

Отчёт по лабораторной работе №4. Создание и процесс обработки программ на языке ассемблера NASM

Архитектура вычислительных систем

Колосов Даниил Дмитриевич. НБИбд-02-22.

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Вывод	11

Список иллюстраций

3.1	Рис 1	7
3.2	Рис 2	7
3.3	Рис 4	8
3.4	Рис 5	8
3.5	Рис 6	8
3.6	Рис 7	9
3.7	Рис 8	9
3.8	Рис 9	9
3.9	Рис 10	9
3.10	Рис 11	10

Список таблиц

1 Цель работы

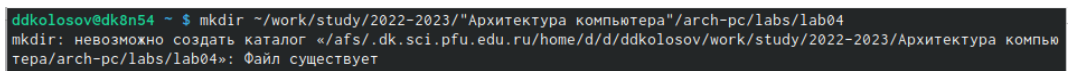
Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

2 Задание

1. В каталоге `~/work/arch-рс/lab04` с помощью команды `ср` создайте копию файла `hello.asm` с именем `lab4.asm`
2. С помощью любого текстового редактора внесите изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с вашими фамилией и именем.
3. Оттранслируйте полученный текст программы `lab4.asm` в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.
4. Скопируйте файлы `hello.asm` и `lab4.asm` в Ваш локальный репозиторий в каталог `~/work/study/2022-2023/“Архитектура компьютера”/arch-рс/labs/lab04/`. Загрузите файлы на Github.

3 Выполнение лабораторной работы

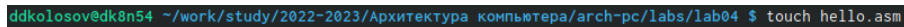
1. Создадим каталог для работы с программами на языке ассемблера NASM



```
ddkolosov@dk8n54 ~ $ mkdir ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab04
mkdir: невозможно создать каталог «/afs/.dk.sci.pfu.edu.ru/home/d/d/kolosov/work/study/2022-2023/Архитектура компью
тера/arch-pc/labs/lab04»: Файл существует
```

Рис. 3.1: Рис 1

2. В данном каталоге создаем файл с именем hello.asm.



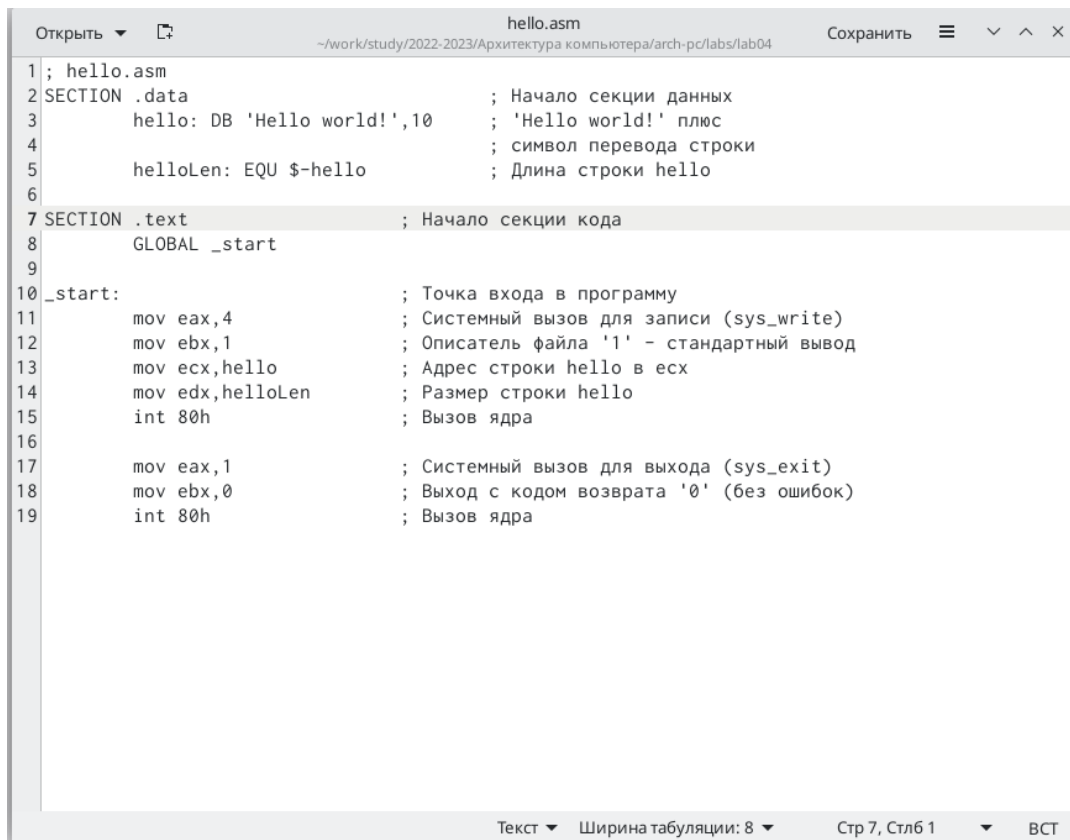
```
ddkolosov@dk8n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ touch hello.asm
```

Рис. 3.2: Рис 2

3. И откроем этот файл с помощью текстового редактора gedit

Рис 3

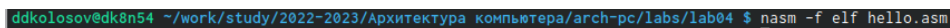
4. Вводим текст из материалов по лабораторной работе №4



```
1 ; hello.asm
2 SECTION .data                ; Начало секции данных
3     hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4                                     ; символ перевода строки
5     helloLen: EQU $-hello      ; Длина строки hello
6
7 SECTION .text                ; Начало секции кода
8     GLOBAL _start
9
10 _start:
11     mov eax,4                ; Точка входа в программу
12     mov ebx,1                ; Системный вызов для записи (sys_write)
13     mov ecx,hello            ; Описатель файла '1' - стандартный вывод
14     mov edx,helloLen         ; Адрес строки hello в ecx
15     int 80h                  ; Размер строки hello
16                               ; Вызов ядра
17
18     mov eax,1                ; Системный вызов для выхода (sys_exit)
19     mov ebx,0                ; Выход с кодом возврата '0' (без ошибок)
20     int 80h                  ; Вызов ядра
```

Рис. 3.3: Рис 4

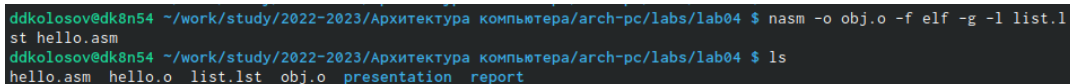
5. NASM превращает текст программы в объектный код.



```
ddkolosov@dk8n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ nasm -f elf hello.asm
```

Рис. 3.4: Рис 5

5. Скомпилирует исходный файл hello.asm в obj.o и проверяем.



```
ddkolosov@dk8n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
ddkolosov@dk8n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o  presentation  report
```

Рис. 3.5: Рис 6

6. Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику


```
ddkolosov@dk8n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ld -m elf_i386 hello.o -o hello
ddkolosov@dk8n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ld -m elf_i386 obj.o -o main
```

Рис. 3.6: Рис 7

7. Запустить на выполнение созданный исполняемый файл, находящийся в текущем каталоге, можно, набрав в командной строке:

```
ddkolosov@dk8n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ./hello
Hello world!
```

Рис. 3.7: Рис 8

8. Создайте копию файла

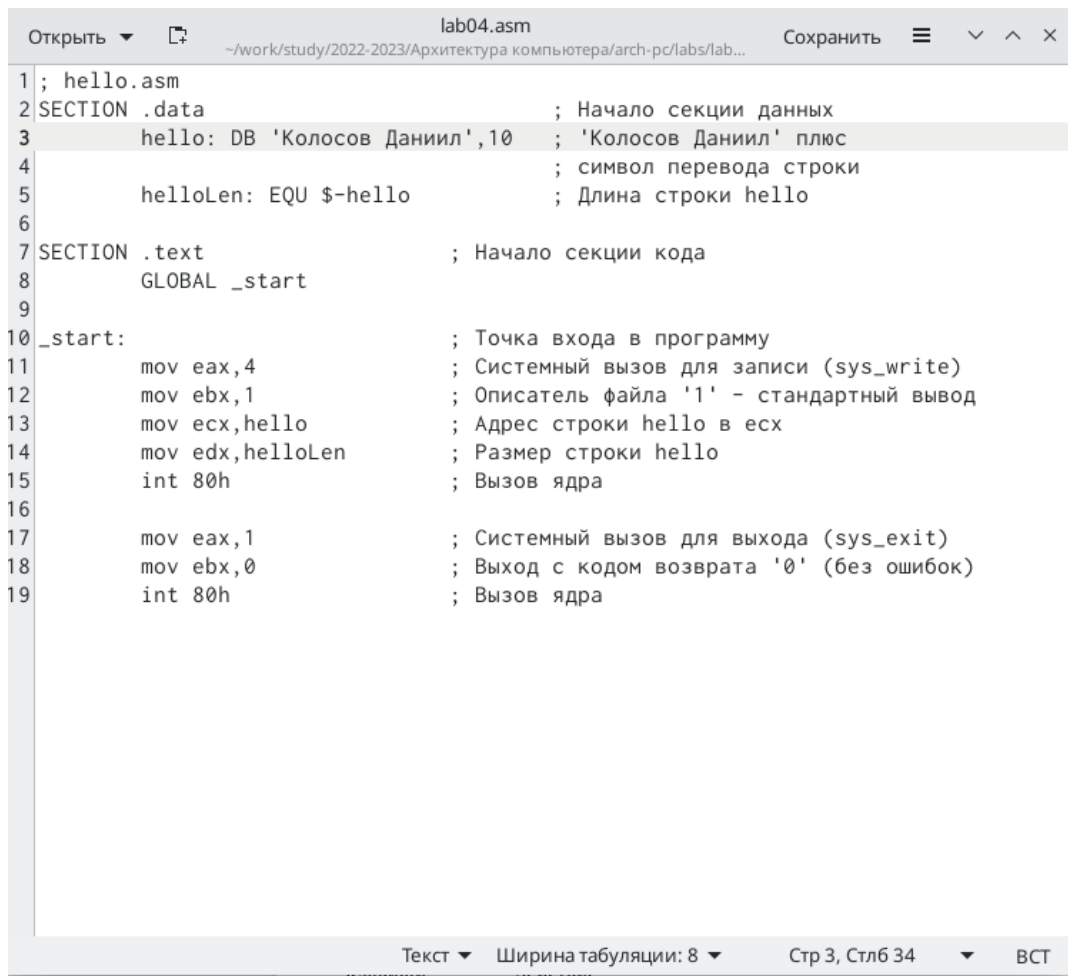
```
ddkolosov@dk8n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs $ cp lab04/hello.asm lab04/lab04.asm
```

Рис. 3.8: Рис 9

9. Проведем подобные действия, которые мы делали с файлом hello.asm, с файлом lab4.asm, чтобы вывести на экран имя и фамилию. Оттранслируем данный файл в объектный и запустим.

```
ddkolosov@dk8n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs $ cd lab04
ddkolosov@dk8n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ gedit lab04.asm
ddkolosov@dk8n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ nasm -f elf lab04.asm
ddkolosov@dk8n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ./lab04
bash: ./lab04: Нет такого файла или каталога
ddkolosov@dk8n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello  hello.asm  hello.o  lab04.asm  lab04.o  list.lst  main  obj.o  presentation  report
ddkolosov@dk8n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ nasm -o obj.o -f elf -g -l list.lst lab04.asm
ddkolosov@dk8n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ld -m elf_i386 lab04.o -o lab04
ddkolosov@dk8n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello  hello.asm  hello.o  lab04  lab04.asm  lab04.o  list.lst  main  obj.o  presentation  report
ddkolosov@dk8n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ./lab04
Колосов Даниил
```

Рис. 3.9: Рис 10



```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3     hello: DB 'Колосов Даниил',10 ; 'Колосов Даниил' плюс
4                                     ; символ перевода строки
5     helloLen: EQU $-hello ; Длина строки hello
6
7 SECTION .text ; Начало секции кода
8     GLOBAL _start
9
10 _start: ; Точка входа в программу
11     mov eax,4 ; Системный вызов для записи (sys_write)
12     mov ebx,1 ; Описатель файла '1' - стандартный вывод
13     mov ecx,hello ; Адрес строки hello в ecx
14     mov edx,helloLen ; Размер строки hello
15     int 80h ; Вызов ядра
16
17     mov eax,1 ; Системный вызов для выхода (sys_exit)
18     mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
19     int 80h ; Вызов ядра
```

Текст ▾ Ширина табуляции: 8 ▾ Стр 3, Стлб 34 ▾ ВСТ

Рис. 3.10: Рис 11

9. Загрузим файлы в репозиторий

4 Вывод

Мы освоили процедуры компиляции и сборки программ, написанных на ассемблере NASM.