

Отчёт по лабораторной работе№4

Создание и процесс обработки программ на языке ассемблера NASM

Кузнецова Александра Сергеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	13

Список иллюстраций

3.1	Создание каталога	7
3.2	Переход в каталог	7
3.3	Создание текстового файла	7
3.4	Открытие файла	7
3.5	Ввод текста	8
3.6	Компиляция текста	8
3.7	проверка, что объектный файл был создан	9
3.8	Создание файлов	9
3.9	Проверка, что файлы были созданы.	9
3.10	Передача файла на компоновку	9
3.11	Проверка, что исполняемый файл hello был создан	9
3.12	Зададим имя создаваемого исполняемого файла	10
3.13	Запуск на выполнение созданный исполняемый файл	10
3.14	Создание копии файла с именем lab4.asm	10
3.15	Внесение изменения в текст программы	11
3.16	Оттранслирование, компоновка, запуск	11
3.17	Копирование файлов в локальный репозиторий	12
3.18	Загрузка файлов на гитхаб	12

Список таблиц

1 Цель работы

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

2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

3 Выполнение лабораторной работы

1. Создайте каталог для работы с программами на языке ассемблера NASM:

```
askuznecova@dk1n22 ~ $ mkdir -p ~/work1/arch-pc/lab04
```

Рис. 3.1: Создание каталога

2. Перейдём в созданный каталог:

```
askuznecova@dk1n22 ~ $ cd ~/work1/arch-pc/lab04
```

Рис. 3.2: Переход в каталог

3. Создадим текстовый файл с именем hello.asm:

```
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ touch hello.asm
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ ls
hello.asm
```

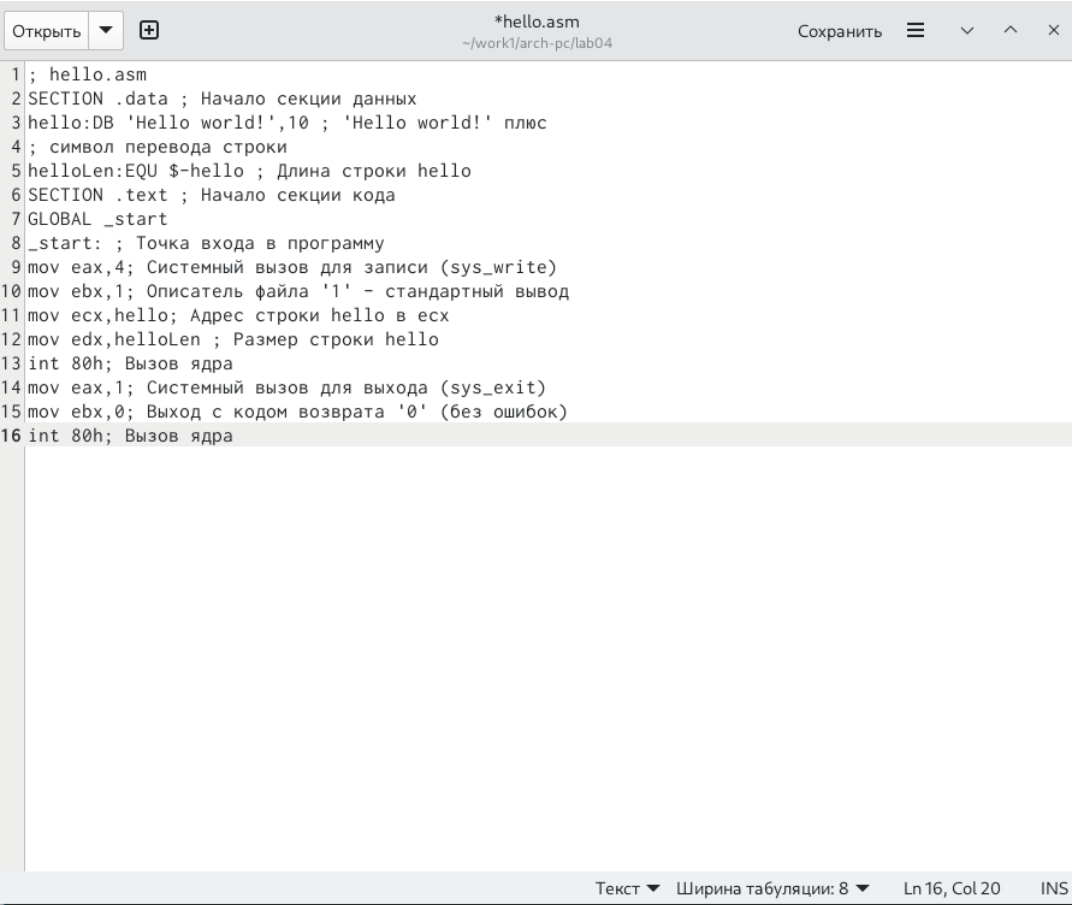
Рис. 3.3: Создание текстового файла

4. Откроем этот файл с помощью текстового редактора

```
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ gedit hello.asm
```

Рис. 3.4: Открытие файла

5. Введём в него текст:

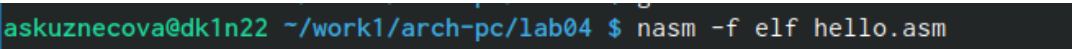


The screenshot shows a text editor window titled '*hello.asm' with the path '~/.work1/arch-pc/lab04'. The editor contains 16 lines of assembly code. The code defines a data section for the string 'Hello world!', calculates its length, and then uses system calls to write the string to stdout and exit the program. The status bar at the bottom indicates 'Текст', 'Ширина табуляции: 8', 'Ln 16, Col 20', and 'INS'.

```
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 SECTION .text ; Начало секции кода
7 GLOBAL _start
8 _start: ; Точка входа в программу
9 mov eax,4; Системный вызов для записи (sys_write)
10 mov ebx,1; Описатель файла '1' - стандартный вывод
11 mov ecx,hello; Адрес строки hello в ecx
12 mov edx,helloLen ; Размер строки hello
13 int 80h; Вызов ядра
14 mov eax,1; Системный вызов для выхода (sys_exit)
15 mov ebx,0; Выход с кодом возврата '0' (без ошибок)
16 int 80h; Вызов ядра
```

Рис. 3.5: Ввод текста

6. Скомпилируем данный текст



The screenshot shows a terminal window with the command 'nasm -f elf hello.asm' being executed. The prompt is 'askuznecova@dk1n22' and the current directory is '~/.work1/arch-pc/lab04'.

```
askuznecova@dk1n22 ~/.work1/arch-pc/lab04 $ nasm -f elf hello.asm
```

Рис. 3.6: Компиляция текста

7. Проверим, что объектный файл был создан:


```
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ ls
hello.asm hello.o
```

Рис. 3.7: проверка, что объектный файл был создан

8. Скомпилируем исходный файл hello.asm в obj.o и создадим файл листинга list.lst

```
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
```

Рис. 3.8: Создание файлов

9. Проверим, что файлы были созданы.

```
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ ls
hello.asm hello.o list.lst obj.o
```

Рис. 3.9: Проверка, что файлы были созданы.

10. Передадим объектный файл на обработку компоновщику.

```
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
```

Рис. 3.10: Передача файла на компоновку

11. Проверим, что исполняемый файл hello был создан.

```
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ ls
hello hello.asm hello.o list.lst obj.o
```

Рис. 3.11: Проверка, что исполняемый файл hello был создан

12. Зададим имя создаваемого исполняемого файла.

```
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ ld -m elf_i386 hello.o -o main
```

Рис. 3.12: Зададим имя создаваемого исполняемого файла

13. Запустим на выполнение созданный исполняемый файл, находящийся в текущем каталоге.

```
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ ./hello  
Hello world!
```

Рис. 3.13: Запуск на выполнение созданный исполняемый файл

14. Создадим копию файла hello.asm с именем lab4.asm

```
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ cp hello.asm lab4.asm
```

Рис. 3.14: Создание копии файла с именем lab4.asm

15. Внесём изменения в текст программы в файле lab5.asm

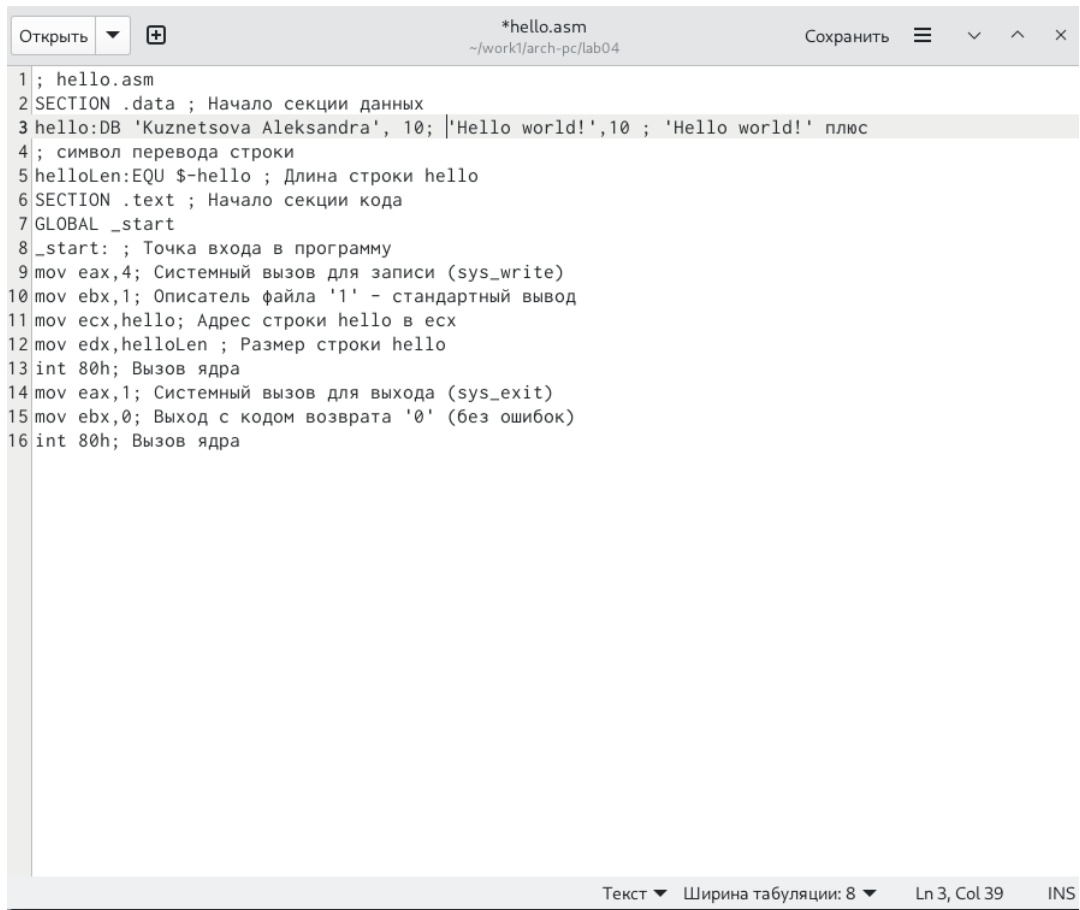


Рис. 3.15: Внесение изменения в текст программы

16. Оттранслируем полученный текст программы lab5.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл.

```

askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ gedit hello.asm
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ gedit hello.asm
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ cp hello.asm lab4.asm
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ nasm -f elf lab4.asm
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst lab4.asm
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ nasm -o Kuznetsova.o -f elf -g -l list.lst lab4.asm
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ ld -m elf_i386 Kuznetsova.o -o Kuznetsova
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ ./Kuznetsova
Kuznetsova Aleksandra

```

Рис. 3.16: Оттранслирование, компоновка, запуск

17. Скопировала файлы hello.asm и lab4.asm в локальный репозиторий в ката-

лог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/ с помощью утилиты `sr` и проверил наличие файлов с помощью утилиты `ls`

```
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ cp hello.asm ~/work/study/2023-2024/“Архитектура компьютера”/study_2023-2024_arh-pc/labs/lab04/report
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ cp lab4.asm ~/work/study/2023-2024/“Архитектура компьютера”/study_2023-2024_arh-pc/labs/lab04/report
askuznecova@dk1n22 ~/work1/arch-pc/lab04 $ cd ~/work/study/2023-2024/“Архитектура компьютера”/study_2023-2024_arh-pc/labs/lab04/report
askuznecova@dk1n22 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab04/report $ ls
hello.asm  lab4.asm
```

Рис. 3.17: Копирование файлов в локальный репозиторий

18. Загружаю файлы на Github

```
askuznecova@dk1n22 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab04/report $ git add .
askuznecova@dk1n22 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab04/report $ git commit -am 'feat(main): make course structure'
[master 4a95aeb] feat(main): make course structure
2 files changed, 32 insertions(+)
create mode 100644 labs/lab04/report/hello.asm
create mode 100644 labs/lab04/report/lab4.asm
askuznecova@dk1n22 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab04/report $ git push
Перечисление объектов: 7, готово.
Подсчет объектов: 100% (7/7), готово.
При сжатии изменений используется до 6 потоков
Сканирование объектов: 100% (4/4), готово.
Запись объектов: 100% (6/6), 899 байтов | 449.00 КиБ/с, готово.
Всего 6 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:askuznecova/study_2023-2024_arh-pc.git
 67fc3a8..4a95aeb master -> master
```

Рис. 3.18: Загрузка файлов на гитхаб

4 Выводы

В ходе выполнения работы, я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.