---
## Front matter
title: "Шаблон отчёта по лабораторной работе"
subtitle: "Простейший вариант"
author: "Ариоке Габриэль .О."
group: НКАбд-05-22
## Generic otions
lang: ru-RU
toc-title: "Содержание"
## Bibliography
bibliography: bib/cite.bib
csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl
## Pdf output format
toc: true # Table of contents
toc-depth: 2
lof: true # List of figures
lot: true # List of tables
fontsize: 12pt
linestretch: 1.5
papersize: a4
documentclass: scrreprt
## I18n polyglossia
polyglossia-lang:
name: russian
options:
- spelling=modern
- babelshorthands=true
polyglossia-otherlangs:
name: english
## I18n babel
babel-lang: russian
babel-otherlangs: english
## Fonts
mainfont: PT Serif
romanfont: PT Serif
sansfont: PT Sans
monofont: PT Mono
mainfontoptions: Ligatures=TeX
romanfontoptions: Ligatures=TeX
sansfontoptions: Ligatures=TeX,Scale=MatchLowercase
monofontoptions: Scale=MatchLowercase,Scale=0.9
## Biblatex
biblatex: true
biblio-style: "gost-numeric"
biblatexoptions:
- parentracker=true
- backend=biber
- hyperref=auto
- language=auto
- autolang=other\*
- citestyle=gost-numeric
## Pandoc-crossref LaTeX customization
figureTitle: "Рис."
tableTitle: "Таблица"
listingTitle: "Листинг"
lofTitle: "Список иллюстраций"
lotTitle: "Список таблиц"
lolTitle: "Листинги"
## Misc options
indent: true
header-includes:
- \usepackage{indentfirst}
- \usepackage{float} # keep figures where there are in the text
- \floatplacement{figure}{H} # keep figures where there are in the text
---

# Содержание

1. Цель работы
2. Задание
3. Теоретическое введение
4. Выполнение лабораторной работы
5. Задание для самостоятельной работы
6. Листинги программ:
7. Выводы

# Цель работы

Освоение арифметических инструкций языка ассемблера NASM

# Задание

1. Создать файл lab7-1.asm и ввести в него программу из листинга 1, создать исполняемый файл и запустить его.

2. Исправить листинг 1, заменив строки

mov eax,'6' mov ebx, '4' на строки mov eax,6 mov ebx,4 Создать исполняемый файл и запустить его, пользуясь таблицей ASCII опреде- лить какому символу соответствует код 10.

3. Создать файл lab7-2.asm, ввести в него программу из листинга 2, создать исполняемый файл и запустить его.

4. Исправить листинг 2, заменив строки

mov eax,'6' mov ebx, 4' на строки mov eax,6 mov ebx,4 Создать исполняемый файл и запустить его.

5. Заменить функцию iprintLF на iprint. Создать исполняемый файл и запустить его. Выяснить чем отличается вывод функций iprintLF и iprint.

6. Создать файл lab7-3.asm, заполнить его соответственно с листингом 3, создать исполняемый файл и запустить его.

7. Изменить файл так, чтобы программа вычисляла выражение ©[®) = (4 @ 6 +

2›/5

8. Создать файл "вариант", заполнить его соответственно с листингом 4, создать исполняемый файл и запустить его.

9. Ответить на вопросы по разделу.

10. Написать программу для вычисления выражения 5 ₽ (© + 18) - 28 и про.

верить его при ×=2 и при ×=3.

# Теоретическое введение

Схема команды целочисленного сложения add (от англ. addition - добавление) выполняет сложение двух операндов и записывает результат по адресу первого операнда. 1. Команда add работает как с числами со знаком, так и без знака и выглядит следующим образом: add , Допустимые сочетания операндов для ко манды add аналогичны сочетаниям операндов для команды mov. Так, например, команда add eax,ebx прибавит зна- чение из регистра еах к значению из регистра ebx и запишет результат в регистр еах. 2. Команда целочисленного вычитания sub (от англ. subtraction - вычитание) работает аналогично команде add и выглядит следующим образом: sub , Так, например, команда sub ebx,5 уменьшает значение регистра ebx на 5 и записывает результат в регистр ebx. 3. Довольно часто при на писании программ встречается операция прибавле- ния или вычитания единицы.

Прибавление единицы называется инкремен-том, а вычитание - декрементом.

Для этих операций существуют специ- альные команды: іпс (от англ. increment)

и dec (от англ. decrement), которые увеличивают и уменьшают на 1 свой операнд.

Эти команды содержат один операнд и имеет следующий вид: inc dec Операндом может быть регистр или ячейка памяти любого размера. Коман- ды инкремента и декремента выгодны тем, что они занимают меньше места, чем соответствующие команды сложения и вычитания. Так, например, команда inc ebx увеличивает значение регистоа еох на 1. а команда inc ах уменьшает значение регистра ах на

1. 4. Еще одна команда, которую можно отнести к арифметическим командам это команда изменения знака neg: neg Команда пед рассматривает свой операнд как число со знаком и меняет знак операнда на противоположный. Операндом может быть регистр или ячейка памяти любого размера. 5. Умножение и деление в отличии от сложения и вычитания, для знаковых и беззнаковых чисел производиться по-разному, поэтому существуют раз- личные команды. Для беззнакового умножения используется команда ul (от англ. multiply - умножение): ти Для знакового умножения используется команда imul: imul Для команд умножения один из сомножителей указывается в команде и дол- жен находиться в регистре или в памяти, но не может быть непосредственным операндом. Второй сомножитель в команде явно не указывается и должен нахо- диться в регистре EAX,AX или

AL, а результат помещается в регистры EDX:EAX, DX:AX или А, в зависимости от размера операнда. 6. Для деления, как и для умножения, существует 2 команды div (от англ. divide - деление) и idiv: div idiv B командах указывается только один операнд - делитель, который может быть регистром или ячейкой памяти, но не может быть непосредственным операндом. Местоположение делимого и результата для команд деления зависит от размера делителя. Кроме того, так как в результате деления получается два числа - частное и остаток, то эти числа помещаются в определенные регистры.

# Выполнение лабораторной работы

1. Я создала файл lab7-1.asm и ввела в него программу из листинга 1, создала исполняемый файл и запустила его







2. Исправила листинг 1, заменив строки

mov eax,'6' mov ebx, '4' на строки mov eax,6 mov ebx,4 Создала исполняемый файл и запустила его, пользуясь таблицей ASCII опреде- лила какому символу соответствует код 10.





Вывод: код 10 соответствует символу переноса строки, но на экране этот символ не отображается.

3. Создала файл lab7-2.asm, ввела в него программу из листинга 2, создала исполняемый файл и запустила его







4. Исправила листинг 2, заменив строки

mov eax, '6' mov ebx,'4' на строки mov eax,6 mov ebx,4 Создала исполняемый файл и запустила его





5. Заменила функцию iprintLF на iprint. Создала исполняемый файл и запу-стила его. Выяснила чем отличается вывод функций iprintLF и iprint





Вывод: отличие состоит в том, что iprint не совершает перенос строки.

6. Создала файл lab7-3.asm, заполнила его соответственно с листингом 3, со-здала исполняемый файл и запустила его





7. Изменила файл так, чтобы программа вычисляла выражение (9) = (4 @ 186 + 2)/5





8. Создала файл "вариант", заполнила его соответственно с листингом 4, со-здала исполняемый файл и запустила его







Выполнив те же вычисления вручную, выяснила, что ответ, данный програм-мои, верен.

9. Отвечаю на вопросы по разделу:

1. Какие строки листинга 7.4 отвечают за вывод на экран сообщения 'Ваш вариант:?
   'mov eax,rem' 'call sprint
2. Для чего используется следующие инструкции? 'mov ecx, x' - адрес вводимой строких записывется в регистресх. 'mov edx, 80' - 80 - дли-
   на вводимой строки, записанає. 'call sread' - считывание ввода с
   клавиатуры.
3. Для чего используется инструкция "call atoi"?
   Эта инструкция вызывает программу из файла "in\_out.asm" и преоб-разует ascil-код символа в целое число и записает результат в регистр
   eax.
4. Какие строки листинга 7.4 отвечают за вычисления варианта?
   xor edx,edx
   mov ebx. 20
   div ebx
   inc edx
5. В какой регистр записывается остаток от деления при выполнении ин-
   струкции "div ebx"?
6. Для чего используется инструкция "inc edx"?
   Для того, чтобы прибавить к значениах единицу
7. Какие строки листинга 7.4 отвечают за вывод на экран результата вы-числений?
   mov eax,edx
   call iprintLF

# Задание для самостоятельной работы

1. Написать программу для вычисления выражения 5 ₽ (© + 18) - 28 и проверить его при x=2 и при ×=3







Проверила себя, выполнив вычисления вручную - ответ получен верный.

# Листинги программ:

1. lab7-1.asm
   %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
2. lab7-2.asm
   %include 'in\_out.asm'
   SECTION text GLOBAL *start* start:
   mov eax,6 mov ebx,4 add eax,ebx call iprint
   call quit
3. lab7-3.asm

-; 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,ed ; обнуляем EDX для корректной работы div mov ebx,5 ; EBX=5 div ebx ; EAX=EAX/5, EDX=остато

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

4. variant.asm

-; Программа вычисления варианта ;

%include 'in out.asm'

SECTION .data msg: DB 'Введите No студенческого билета:,0 гет: 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,× ; вызов подпрограммы преобразования call atoi ; ASCII кода в

число, еах=х

xor edx, edx mov ebx, 20 div ebx inc edx mov eax,rem call sprint mov eax, ed call iprintLF call quit

5. independentwork.asm - самостоятельная работа

-; Программа вычисления функции

%include 'in\_out.asm°

SECTION .data msg: DB 'Введите значение x:',0 em: DB 'Ваш результат",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

add eax, 18 mov ebx,5 mul ebx

xor edx, edx mov ebx, 28 neg ebx add eax, ebx mov edx, eax mov eax,rem call sprint mov eax,edx call iprintLF

call quit

# Выводы

В ходе этой лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.

# Список литературы

1. Текстовый документ "Лабораторная работа No7. Арифметические операции

B NASM." .: (#refs) .: