

Отчёт по лабораторной работе №6

Дисциплина: Архитектура компьютера

Ким Денис Вячеславович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	12
	Список литературы	13

Список иллюстраций

4.1	Создание файла в новом каталоге	8
4.2	Ввод заданного текста программы	8
4.3	Запуск файла программы	9
4.4	Изменение текста программы	9
4.5	Создание файла и запуск программы	9
4.6	Запуск изменённой программы	10
4.7	Запуск программы вычисления функции	10
4.8	Запуск программы вычисления варианта задания	10
4.9	Запуск программы для вычисления заданной функции	11

Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . . .	7
-----	---	---

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

В процессе данной лабораторной работы мне предстоит познакомиться с арифметическими инструкциями языка ассемблера NASM и освоить работу с ними.

3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы. Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux	
Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую систему
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно про Unix см. в [1–4].

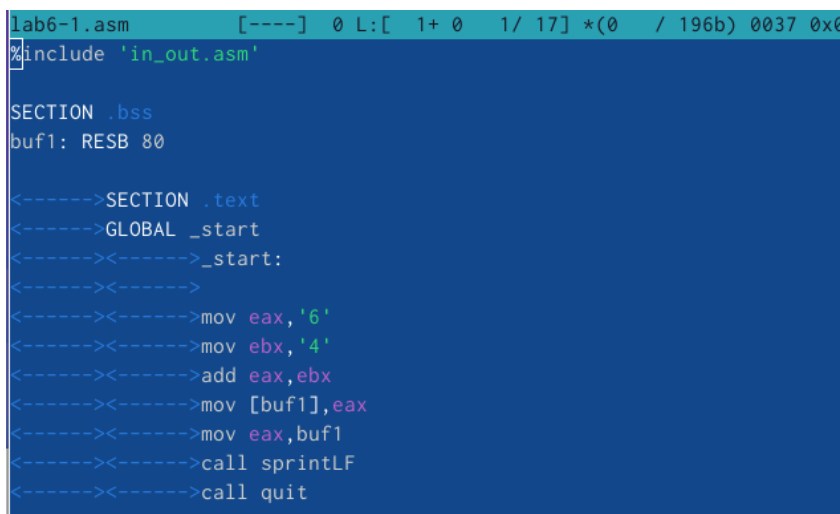
4 Выполнение лабораторной работы

Создаём каталог для программ лабораторной работы № 6, переходим в него и создаём файл lab6-1.asm: (рис. 4.1):

```
dvkim@dk8n76 ~/work/arch-pc/lab06 $ touch lab6-1.asm
dvkim@dk8n76 ~/work/arch-pc/lab06 $
```

Рис. 4.1: Создание файла в новом каталоге

Вводим в файл lab6-1.asm текст программы из листинга 6.1.: (рис. 4.2).



```
lab6-1.asm      [----] 0 L: [ 1+ 0  1/ 17] *(0  / 196b) 0037 0xc
%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: Ввод заданного текста программы

Запускаем файл: (рис. 4.3).


```

dvkim@dk8n76 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
dvkim@dk8n76 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
dvkim@dk8n76 ~/work/arch-pc/lab06 $ ./lab6-1
j
dvkim@dk8n76 ~/work/arch-pc/lab06 $ 

```

Рис. 4.3: Запуск файла программы

Изменяем текст программы, как задано в условии. При его запуске на выход идёт пустая строка: (рис. 4.4).

```

lab6-1.asm      [-M--]  0 L:[ 1+16 17/ 17] *(192 / 192b) <EOF>
#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. 4.4: Изменение текста программы

Создаём файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и вводим в него текст программы из листинга 6.2. Далее запускаем его и получаем на выходе число 106: (рис. 4.5).

```

dvkim@dk8n76 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-2.asm
dvkim@dk8n76 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
dvkim@dk8n76 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
dvkim@dk8n76 ~/work/arch-pc/lab06 $ ./lab6-2
106
dvkim@dk8n76 ~/work/arch-pc/lab06 $ 

```

Рис. 4.5: Создание файла и запуск программы

Аналогично предыдущему примеру меняем символы на числа. Запускаем изменённый файл и получаем на выходе число 10. Если заменить функцию `iprintLF` на `iprint`, то отступа после вывода результата не будет: (рис. 4.6).

```

dvkim@dk8n76 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
dvkim@dk8n76 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
dvkim@dk8n76 ~/work/arch-pc/lab06 $ ./lab6-2
10

```

Рис. 4.6: Запуск изменённой программы

В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения $f(x) = (5 \cdot 2 + 3)/3$. Создаём новый файл и вводим в него заданный текст. Запускаем программу: (рис. 4.7).

```

dvkim@dk8n76 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
dvkim@dk8n76 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
dvkim@dk8n76 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1

```

Рис. 4.7: Запуск программы вычисления функции

В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета, работающую по заданному алгоритму. Это делается в новом файле. Запускаем программу и вводим номер студенческого билета. Нам достался вариант 11: (рис. 4.8).

```

dvkim@dk8n76 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/variant.asm
dvkim@dk8n76 ~/work/arch-pc/lab06 $ nasm -f elf lab6-4.asm
nasm: fatal: unable to open input file 'lab6-4.asm' No such file or directory
dvkim@dk8n76 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
dvkim@dk8n76 ~/work/arch-pc/lab06 $ ^C
dvkim@dk8n76 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
dvkim@dk8n76 ~/work/arch-pc/lab06 $ ^C
dvkim@dk8n76 ~/work/arch-pc/lab06 $ ./variant
Введите № студенческого билета:
1132246770
Ваш вариант: 11

```

Рис. 4.8: Запуск программы вычисления варианта задания

Отвечаем на заданные вопросы:

1. `mov eax,rem; call sprint`
2. Для записи длины вводимого сообщения
3. Для перевода ASCII кода в число
4. Начиная с `xor edx,edx` и заканчивая `inc edx`

5. В регистр `eax`
6. Для увеличения операнда на единицу
7. `mov eax,edx; call iprintLF`

Выполняем задания для самостоятельной работы. Напишем программу вычисления выражения $y = f(x)$. Нам достался 11 вариант, поэтому пишем программу для вычисления функции $f(x) = 10(x + 1) - 10$. После этого вводим заданные значения: (рис. 4.9).

```
dvkim@dk8n74 ~/work/arch-pc/lab06 $ nasm -f elf lab6-4.asm
dvkim@dk8n74 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-4 lab6-4.o
dvkim@dk8n74 ~/work/arch-pc/lab06 $ ./lab6-4
Введите значение x: 1
Результат: 10dvkim@dk8n74 ~/work/arch-pc/lab06 $ ./lab6-4
Введите значение x: 7
Результат: 70dvkim@dk8n74 ~/work/arch-pc/lab06 $ □
```

Рис. 4.9: Запуск программы для вычисления заданной функции

5 Выводы

В ходе данной работы я освоил работу с арифметическими инструкциями языка ассемблера NASM. Кроме того, я научился писать собственные программы для вычисления функций с неизвестными переменными.

Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.
2. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.