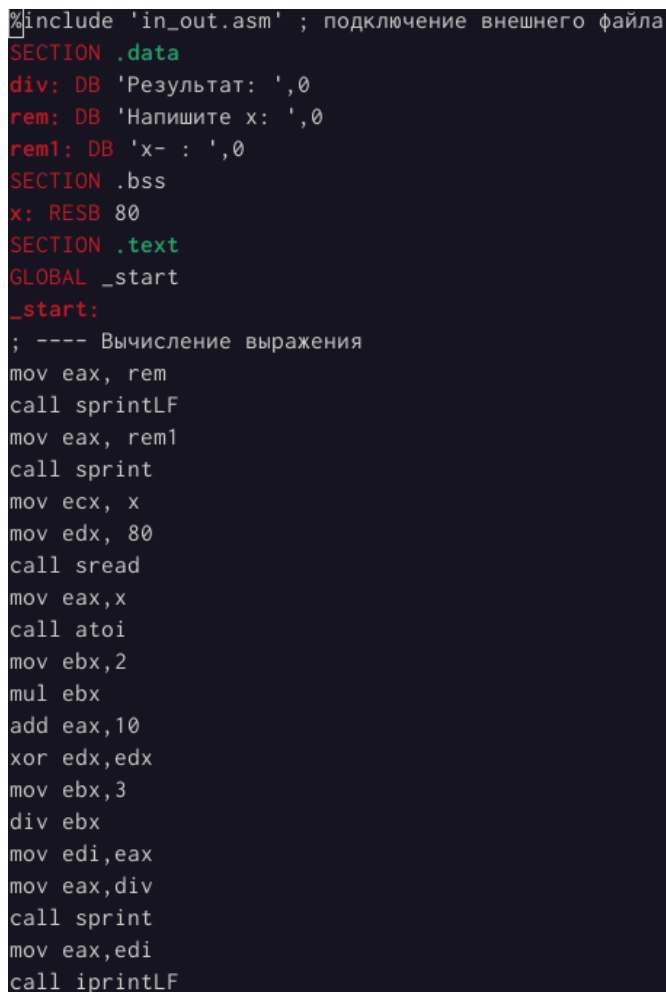
1)Создаю файл lab6-4.asm в каталоге ~/work/arch-pc/lab06.Вычисление арифметического выражения $(10 + 2x)/3$.



```
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ touch lab6-4.asm
```

Рис. 4.19: Создание

2)Ввожу в файл lab6-4.asm текст программы из листинга.



```
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Напишите x: ',0
rem1: DB 'x- : ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax, rem
call sprintLF
mov eax, rem1
call sprint
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
mov ebx, 2
mul ebx
add eax, 10
xor edx, edx
mov ebx, 3
div ebx
mov edi, eax
mov eax, div
call sprint
mov eax, edi
call iprintLF
```

Рис. 4.20: Изменение

3)Создаю исполняемый файл и запускаю его.


```
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ ./lab6-4
Напишите x:
x- : 1
Результат: 4
ekasihmova@dk3n38 ~/work/arch-pc/lab06 $ ./lab6-4
Напишите x:
x- : 10
Результат: 10
```

Рис. 4.21: Вывод

5 Выводы

Проделав данную лабораторную работу, я освоила арифметические инструкции языка ассемблера NASM.

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.