

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: Мартемьянов Максим Сергеевич

Студ. Билет 1032250312

Группа: НКАбд-04-25

МОСКВА

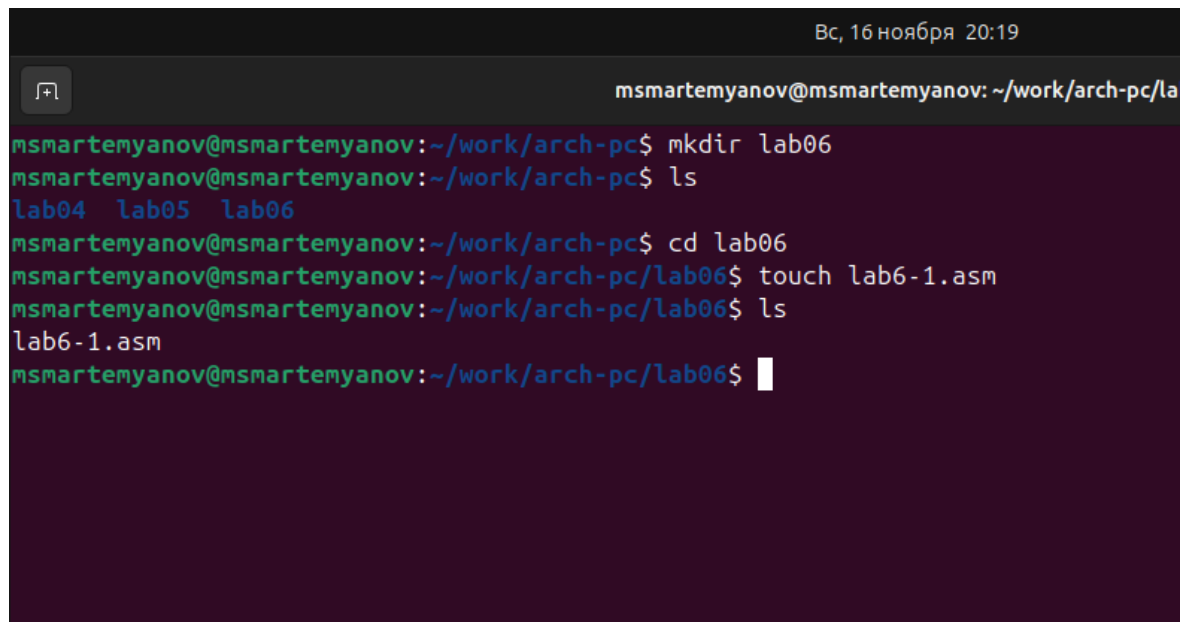
2025г.

Цель работы: Освоение арифметических инструкций языков ассемблера NASM

2. Выполнение лабораторной работы

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

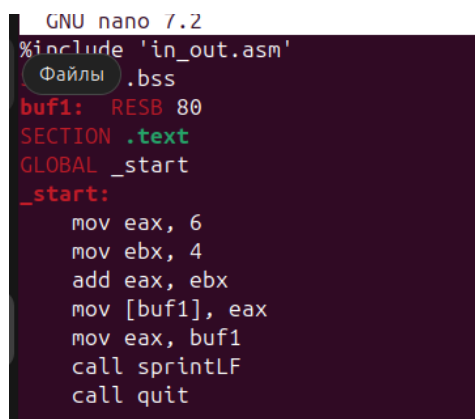
Создал каталог для программ лабораторной работы № 6, и перешел в него, создал файл lab6-1.asm



```
Вс, 16 ноября 20:19
msmartemyanov@msmartemyanov: ~/work/arch-pc/la
msmartemyanov@msmartemyanov:~/work/arch-pc$ mkdir lab06
msmartemyanov@msmartemyanov:~/work/arch-pc$ ls
lab04  lab05  lab06
msmartemyanov@msmartemyanov:~/work/arch-pc$ cd lab06
msmartemyanov@msmartemyanov:~/work/arch-pc/lab06$ touch lab6-1.asm
msmartemyanov@msmartemyanov:~/work/arch-pc/lab06$ ls
lab6-1.asm
msmartemyanov@msmartemyanov:~/work/arch-pc/lab06$
```

Введите в файл lab6-1.asm текст программы из листинга 6.1. В данной программе в регистр еах записывается символ 6 (mov еах,'6'), в регистр еbх символ 4 (mov еbх,'4'). Далее к значению в регистре еах прибавляем значение регистра еbх (add еах,еbх, результат сложения запишется в регистр еах). Далее выводим результат. Так как для работы функции sprintLF в регистр еах должен быть записан адрес, необходимо использовать дополнительную переменную.

Ввел в файл lab6-1.asm текст программы из листинга 6.1. В данной программе в регистр еах записывается символ 6 (mov еах,'6'), в регистр еbх символ 4 (mov еbх,'4') и выполнил все действия.



```
GNU nano 7.2
#include 'in_out.asm'
Файлы .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
    mov еах, 6
    mov еbх, 4
    add еах, еbх
    mov [buf1], еах
    mov еах, buf1
    call sprintLF
    call quit
```

Создал файл и запустил его:

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

Создайте исполняемый файл и запустите его. Как и в предыдущем случае при исполнении программы мы не получим число 10. В данном случае выводится символ с кодом 10. Отображается ли этот символ при выводе на экран?

Исправил файл (убрал кавычки) и запустил его. При выводе на экран символ не отображается

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

msmartemyanov@msmartemyanov:~/work/arch-pc/lab06$
```

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

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

```
msmartemyanov@msmartemyanov: ~/work/arch-pc/lab06
msmartemyanov@msmartemyanov:~/work/arch-pc/lab06$ touch lab6-2.asm
msmartemyanov@msmartemyanov:~/work/arch-pc/lab06$ ls
in_out.asm lab6-1 lab6-1.asm lab6-1.o lab6-2.asm
msmartemyanov@msmartemyanov:~/work/arch-pc/lab06$
```

```
%include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:
    mov eax,6
    mov ebx,4
    add eax, ebx
    call iprintLF
    call quit
```

1.6 Создание и изменение файла lab6-2.asm

Создайте исполняемый файл и запустите его.

Создал файл и запустил его:

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

Исправил строки чтобы получить 10 (iprint)

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

2.2 Выполнение арифметических операций в NASM • В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения $f(x) = (5 * 2 + 3)/3$.

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

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

Создал файл и ввел в него тест из листинга

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

```

; ---
; 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

```

Прочитано 38 строк

Измененный текст

```

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

```

2.5 Вывод измененной программы lab6-3.asm • Создайте файл variant.asm в каталоге ~/work/arch-pc/lab06: touch ~/work/arch-pc/lab06/variant.asm Внимательно изучите текст программы из листинга 6.4 и введите в файл variant.asm.

Создал файл и ввел текст программы

```

msmartemyanov@msmartemyanov:~/work/arch-pc/lab06$ touch variant.asm
msmartemyanov@msmartemyanov:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1      lab6-1.o  lab6-2.asm  lab6-3      lab6-3.o
lab-3.asm   lab6-1.asm  lab6-2    lab6-2.o    lab6-3.asm  variant.asm
msmartemyanov@msmartemyanov:~/work/arch-pc/lab06$

```

Создал файл и запустил его

```
msmartemyanov@msmartemyanov: ~/work/arch-pc/lab06
msmartemyanov@msmartemyanov:~/work/arch-pc/lab06$ nasm -f elf variant.asm
msmartemyanov@msmartemyanov:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
msmartemyanov@msmartemyanov:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1032250312
Ваш вариант: 13
msmartemyanov@msmartemyanov:~/work/arch-pc/lab06$
```

Ответы на вопросы по листингу 6.4:

1. mov eax,rem и call sprint
2. Это подготовка к вызову sread: в ехх кладется адрес буфера х, в edx — его размер (80 байт). call sread читает строку с клавиатуры в этот буфер.
3. call atoi преобразует строку (символы), которую ввел пользователь, в число для арифметических операций.
4. Строки от xor edx,edx до inc edx. 12
5. Остаток от деления записывается в регистр edx.
6. inc edx увеличивает остаток на 1, чтобы получить вариант от 1 до 20 (т.к. остаток от деления на 20 будет от 0 до 19).
7. mov eax,rem + call sprint (вывод сообщения) и mov eax,edx + call iprintLF (вывод номера варианта).

2.3 Задания для самостоятельной работы

1. 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

Вариант 13

```

GNU nano 1.2 /home/msmartemyanov/work/arch-pc/lab06/samostoyatelnaya13.asm
#include 'in_out.asm'

SECTION .data
msg: DB 'f(x) = (8x+6)*10',0
input_msg: DB 'Введите значение x: ',0
result_msg: DB 'Результат вычисления: ',0
x1: DB '1',0
x2: DB '4',0

SECTION .bss
buf: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprintLF
call sprintLF

mov eax, 'X'
call sprint
mov eax, '1'
call sprint
mov eax, '='
call sprint
mov eax, x1
call sprint

```

```

BC, 16 ноября 23:17 en
msmartemyanov@msmartemyanov: ~/work/arch-pc/lab06
msmartemyanov@msmartemyanov:~/work/arch-pc/lab06$ nasm -f elf samostoyatelnaya13.asm
msmartemyanov@msmartemyanov:~/work/arch-pc/lab06$ ld -m elf_i386 -o samostoyatelnaya13 samostoyatelnaya13.o
msmartemyanov@msmartemyanov:~/work/arch-pc/lab06$ ./samostoyatelnaya13
f(x) = (8x+6)*10
f(x) = (8x+6)*10
Ошибка сегментирования (образ памяти сброшен на диск)
msmartemyanov@msmartemyanov:~/work/arch-pc/lab06$

```

2.4 Вопросы для самопроверки

1. ADD операнд1, операнд2 (операнд2 = операнд2 + операнд1)
2. MUL
3. DIV операнд (делитель)
4. В пару регистров DX:AX (старшие:младшие биты).
5. • ADD - сложение • SUB - вычитание • MUL - умножение без знака • IMUL - умножение со знаком • DIV - деление без знака • IDIV - деление со знаком • INC / DEC - инкремент/декремент
6. Для байта - в AX, для слова - в паре DX:AX.
7. Частное - в AX (или AL), остаток - в DX (или AH).

Вывод

В ходе проведенной лабораторной работы освоил арифметические инструкции языка ассемблера NASM.

- Список литературы 1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/LearningbashShell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Lupin С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.
15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).