

Отчёта по лабораторной работе

Арифметические операции в NASM.

Саакян Нерсес Варданович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Ответы на вопросы	14
5	Выводы	15
	Список литературы	16

Список иллюстраций

3.1	Создание каталога и файла лаб 6-1.asm	7
3.2	Программа 1	7
3.3	Запуск программы	8
3.4	Создание файла lab6-2.asm	8
3.5	Введение текста	8
3.6	Запуск файла	8
3.7	Замена символов на числа	9
3.8	Запуск файла	9
3.9	Замена функции	9
3.10	Запуск файла	10
3.11	Создание файла lab6-3.asm	10
3.12	Введение текста	10
3.13	Запуск файла	11
3.14	Изменение ткста	11
3.15	Запуск файла	11
3.16	Создание файла variant.asm	12
3.17	Введение текста	12
3.18	Запуск файла	13

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

1 Написать программу вычисления выражения $x = f(x)$. Вид функции $f(x)$ выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. 2 Загрузите файлы на GitHub.

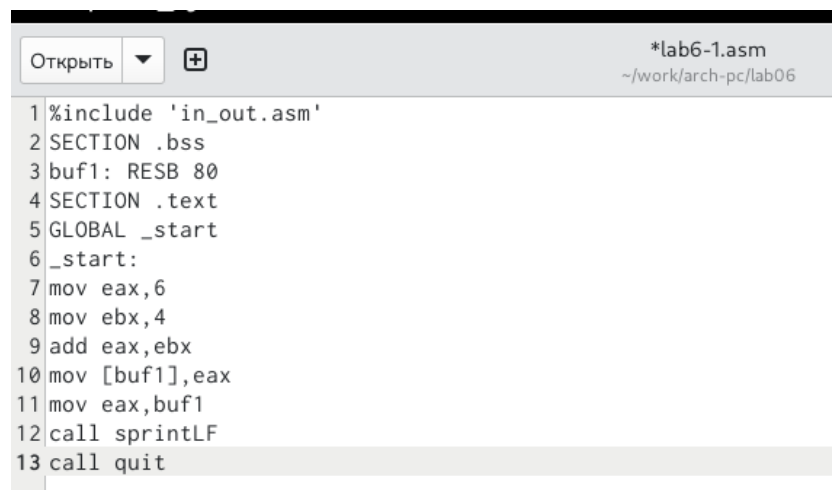
3 Выполнение лабораторной работы

Создайте каталог для программ лабораторной работы No 6, перейдите в него и создайте файл lab6-1.asm (рис. 3.1).

```
nvsaakyan@dk4n71 ~ $ mkdir ~/work/arch-pc/lab06
nvsaakyan@dk4n71 ~ $ cd ~/work/arch-pc/lab06
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ touch lab6-1.asm
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $
```

Рис. 3.1: Создание каталога и файла лаб 6-1.asm

Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения записанные в регистр eax. (рис. 3.2).



```
*lab6-1.asm
~/work/arch-pc/lab06

1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7 mov eax,6
8 mov ebx,4
9 add eax,ebx
10 mov [buf1],eax
11 mov eax,buf1
12 call sprintLF
13 call quit
```

Рис. 3.2: Программа 1

Создайте исполняемый файл и запустите его. (рис. 3.3).

```
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ ./lab6-1
```

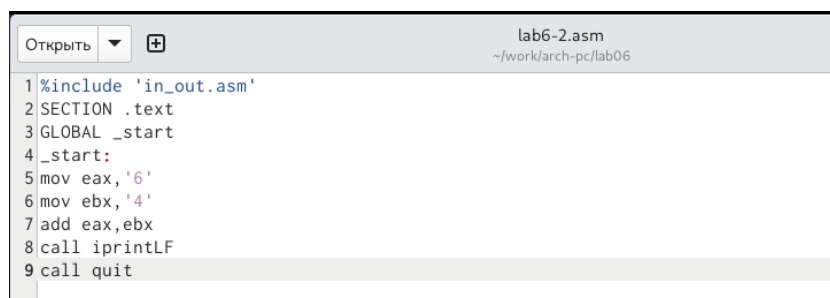
Рис. 3.3: Запуск программы

Создайте файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 (рис. 3.4).

```
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-2.asm
```

Рис. 3.4: Создание файла lab6-2.asm

Введите в него текст программы из листинга 6.2. (рис. 3.5).



```
lab6-2.asm
~/.work/arch-pc/lab06

1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax, '6'
6 mov ebx, '4'
7 add eax, ebx
8 call iprintLF
9 call quit
```

Рис. 3.5: Введение текста

Создайте исполняемый файл и запустите его. (рис. 3.6).

```
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ ./lab6-2
106
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $
```

Рис. 3.6: Запуск файла

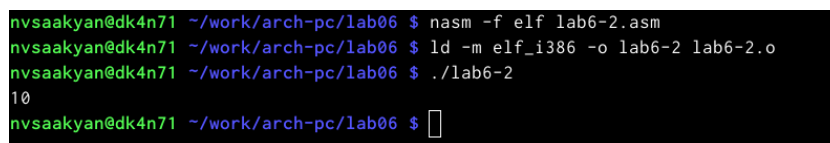
Аналогично предыдущему примеру изменим символы на числа. Замените строки (рис. 3.7).



```
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax,6
6 mov ebx,4
7 add eax,ebx
8 call iprintLF
9 call quit
```

Рис. 3.7: Замена символов на числа

Создайте исполняемый файл и запустите его. (рис. 3.8).



```
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ ./lab6-2
10
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $
```

Рис. 3.8: Запуск файла

В результате получили число 10.

Замените функцию iprintLF на iprint (рис. 3.9).



```
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax,6
6 mov ebx,4
7 add eax,ebx
8 call iprint
9 call quit
```

Рис. 3.9: Замена функции

Создайте исполняемый файл и запустите его. (рис. 3.10).

```
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ ./lab6-2
10nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $
```

Рис. 3.10: Запуск файла

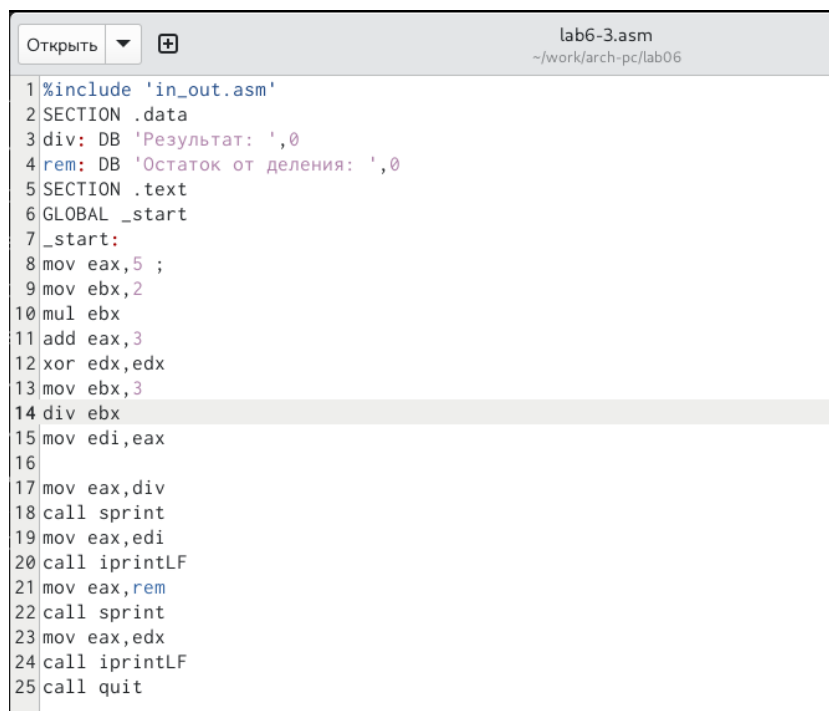
В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения $(5 \times 2 + 3)/3$

Создайте файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 (рис. 3.11).

```
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-3.asm
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $
```

Рис. 3.11: Создание файла lab6-3.asm

Внимательно изучите текст программы из листинга 6.3 и введите в lab6-3.asm. (рис. 3.12).



```
lab6-3.asm
~/work/arch-pc/lab06

1 %include 'in_out.asm'
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8 mov eax,5 ;
9 mov ebx,2
10 mul ebx
11 add eax,3
12 xor edx,edx
13 mov ebx,3
14 div ebx
15 mov edi,eax
16
17 mov eax,div
18 call sprint
19 mov eax,edi
20 call iprintLF
21 mov eax,rem
22 call sprint
23 mov eax,edx
24 call iprintLF
25 call quit
```

Рис. 3.12: Введение текста

Создайте исполняемый файл и запустите его. (рис. 3.13).

```

nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $

```

Рис. 3.13: Запуск файла

Измените текст программы для вычисления выражения $(4 \times 6 + 2)/5$ (рис. 3.14).

```

lab6-3.asm
~/work/arch-pc/lab06
1 %include 'in_out.asm'
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8 mov eax,4
9 mov ebx,6
10 mul ebx
11 add eax,2
12 xor edx,edx
13 mov ebx,5
14 div ebx
15 mov edi,eax
16
17 mov eax,div
18 call sprint
19 mov eax,edi
20 call iprintLF
21 mov eax,rem
22 call sprint
23 mov eax,edx
24 call iprintLF
25 call quit

```

Рис. 3.14: Изменение текста

Создайте исполняемый файл и проверьте его работу. (рис. 3.15).

```

nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $

```

Рис. 3.15: Запуск файла

В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму:

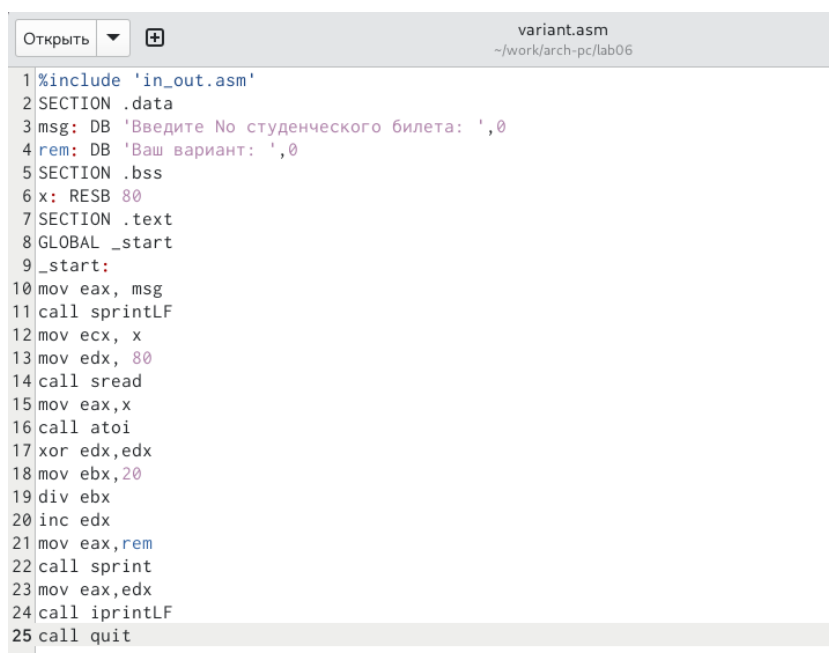
- вывести запрос на введение No студенческого билета
- вычислить номер варианта по формуле: $(\text{№} \bmod 20) + 1$, где № – номер студенческого билета (В данном случае $\text{№} \bmod \text{№}$ – это остаток от деления № на №).
- вывести на экран номер варианта.

Создайте файл variant.asm в каталоге ~/work/arch-pc/lab06 (3.16).

```
nvsaakyan@dk5n56 ~ $ touch ~/work/arch-pc/lab06/variant.asm
nvsaakyan@dk5n56 ~ $
```

Рис. 3.16: Создание файла variant.asm

Внимательно изучите текст программы из листинга 6.4 и введите в файл variant.asm. (рис. 3.17).



```
1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите No студенческого билета: ',0
4 rem: DB 'Ваш вариант: ',0
5 SECTION .bss
6 x: RESB 80
7 SECTION .text
8 GLOBAL _start
9 _start:
10 mov eax, msg
11 call sprintf
12 mov ecx, x
13 mov edx, 80
14 call sread
15 mov eax, x
16 call atoi
17 xor edx, edx
18 mov ebx, 20
19 div ebx
20 inc edx
21 mov eax, rem
22 call sprintf
23 mov eax, edx
24 call iprintLF
25 call quit
```

Рис. 3.17: Введение текста

Создайте исполняемый файл и запустите его (рис. 3.18).

```
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $ ./variant
Введите No студенческого билета:
1132239125
Ваш вариант: 6
nvsaakyan@dk4n71 ~/work/arch-pc/lab06 $
```

Рис. 3.18: Запуск файла

4 Ответы на вопросы

1 `“mov eax, rem”` и `“call sprint”` в секции кода отвечают за вывод сообщения “Ваш вариант:” на экран. 2 `“mov ecx, x”` и `“mov edx, 80”` загружают адрес буфера (x) и длину буфера (80) соответственно в регистры ecx и edx для вызова подпрограммы `sread`, которая считывает строку из консоли. 3 `“call atoi”` вызывает подпрограмму `atoi` для преобразования ASCII кодов символов в число, результат которого сохраняется в регистре eax. 4 Код для вычисления варианта начинается с `“xor edx, edx”` и `“mov ebx, 20”`, после чего происходит деление числа, сохраненного в eax, на 20 с помощью инструкции `“div ebx”`. Результат деления (цифра варианта) записывается в нижнюю часть регистра AX (AL). Затем происходит увеличение цифры варианта на единицу с помощью инструкции `“inc edx”`. 5 Результат деления (цифра варианта) записывается в нижнюю часть регистра AX (AL). 6 Инструкция `“inc edx”` увеличивает цифру варианта на единицу для того, чтобы результат деления не оказывался равным нулю. 7 `“mov eax, edx”` загружает цифру варианта из AL (нижней части регистра AX) в регистр eax для вывода результата на экран с помощью подпрограммы `iprintLF`.

5 Выводы

В ходе выполнения работы, я освоил работу с арифметическими операциями на языке assembly.

Список литературы

GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>. 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/Learning-bash-Shell-Programming-Nutshell/dp/0596009658>. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВ- Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1. 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. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science). Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015 — 1120 с. — (Классика Computer Science). :: {#refs} ::