

Отчёта по лабораторной работе №6

Дисциплина: Архитектура компьютера

Хоюгбан Ганчыыр Анатолевич

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7

Список иллюстраций

3.1	Текст программы lab6-1.asm	8
3.2	Исполнение программы lab6-1.asm	9
3.3	Текст измененной программы lab6-1.asm	10
3.4	Исполнение измененного файла lab6-1.asm	10
3.5	Текст программы lab6-2.asm	11
3.6	Исполнение файла lab6-2.asm	11
3.7	Текст измененной программы lab6-2.asm	12
3.8	Исполнение измененного файла lab6-2.asm	12
3.9	Создание файла lab6-3.asm	12
3.10	Текст программы для выражения $f(x) = (5 * 2 + 3)/3$	13
3.11	Исполнение файла lab6-3.asm	14
3.12	Текст программы для выражения $f(x) = (4 * 6 + 2)/5$	14
3.13	Исполнение измененного файла lab6-3.asm	15
3.14	Текст программы файла variant.asm	15
3.15	Исполнение файла variant.asm	16
3.16	Создание файла lab6-4.asm	16
3.17	Текст программы для выражения $f(x) = x^3 * 1/3 + 21$	17
3.18	Первое исполнение файла lab6-4.asm	17
3.19	Второе исполнение файла lab6-4.asm	18

Список таблиц

2.1	Описание некоторых каталогов файловой системы GNU Linux . .	6
-----	---	---

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы. Например, в табл. 2.1 приведено краткое описание стандартных каталогов Unix.

Таблица 2.1: 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].

3 Выполнение лабораторной работы

(рис. 3.1) {#fig:001 width=70%}

Для начала я перешел в свой каталог, который имеет такой путь: ~/work/arch-rc/lab06. В этом каталоге я создал файл lab6-1.asm. В него я ввел текст программы на рисунке(рис. 3.1)

```
xx%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. 3.1: Текст программы lab6-1.asm

Затем я оттранслировал программу в объектный файл, отправил на компоновку и провел исполнение этого файла в действии. На выходе я получил просто j(рис. 3.2)


```

gakhoyugban@dk8n60 ~/work/arch-pc $ cd lab06
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ ls
in_out.asm  lab6-1.asm  lab6-1.asm.save
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-1
j
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ ls
in_out.asm  lab6-1  lab6-1.asm  lab6-1.asm.save  lab6-1.o
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ 

```

Рис. 3.2: Исполнение программы lab6-1.asm

Я изменил код программы в том же файле, как было сказано по условию. То есть я убрал кавычки у 4 и 6, что я показываю на рисунке(рис. 3.3)

```
%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. 3.3: Текст измененной программы lab6-1.asm

Я оттранслировал файл в объектный, отправил его на компоновку и также отправил на исполнение, где получил вместо ответа пустоту(рис. 3.4)

```
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-1

gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $
```

Рис. 3.4: Исполнение измененного файла lab6-1.asm

Я создал новый файл lab6-2.asm. В него я ввел текст для программы вывода

значения регистра еах, что я показана на рисунке(рис. 3.5)

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

Рис. 3.5: Текст программы lab6-2.asm

Я оттранслировал текст программы файла lab6-2.asm в объектный файл, от-
правил на компоновку и провел исполнение программы, где получил 106(рис.
3.6)

```
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-2
106
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $
```

Рис. 3.6: Исполнение файла lab6-2.asm

Затем, как было сказано по условию, я изменил текст программы, то есть убрал
кавычки у 6 и 4. На рисунке текст программы(рис. 3.7)

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

Рис. 3.7: Текст измененной программы lab6-2.asm

Я оттранслировал файл в объектный, отправил его на компоновку и отправил на исполнение, где получил ответ просто 10(рис. 3.8)

```
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-2
10
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $
```

Рис. 3.8: Исполнение измененного файла lab6-2.asm

Для начала я создал файл lab6-3.asm(рис. 3.9)

```
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ touch lab6-3.asm
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ ls
in_out.asm  lab6-1  lab6-1.asm  lab6-1.asm.save  lab6-1.o  lab6-2  lab6-2.asm  lab6-2.o  lab6-3.asm
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $
```

Рис. 3.9: Создание файла lab6-3.asm

По условию лабораторной работы я переписал программу в мой файл lab6-3.asm программу вычисления выражения $f(x) = (5 * 2 + 3)/3$ (рис. 3.10)

```
;-----  
; 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  
call sprint ; сообщения 'Результат: '  
mov eax,edi ; вызов подпрограммы печати значения  
call iprintLF ; из 'edi' в виде символов  
mov eax,rem ; 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. 3.10: Текст программы для выражения $f(x) = (5 * 2 + 3)/3$

Я оттранслировал файл lab6-3.asm в объектный файл, првоел компоновку и отправил на исполнение, откуда и получил правильные ответы из условий лабораторной работы(рис. 3.11)

```

gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ mc
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $

```

Рис. 3.11: Исполнение файла lab6-3.asm

Затем мне надо было написать текст для программы вычисления выражения $f(x) = (4 * 6 + 2)/5$. Код программы на рисунке(рис. 3.12)

```

#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=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 iprintLF ; из 'edi' в виде символов
mov eax,rem ; 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. 3.12: Текст программы для выражения $f(x) = (4 * 6 + 2)/5$

Измененный файл lab6-3.asm я оттранслировал в объектный файл, провел его компоновку и отправил на исполнение, откуда получил правильные ответы из условия лабораторной работы(рис. 3.13)

```

gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ mc

gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ touch variant.asm
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ mc

gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ █

```

Рис. 3.13: Исполнение измененного файла lab6-3.asm

Для 7 пункта я создал новый файл variant.asm. Затем я переписал предложенный текст программы, который я продемонстрирую вам на следующем рисунке(рис. 3.14)

```

%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 sprintf
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
mov ebx, 20
div ebx
inc edx
mov eax, rem
call sprintf
mov eax, edx
call iprintLF
call quit

```

Рис. 3.14: Текст программы файла variant.asm

Я оттранслировал файл variant.asm в объектный, провел его компоновку и от-

правил на исполнение. Мне предложили ввести мой студенческий, что я сделал. Затем я получил 20 вариант(рис. 3.15)

```
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-2
10
gakhoyugban@dk8n60 ~/work/arch-pc/lab06 $
```

Рис. 3.15: Исполнение файла variant.asm

По условию лабораторной работы меня попросили ответить на 7 вопросов, что я и сделаю:

1)За вывод сообщения “Ваш вариант” отвечают строки кода: `mov eax,rem call sprint` 2)Инструкция `mov esx, x` используется, чтобы положить адрес вводимой строки `x` в регистр `esx` `mov edx, 80` - запись в регистр `edx` длины вводимой строки `call sread` - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры 3)`call atoi` используется для вызова подпрограммы из внешнего файла, которая преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax` 4)За вычисления варианта отвечают строки: `xor edx,edx` ; обнуление `edx` для корректной работы `div mov ebx,20` ; `ebx = 20` `div ebx` ; `eax = eax/20`, `edx` - остаток от деления `inc edx` ; `edx = edx + 1` 5)При выполнении инструкции `div ebx` остаток от деления записывается в регистр `edx` 6)Инструкция `inc edx` увеличивает значение регистра `edx` на 1 7)За вывод на экран результатов вычислений отвечают строки: `mov eax,edx call iprintLF`

#Выполнение самостоятельной работы

Так как по ходу лабораторной работы мне выпал 20 вариант, мне выпадает написать текст для программы вычисления выражения $f(x) = x^3 * 1/3 + 21$. Значит для начала я создам файл `lab6-4.asm`, в котором будет текст моего кода(рис. 3.16)

```
gakhoyugban@dk8n53 ~/work/arch-pc/lab06 $ touch lab6-4.asm
gakhoyugban@dk8n53 ~/work/arch-pc/lab06 $ ls
lab6-1  lab6-1.asm  lab6-1.asm.save  lab6-1.o  lab6-2  lab6-2.asm  lab6-2.o  lab6-3  lab6-3.asm  lab6-3.o  lab6-4.asm  variant  variant.asm  variant.o
```

Рис. 3.16: Создание файла lab6-4.asm

Для успешного получения правильного решения, я ввел вот такой текст программы(рис. 3.17)

```
include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; секция инициализированных данных
msg: DB 'Введите значение переменной x: ',0
rem: DB 'Результат: ',0
SECTION .bss ; секция не инициализированных данных
x: RESB 80 ; Переменная, значение к-рой будем вводить с клавиатуры, выделенный размер - 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
; ---- Вычисление выражения
mov eax, msg ; запись адреса выводимого сообщения в eax
call sprint ; вызов подпрограммы печати сообщения
mov ecx, x ; запись адреса переменной в ecx
mov edx, 80 ; запись длины вводимого значения в edx
call sread ; вызов подпрограммы ввода сообщения
mov eax, x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, 'eax=x'
mov ebx, eax ; записать значение eax в регистр ebx
mul ebx ; EAX=EAX*EBX
mul ebx ; EAX=EAX*EBX
mov ebx, 1 ; записать значение 1 в регистр ebx
mul ebx ; EAX=EAX*EBX
mov ebx, 3 ; записать значение 3 в регистр ebx
div ebx ; EAX=EAX/3
add eax, 21 ; eax = eax+21 = x*x*x*1/3+21
mov edi, eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax, rem ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax, edi ; вызов подпрограммы печати значения
call iprint ; из 'edi' в виде символов
call quit ; вызов подпрограммы завершения
```

Рис. 3.17: Текст программы для выражения $f(x) = x^3 * 1/3 + 21$

Я оттранслировал текст файла lab6-4.asm в объектный, провел компоновку и отправил на исполнение. Для начала я ввел значение 1 для моего x, откуда получаю ответ 21, что по моим расчетам верно(рис. 3.18)

```
gakhoyugban@dk5n53 ~/work/arch-pc/lab06 $ nasm -f elf lab6-4.asm
gakhoyugban@dk5n53 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-4 lab6-4.o
gakhoyugban@dk5n53 ~/work/arch-pc/lab06 $ ./lab6-4
Введите значение переменной x: 1
```

Рис. 3.18: Первое исполнение файла lab6-4.asm

Я оттранслировал текст файла lab6-4.asm в объектный, провел компоновку и отправил на исполнение. Теперь я введу 3 для моего значения, откуда получаю ответ 30, что по моим расчетам верно(рис. 3.19)

```

Результат: 21gakhoyugban@dk5n53 ~/work/arch-pc/lab06 $ mc
gakhoyugban@dk5n53 ~/work/arch-pc/lab06 $ nasm -f elf lab6-4.asm
gakhoyugban@dk5n53 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-4 lab6-4.o
gakhoyugban@dk5n53 ~/work/arch-pc/lab06 $ ./lab6-4
Введите значение переменной x: 3
Результат: 30gakhoyugban@dk5n53 ~/work/arch-pc/lab06 $ mc

```

Рис. 3.19: Второе исполнение файла lab6-4.asm

#Вывод

При выполнении данной лабораторной работы я освоил арифметические инструкции языка ассемблера NASM.

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.