

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

дисциплина: Архитектура компьютера

Студент: Армихос Гонзалез Карла

Группа: НКАбд-02-24

МОСКВА

2024 г.

Содержание

1. Цель работы.....	4
2. Задание.....	5
3. Технические введение.....	6
4. Порядок выполнения лабораторной работы.....	8
4.1 Символьные и численные данные в NASM.....	8
4.2 Выполнение арифметических операций в NASM.....	11
5. Задание для самостоятельной работы.....	16
6. Выводы.....	18
7. Список используемой литературы.....	18

Список иллюстраций

рис. 4.1.1 Создание директории.....	8
рис. 4.2 Копи 4.1.2.....	8
рис. 4.1.3 Запустить программу (j).....	9
рис. 4.1.4 Выполнить исправленную программу.....	9
рис. 4.1.5 Исправленный лист.....	9
рис. 4.1.6 Новый файл lab6-2. asm.....	9
рис. 4.1.7 Программа вывода значения регистра eax.....	10
рис 4.1.8 Создайте исполняемый файл и запустите его.....	10
рис 4.1.9 Измененная программа.....	11
рис 4.1.10 Запустить программу.....	11
рис. 4.1.11 Выполнить измененную функцию inprint.....	11
рис. 4.2.2 Новый файл lab6-3.asm.....	11
рис. 4.2.3 Программа для вычисления арифметического выражения.....	12
рис 4.2.4 Результат первой проблемы.....	12
рис 4.2.5 Программа для вычисления арифметического выражения.....	13
рис 4.2.6 Результат второй проблемы.....	13
Рис 4.2.7 Программу расчета варианта.....	14
Рис 4.2.8 Вариант.....	15
рис 5.1 Программа.....	16
рис 5.2 Запустить программу.....	17
рис 5.3 Запустить программу.....	17

1. Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2. Задание

1. Работа с числовыми символами в NASM.
2. Запускать разные программы
3. Выполнение арифметических задач в NASM.
4. Выполнять работу самостоятельно

3. Технические введение

В ассемблере команда сложения называется **ADD**.

В ассемблере команда сложения называется **ADD**. Синтаксис выглядит так:

ADD destination, source

Здесь **destination** — это регистр или память, куда будет записан результат, а **source** — это регистр, память или константа, которые прибавляются к **destination**.

Команда для умножения без знака называется **MUL**. Она используется для умножения операндов, рассматриваемых как положительные числа.

Команда для деления без знака — это **DIV**. Синтаксис деления чисел:

DIV divisor

Здесь **divisor** — это делитель. Делимое обычно находится в регистрах **AX**, **DX:AX**, или в регистрах шириной больше 8 бит.

При умножении двухбайтовых операндов (например, **MUL** с 16-битными значениями) результат записывается в регистры **DX:AX**, где **DX** содержит старшие 16 бит, а **AX** — младшие 16 бит результата.

Арифметические команды с целочисленными операндами и их назначение:

ADD — сложение целых чисел.

SUB — вычитание целых чисел.

MUL — умножение без знака.

IMUL — умножение со знаком.

DIV — деление без знака.

IDIV — деление со знаком.

INC — увеличение на 1 (инкремент).

DEC — уменьшение на 1 (декремент).

Местоположение делимого при целочисленном делении операндов:

Для целочисленного деления без знака (**DIV**), делимое должно находиться в регистрах **AX** (для 8-битного деления) или **DX:AX** (для 16-битного деления), где **DX** содержит старшие биты, а **AX** — младшие биты делимого.

4. Порядок выполнения лабораторной работы


4.1 Символьные и численные данные в NASM

С помощью команды `mkdir` создайте каталог работы № 6, перейдите в него с команды `cd` и создайте файл `lab6-1.asm`: (рис. 4.1.1)

```
gkarmikhos@fedora:~$ mkdir ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/lab06
gkarmikhos@fedora:~$ cd ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/lab06
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch lab6-1.asm
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

рис. 4.1.1 Создание директории

В файле `6-1.asm` напишите программу вывода значения регистра `eax` (рис. 4.1.2)



```
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-p...
/home/gkarmikhos/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06/lab6-1.asm  Modifie
%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. 4.2 Копи 4.1.2

Создайте исполняемый файл и запустите его. В этом случае при отображении значения регистра `eax` результатом будет символ `j`. (рис.4.1.3)


```
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-1.asm
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 o lab6-1 lab6-1.o
ld: cannot find o: No such file or directory
ld: cannot find lab6-1: No such file or directory
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-1
j
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

рис. 4.1.3 Запустить программу (j)

Изменить текст программы: `mov eax,'6' → mov eax,6` и `mov ebx'4' → mov ebx,4` (рис. 4.1.5). В результате изменения кода при запуске программы возникает пробел (рис. 4.1.4).

```
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-1.asm
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-1

gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

рис. 4.1.4 Выполнить исправленную программу

```
%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. 4.1.5 Исправленный лист

Мы создаем новый файл с именем lab6-2. asm с помощью команды touch (рис. 4.1.6)

```
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch lab6-2.asm
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ mc
```

рис. 4.1.6 Новый файл lab6-2. asm

Скопируйте следующую программу вывода значения регистра `eax` в наш `lab6-2.asm` (рис. 4.1.7)

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. 4.1.7 Программа вывода значения регистра `eax`

Результат выполнения программы дал нам 106, поскольку команда `add` суммирует коды символов (рис 4.1.8)

```
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch lab6-2.asm
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ mc

gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-2.asm
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-2
106
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

рис 4.1.8 Создайте исполняемый файл и запустите его.

Мы редактируем код, удаляя кавычки (рис 4.1.9), и результат нашего кода дает нам результат 10 (рис 4.1.10).

```
%include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

    mov     eax, 6
    mov     ebx, 4
    add     eax, ebx
    call    iprint
    call    quit
```

рис 4.1.9 Измененная программа

```
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-2.asm
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-2
10
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

рис 4.1.10 Запустить программу

Меняем функцию `inprintLF` на `inprint` (рис. 4.1.11)

```
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-2.asm
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-2
10gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

рис. 4.1.11 Выполнить измененную функцию `inprint`

4.2 Выполнение арифметических операций в NASM

Создайте новый файл `lab6-3.asm` с помощью `touch`(рис. 4.2.1)

```
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch lab6-3.asm
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

рис. 4.2.2 Новый файл `lab6-3.asm`

Программа для вычисления арифметического выражения $f(x) = (5*2+3)/3$. (рис. 4.2.3)

```

;-----
; 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
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. 4.2.3 Программа для вычисления арифметического выражения

Результат арифметической задачи равен 4(рис 4.2.4)

```

gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-3.asm
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$

```

рис 4.2.4 Результат первой проблемы

Решите следующую арифметическую задачу $f(x) = (4 * 6 + 2)/5$ (рис 4.2.5)

```

;-----
; 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,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/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 4.2.5 Программа для вычисления арифметического выражения

При выполнении программы мы получаем результат 5 (рис 5.2.6)

```

gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-3.asm
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1

```

рис 4.2.6 Результат второй проблемы

Создайте файл variant.asm и написать программу расчета варианта задания по номеру студенческого билета (рис 4.2.7)

```
*~/work/study/2024-2025/Архитектура компьютера/arch-пс/lab06/variant.asm - Мо...
File Edit Search View Document Help
;-----
; Программа вычисления варианта
;-----

%include 'in_out.asm'

SECTION .data
msg: DB 'Введите № 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`
Архитектура ЭВМ
xor edx,edx
mov ebx,20
div ebx
inc edx
mov eax,rem
call sprint
mov eax,edx
call iprintLF
call quit
```

Рис 4.2.7 Программу расчета варианта

Написать студенческий билет. Когда я писал свой студенческий билет, вариант был 16 (рис 4.2.8)

```
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf variant.asm
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1032244775
Ваш вариант: 16
```

Рис 4.2.8 Вариант

5. Задание для самостоятельной работы

Написать программу вычисления выражения $y = f(x)$ (рис 5.1).

```
/home/gkarmikhos/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06/
#include 'in_out.asm' ; подключение внешнего файла

SECTION .data
msg: DB 'Введите значение переменной x: ',0
rem: DB 'Результат: ',0

SECTION .bss
x: RESB 80 ; Переменная, значение к-рой будем вводить с клавиатуры
SECTION .text
GLOBAL _start
_start:

; --- Вычисление выражения

mov eax, msg
call sprint
mov ecx, x
mov edx, 80
call sread

mov eax, x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, 'eax=x'
add eax, 11; eax = eax + 11 + x + 11
mov ebx, 2
mul ebx; EAX=EAX*EBX = (x+11)*2
add eax, -6; eax = eax - 6 = (x+11)*2 - 6
mov edi, eax ; запись результата вычисления в 'edi'
; --- Выводд 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 5.1 Программа

Вводим число и запускаем программу (рис 5.2)

```
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-4.asm
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-4
Введите значение переменной x: 3
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-4
Введите значение переменной x: 3
Результат: 22gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

рис 5.2 Запустить программу

Вводим число и запускаем программу (рис 5.3)

```
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-4
Введите значение переменной x: 1
Результат: 18gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

рис 5.3 Запустить программу

6. Выводы

В общем, **mc** - это инструмент управления файлами в терминальном режиме, позволяющий выполнять ряд обычных операций, таких как копирование, перемещение и удаление файлов, как с помощью команд `bash`, так и с помощью определенных комбинаций клавиш. Что касается языка ассемблера `NASM`, то его структура состоит из ключевых секций (**.data**, **.bss** и **.text**), которые позволяют определять различные типы данных и кода, оптимизируя использование памяти.

В этой лаборатории мы смогли познакомиться с работой `NASM` и `Midnight Commander`.

Зная функции клавиш `F` (1, 2, 5...10), мы смогли получить доступ к программе и работать над ней, а также использовать команды, ранее использовавшиеся в других лабораториях, такие как `touch`, `nano`.

7. Список используемой литературы

- 1) Архитектура ЭВМ

