

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

дисциплина: Архитектура вычислительных систем

Студент: Уханаева Сансара Зоригтуевна

Студ. билет № 1132229047

Группа: НБИбд-03-22

МОСКВА

2022 г.

Цель работы:

Освоение арифметических инструкций языка ассемблера NASM.

Выполнение лабораторной работы:

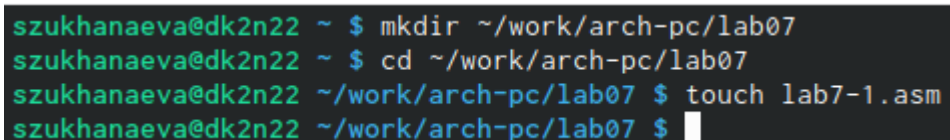
6.3.1. Символьные и численные данные в NASM

1. Создадим каталог для программ лабораторной работы No 7, перейдите в него и создайте файл lab7-1.asm:

```
mkdir ~/work/arch-pc/lab07
```

```
cd ~/work/arch-pc/lab07
```

```
touch lab7-1.asm
```



```
szukhanaeva@dk2n22 ~ $ mkdir ~/work/arch-pc/lab07
szukhanaeva@dk2n22 ~ $ cd ~/work/arch-pc/lab07
szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $ touch lab7-1.asm
szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $
```

2. Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения записанные в регистр eax. Введите в файл lab7-1.asm текст программы из листинга 7.1.



```
Открыть  [icon] *lab7-1.asm
~/work/arch-pc/lab07 Сохранить

1 %include 'in_out.asm'
2
3 SECTION .bss
4 buf1:   RESB 80
5
6 SECTION .text
7 GLOBAL _start
8
9 _start:
10
11 mov eax, '6'
12 mov ebx, '4'
13 add eax, ebx
14 mov [buf1], eax
15 mov eax, buf1
16 call sprintf
17
18 call quit
```

Создадим исполняемый файл и запустите его.

```
nasm -f elf lab7-1.asm
```

```
cd ~/work/arch-pc/lab07
```

```
ld -m elf_i386 -o lab7-1 lab7-1.o
```

```
./lab7-1
```

```
szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $ cd ~/work/arch-pc/lab07
szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $ ./lab7-1
j
szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $
```

3. Далее изменим текст программы и вместо символов, запишем в регистры числа. Исправьте текст программы (Листинг 1) следующим образом:

замените строки

```
mov eax,'6'
```

```
mov ebx,'4'
```

на строки

```
mov eax,6
```

```
mov ebx,4
```

```
GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/s/z/szukhanaeva/work
#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
```

Создадим 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. Пользуясь таблицей ASCII определим какому символу соответствует код 10. Отображается ли этот символ при выводе на экран?

```

szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $ cd ~/work/arch-pc/lab07
szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $ ./lab7-1

szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $ █

```

4. Как отмечалось выше, для работы с числами в файле `in_out.asm` реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Преобразуем текст программы из Листинга 7.1 с использованием этих функций.

Создадим файл `lab7-2.asm` в каталоге `~/work/arch-pc/lab07` и введем в него текст программы из листинга 7.2.

```
touch ~/work/arch-pc/lab07/lab7-2.asm
```

```

szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $ cd
szukhanaeva@dk2n22 ~ $ touch ~/work/arch-pc/lab07/lab7-2.asm
szukhanaeva@dk2n22 ~ $ █

```

Создадим исполняемый файл и запустим его.

```
nasm -f elf lab7-1.asm
```

```
ld -m elf_i386 -o lab7-1 lab7-1.o
```

```
./lab7-1
```

```

szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $ ./lab7-1

```

5. Аналогично предыдущему примеру изменим символы на числа. Заменяем строки

```
mov eax,'6'
```

```
mov ebx,'4'
```

на строки

```
mov eax,6
```

```
mov ebx,4
```

```

mov eax,6
mov ebx,4

```

Создадим исполняемый файл и запустите его.

Заменяем функцию `iprintLF` на `iprint`. Создадим исполняемый файл и запустим его.

6.3.2. Выполнение арифметических операций в NASM

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

Создадим файл `lab7-3.asm` в каталоге `~/work/arch-pc/lab07`:

```
touch ~/work/arch-pc/lab07/lab7-3.asm
```

Внимательно изучим текст программы из листинга 7.3 и введем в `lab7-3.asm`.

```
szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $ touch ~/work/arch-pc/lab07/lab7-3.asm
szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $
```

Создадим исполняемый файл и запустите его. Результат работы программы должен быть следующим:

```
user@dk4n31:~$ ./lab7-3
```

Результат: 4

Остаток от деления: 1

```
user@dk4n31:~$
```

Измените текст программы для вычисления выражения $f(x) = (4 * 6 + 2) / 5$.

Создадим исполняемый файл и проверим его работу.

7. 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: $(S_n \bmod 20) + 1$, где S_n – номер студенческого билета (В данном случае $a \bmod b$ – это остаток от деления a на b).

- вывести на экран номер варианта.

Создадим файл `variant.asm` в каталоге `~/work/arch-pc/lab07`:

`touch ~/work/arch-pc/lab07/variant.asm`

```
szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $ touch ~/work/arch-pc/lab07/variant.asm
szukhanaeva@dk2n22 ~/work/arch-pc/lab07 $
```

Внимательно изучим текст программы из листинга 7.4 и введем в файл `variant.asm`

Создадим исполняемый файл и запустим его. Проверим результат работы программы вычислив номер варианта аналитически.

Задание для самостоятельной работы

1. Напишем 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. Создайте исполняемый файл и проверьте его работу для значений x_1 и x_2 из 6.3

Вывод: Я освоила арифметических инструкций языка ассемблера NASM.