

Шаблон отчёта по лабораторной работе 06

Простейший вариант

Абдуллахи шугофа

Содержание

1	Цель работы	5
2	Порядок выполнения лабораторной работы	6
3	Теоретическое введение	22
4	Выводы	24
	Список литературы	25

Список иллюстраций

2.1	Создала файл	6
2.2	6
2.3	вывода значения регистра еах	7
2.4	Создала исполняемый файл и запустила его.	7
2.5	8
2.6	изменила текст программы	9
2.7	результат изменила на символ	9
2.8	Создала файл lab6-2.asm	10
2.9	вывода значения регистра еах	11
2.10	Создала исполняемый файл и запустила его	11
2.11	изменила символы на числа. Замените строки	12
2.12	Создала исполняемый файл	12
2.13	Заменяла функцию iprintLF на iprint	13
2.14	Создала исполняемый файл и запустила его	13
2.15	Создайте файл	14
2.16	Программа вычисления выражении	14
2.17	Создала исполняемый файл и запустила его	15
2.18	Изменила текст программы	15
2.19	Создала исполняемый файл и проверяла его работу	15
2.20	Создала файл variant.asm	16
2.21	16
2.22	ввела в файл текст	17
2.23	Проверяла результат работы программы	17
2.24	мой вариант	19
2.25	Создала файл	19
2.26	20
2.27	результат	21
3.1	Название рисунка	23

Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . .	22
-----	---	----

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Порядок выполнения лабораторной работы

№ 1: Символьные и численные данные в NASM

1.1: Создайте каталог для программ лабораторной работы № 6, перейдите в него и создайте файл lab6-1.asm:

```
shogofa@shogofa-VirtualBox:~$ mkdir ~/work/arch-pc/lab06
shogofa@shogofa-VirtualBox:~$ cd ~/work/arch-pc/lab06
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$ touch lab6-1.asm
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 2.1: Создала файл

.h	Name	Size	Modify time	.h
UP	..	DIR	Nov 11 20:51	UP
189	lab6-1.asm		Nov 11 21:00	lab6-

Рис. 2.2:

1.2: Листинг 6.1. Программа вывода значения регистра eax

```
%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
```

Рис. 2.3: вывода значения регистра eax

Создайте исполняемый файл и запустите его.

```
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$ ./lab6-1
j
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$ mc
```

Рис. 2.4: Создала исполняемый файл и запустила его.

.h	Name	Size	Modify time	.h
../	UP--DIR	Nov 11 20:51	../	
in_out.asm	3942	Nov 9 10:18	in_out.asm	
*lab6-1	5160	Nov 11 21:05	*lab6-1	
lab6-1.asm	189	Nov 11 21:00	lab6-1.asm	
lab6-1.o	1200	Nov 11 21:05	lab6-1.o	

Рис. 2.5:

1.3: Далее изменим текст программы и вместо символов, запишем в регистры числа. Ис- правьте текст программы (Листинг 6.1) следующим образом: замените строки


```

#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

```

Рис. 2.6: изменила текст программы

```

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

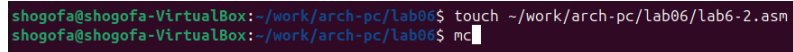
```

Рис. 2.7: результат изменила на символ

1.4: Как отмечалось выше, для работы с числами в файле in_out.asm реализованы подпрограммы для преобразования ASCII символов в числа и обратно.

Преобразуем текст программы из Листинга 6.1 с использованием этих функций.

Создайте файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и введите в него текст программы из листинга 6.2.

A terminal window with a dark background and light green text. The prompt is 'shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06\$'. The first command is 'touch ~/work/arch-pc/lab06/lab6-2.asm' and the second is 'mc', followed by a cursor.

```
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-2.asm
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$ mc
```

Рис. 2.8: Создала файл lab6-2.asm

2. Листинг 6.2. Программа вывода значения регистра eax

```
%include 'in_out.asm'

SECTION .text
GLOBAL _start

_start:
mov    eax, '6'
mov    ebx, '4'
add    eax, ebx
call   iprintLF

call   quit
```

Рис. 2.9: вывода значения регистра eax

Создайте исполняемый файл и запустите его

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

Рис. 2.10: Создала исполняемый файл и запустила его

2.1: Аналогично предыдущему примеру изменим символы на числа. Замените строки

```
%include 'in_out.asm'

SECTION .text
GLOBAL _start

_start:
mov    eax,6
mov    ebx,4
add    eax,ebx
call   iprintLF

call   quit
```

Рис. 2.11: изменила символы на числа. Замените строки

2.2: Создайте исполняемый файл и запустите его. Какой результат будет получен при исполнении программы?

```
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-9.asm
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$ nc
```

Рис. 2.12: Создайла исполняемый файл

Замените функцию `iprintLF` на `iprint`. Создайте исполняемый файл и запустите его. Чем отличается вывод функций `iprintLF` и `iprint`?

```
%include 'in_out.asm'

SECTION .text
GLOBAL _start

_start:
mov    eax, '6'
mov    ebx, '4'
add    eax, ebx
call   iprint

call   quit
```

Рис. 2.13: Заменяла функцию `iprintLF` на `iprint`

```
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-9.asm
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$ ./lab6-2
106
```

Рис. 2.14: Создайла исполняемый файл и запустила его

Но 2: Выполнение арифметических операций в NASM

2.1: В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения $\text{X}(\text{X}) = (5 \times 2 + 3)/3$.

Создайте файл lab6-3.asm в каталоге ~/work/arch-pc/lab06: 2.6 Задание 6:

```
shogofa@shogofa-VirtualBox: ~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-3.asm
```

Рис. 2.15: Создайте файл

Внимательно изучите текст программы из листинга 6.3 и введите в lab6-3.asm.

3. Листинг 6.3. Программа вычисления выражения $\text{X}(\text{X}) = (5 \times 2 + 3)/3$

```
;-----
; Программа вычисления выражения
;-----

%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,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
```

Рис. 2.16: Программа вычисления выражения

Создайте исполняемый файл и запустите его. Результат работы программы должен быть следующим:

user@dk4n31:~\$./lab6-3 Результат: 4 Остаток от деления: 1 user@dk4n31:~\$

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

Рис. 2.17: Создала исполняемый файл и запустила его

Измените текст программы для вычисления выражения $\text{X}(\text{X}) = (4 \times 6 + 2)/5$.
Создайте исполняемый файл и проверьте его работу.

```
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

SECTION .text
GLOBAL _start
_start:

; ---- Вычисление выражения
mov eax,4      ; EAX=4
mov ebx,6      ; EBX=6
mul ebx        ; EAX=EAX*EBX
add eax,2      ; EAX=EAX+2
xor edx,edx    ; обнуляем EDX для корректной работы div
mov ebx,5      ; EBX=5
div ebx        ; EAX=EAX/5, 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 iprintfLF ; из 'edi' в виде символов

mov eax,rem    ; вызов подпрограммы печати
call sprint    ; сообщения 'Остаток от деления: '
```

Рис. 2.18: Изменила текст программы

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

Рис. 2.19: Создала исполняемый файл и проверяла его работу

3.1: Создайте файл variant.asm в каталоге ~/work/arch-pc/lab06:

```
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$ mc
```

Рис. 2.20: Создайла файл variant.asm



variant.asm	0	Nov 16 09:42	variant.asm
-------------	---	--------------	-------------

Рис. 2.21:

Внимательно изучите текст программы из листинга 6.4 и введите в файл variant.asm.

4. Листинг 6.4. Программа вычисления вычисления варианта задания по номеру студенческого билета


```

;-----
; Программа вычисления варианта
;-----

%include 'in_out.asm'

SECTION .data
msg: DB 'Введите No студенческого билета: ',0
rem: DB 'Ваш вариант: ',0

SECTION .bss
x:   RESB 80

SECTION .text
GLOBAL _start
_start:

mov  eax, msg
call sprintLF

mov  ecx, x
mov  edx, 80
call sread

mov  eax, x           ; вызов подпрограммы преобразования
call atoi             ; ASCII кода в число, `eax=x`

xor  edx, edx

```

Рис. 2.22: ввела в файл текст

```

shogofa@shogofa-VirtualBox:~/work/arch-pc/Lab06$ nasm -f elf variant.asm
shogofa@shogofa-VirtualBox:~/work/arch-pc/Lab06$ ld -m elf_i386 -o variant variant.o
shogofa@shogofa-VirtualBox:~/work/arch-pc/Lab06$ ./variant
Введите No студенческого билета:
1032225505
Ваш вариант: 6

```

Рис. 2.23: Проверила результат работы программы

Создайте исполняемый файл и запустите его. Проверьте результат работы программы вычислив номер варианта аналитически.

Включите в отчет по выполнению лабораторной работы ответы на следующие вопросы:

1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения 'Ваш вариант:'?

Строка `mov eax, ret` и строка `call sprint` отвечают за вывод на экран сообщения “мой вариант:”.

2. Для чего используются следующие инструкции? `mov ecx, x` `mov edx, 80` `call sread`

Инструкции `mov ecx, x` и `mov edx, 80` используются для подготовки аргументов перед вызовом подпрограммы `sread`. `mov ecx, x` загружает адрес переменной `x` в регистр `ecx`, который будет использован в качестве аргумента для функции `sread`. `mov edx, 80` загружает значение 80 в регистр `edx`, указывая функции `sread`, сколько байт нужно прочитать.

3. Для чего используется инструкция “`call atoi`”?

Инструкция `call atoi` используется для вызова подпрограммы `atoi`, которая преобразует ASCII-код, хранящийся в регистре `eax`, в число. Результат преобразования сохраняется в регистре `eax`.

4. Какие строки листинга 6.4 отвечают за вычисления варианта?

Строка `xor edx, edx` и строка `mov ebx, 20` отвечают за подготовку значений перед выполнением вычислений варианта. `xor edx, edx` устанавливает регистр `edx` в ноль, а `mov ebx, 20` загружает значение 20 в регистр `ebx`, которое будет использовано для деления.

5. В какой регистр записывается остаток от деления при выполнении инструкции “`div ebx`”?

Остаток от деления при выполнении инструкции `div ebx` записывается в регистр `edx`.

6. Для чего используется инструкция “`inc edx`”?

Инструкция `inc edx` используется для увеличения значения в регистре `edx` на единицу. В данном случае, она увеличивает значение остатка от деления на 1.

7. Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений?

Строка `mov eax, edx` и строка `call iprintLF` отвечают за вывод на экран результата вычислений. `mov eax, edx` загружает значение в регистр `eax`, чтобы передать его в функцию `iprintLF`, которая выводит значение на экран с новой строкой.

4. Задание для самостоятельной работы

Написать программу вычисления выражения $x = x(x)$. 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 $x(x)$ 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

$$6 \qquad x^3/2 + 1 \qquad 2 \qquad 5$$

Рис. 2.24: мой вариант

```
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/c.работы.asm
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$ mc
```

Рис. 2.25: Создала файл

```

;-----
; Программа вычисления варианта
;-----

#include 'in_out.asm'

SECTION .data
msg: DB 'Введите число: ',0
rem: DB 'Результат: ',0

SECTION .bss
x:   RESB 80

SECTION .text
GLOBAL _start
_start:

mov  eax,msg
call sprintf

mov  ecx,x
mov  edx,80
call sread

mov  eax,x          ; вызов подпрограммы преобразования
call atoi           ; ASCII кода в число, `eax=x`

xor  ebx,ebx
mov  ebx,eax
mul  ebx
mul  ebx
mov  ebx,2
div  ebx
add  eax,1

mov  edi,eax

mov  eax,rem
call sprint
mov  eax,edi
call iprintf

```

Рис. 2.26:

```
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$ ./с.работы
Введите число:
2
Результат: 5
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$ ./с.работы
Введите число:
5
Результат: 63
shogofa@shogofa-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 2.27: результат

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].

Выполнение лабораторной работы

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 3.1).

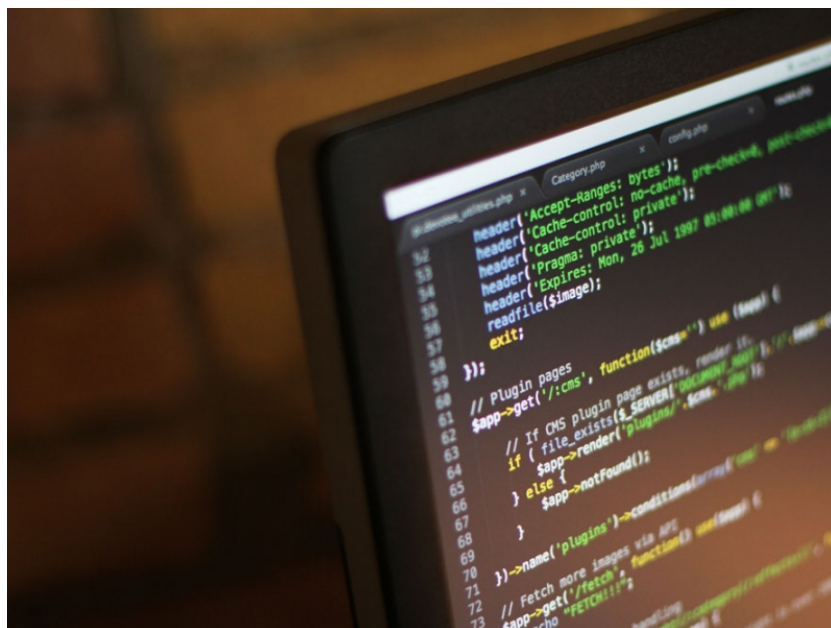


Рис. 3.1: Название рисунка

4 Выводы

Здесь кратко описываются итоги проделанной работы.

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.