

# Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютера

Калашникова Ольга Сергеевна

## Содержание

Цель работы	2
Задание	2
Выполнение лабораторной работы	3
Программа Hello world! . . . . .	3
Транслятор NASM . . . . .	5
Расширенный синтаксис командной строки NASM . . . . .	5
Компоновщик LD . . . . .	5
Запуск исполняемого файла . . . . .	6
Задание для самостоятельной работы . . . . .	6
Выводы	9

## Список иллюстраций

1	Создание каталога . . . . .	3
2	Переход в каталог . . . . .	3
3	Создание текстового файла . . . . .	4
4	Открытие файла . . . . .	4
5	Ввод программы . . . . .	4
6	Компиляция и проверка . . . . .	5
7	Компиляция и проверка . . . . .	5
8	Выполнение компоновки и проверка 1 . . . . .	5
9	Выполнение компоновки и проверка 2 . . . . .	6
10	Запуск команды, выводящей “Hello World” . . . . .	6
11	Создание нового файла . . . . .	6
12	Открытие файла lab4.asm . . . . .	7

13	Изменения в файле . . . . .	7
14	Компиляция и проверка . . . . .	8
15	Компиляция и проверка . . . . .	8
16	Выполнение компоновки и проверка 1 . . . . .	8
17	Выполнение компоновки и проверка 2 . . . . .	8
18	Запуск «lab4» . . . . .	9
19	Копирование файлов в локальный репозиторий и проверка . .	9
20	Загрузка файлов на github . . . . .	9

## Список таблиц

## Цель работы

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

## Задание

1. Создать каталог для работы с программами на языке ассемблера NASM.
2. Создать текстовый файл с именем «hello.asm» и открыть этот файл с помощью текстового редактора.
3. Ввести в текстовый файл нужный текст.
4. Оттранслировать полученный текст программы «hello.asm» в объектный файл.
5. Выполнить компоновку объектного файла и запустить получившийся исполняемый файл.
6. Создать копию файла «hello.asm» с именем «lab4.asm» и проделать с ним те же действия, что и с файлом «hello.asm».

7. Скопировать файлы «hello.asm» и «lab4.asm» в локальный репозиторий в каталог «~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/».
8. Загрузить файлы на Github.

## Выполнение лабораторной работы

### Программа Hello world!

Создаём каталог для работы с программами на языке ассемблера NASM  
( `mkdir -p ~/work/arch-pc/lab04` ) (рис. @fig:001)

```
yskalashnikova@dk6n62:~$ mkdir -p ~/work/arch-pc/lab04
yskalashnikova@dk6n62:~$
```

Рис. 1: Создание каталога

Переходим в созданный каталог (`cd`) (рис. [-@fig:002])

```
yskalashnikova@dk6n62:~$ cd ~/work/arch-pc/lab04
yskalashnikova@dk6n62:~/work/arch-pc/lab04$ █
```

Рис. 2: Переход в каталог

Создаём текстовый файл с именем «hello.asm» (`touch hello.asm`)(рис. [-@fig:003])

```
oskashnikova@dk6n62:~/work/arch-pc/lab04$ touch hello.asm
oskashnikova@dk6n62:~/work/arch-pc/lab04$
```

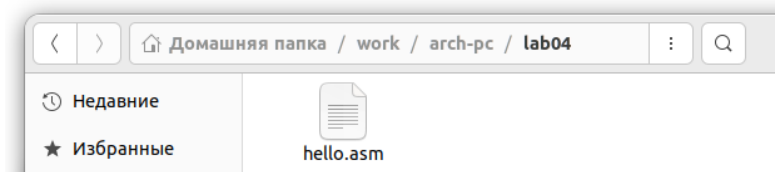


Рис. 3: Создание текстового файла

Откроем этот файл с помощью текстового редактора gedit (gedit hello.asm) (рис. [-@fig:004])

```
oskashnikova@dk6n62:~/work/arch-pc/lab04$ gedit hello.asm
oskashnikova@dk6n62:~/work/arch-pc/lab04$
```

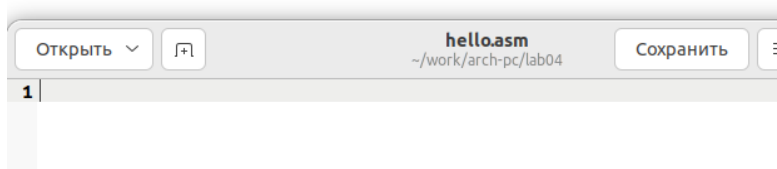


Рис. 4: Открытие файла

Вводим в открытый файл приведённый текст (рис. [-@fig:005])



Рис. 5: Ввод программы

## Транслятор NASM

Компилируем приведённый выше текст программы «Hello World» (`nasm -f elf hello.asm`) и проверяем (`ls`) (рис. [-@fig:006])

```
oskashnikova@dk6n62:~/work/arch-pc/lab04$ nasm -f elf hello.asm
oskashnikova@dk6n62:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
oskashnikova@dk6n62:~/work/arch-pc/lab04$ █
```

Рис. 6: Компиляция и проверка

## Расширенный синтаксис командной строки NASM

Компилируем исходный файл (`nasm -o obj.o -f elf -g -l list.lst hello.asm`) и проверяем (`ls`) (рис. [-@fig:007])

```
oskashnikova@dk6n62:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
oskashnikova@dk6n62:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
oskashnikova@dk6n62:~/work/arch-pc/lab04$ █
```

Рис. 7: Компиляция и проверка

## Компоновщик LD

Передаём объектный файл «hello.o» на обработку компоновщику (`ld -m elf_i386 hello.o -o hello`) и проверяем (`ls`) (рис. [-@fig:008])

```
oskashnikova@dk6n62:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
oskashnikova@dk6n62:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  obj.o
oskashnikova@dk6n62:~/work/arch-pc/lab04$ █
```

Рис. 8: Выполнение компоновки и проверка 1

Выполняем следующую команду (`ld -m elf_i386 obj.o -o main`) и проверяем (`ls`) (рис. [-@fig:009])

```
oskashnikova@dk6n62:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
oskashnikova@dk6n62:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o list.lst main obj.o
oskashnikova@dk6n62:~/work/arch-pc/lab04$
```

Рис. 9: Выполнение компоновки и проверка 2

Вопросы: 1) Какое имя будет иметь исполняемый файл? Исполняемый файл будет иметь имя «main» 2) Какое имя имеет объектный файл из которого собран этот исполняемый файл? Объектный файл из которого собран этот исполняемый называется «obj.o»

## Запуск исполняемого файла

Запускаем на выполнение созданный исполняемый файл, находящийся в текущем каталоге (./hello) (рис. [-@fig:010])

```
oskashnikova@dk6n62:~/work/arch-pc/lab04$ ./hello
Hello world!
oskashnikova@dk6n62:~/work/arch-pc/lab04$ █
```

Рис. 10: Запуск команды, выводящей “Hello World”

## Задание для самостоятельной работы


В каталоге ~/work/arch-pc/lab04 с помощью команды cp создаём копию файла hello.asm с именем lab4.asm (рис. [-@fig:011])

```
oskashnikova@dk6n62:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
oskashnikova@dk6n62:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm list.lst main obj.o
oskashnikova@dk6n62:~/work/arch-pc/lab04$ █
```

Рис. 11: Создание нового файла

С помощью текстового редактора «gedit» открываем файл «lab4.asm» (рис. [-@fig:012])

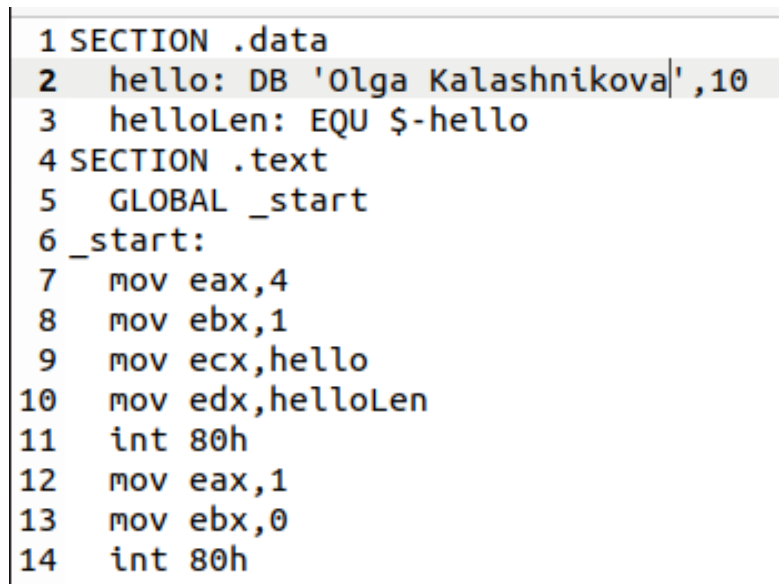
```
oskalashnikova@dk6n62:~/work/arch-pc/lab04$ gedit lab4.asm
oskalashnikova@dk6n62:~/work/arch-pc/lab04$
```



```
1 SECTION .data
2  hello: DB 'Hello world!',10
3  helloLen: EQU $-hello
4 SECTION .text
5  GLOBAL _start
6 _start:
```

Рис. 12: Открытие файла lab4.asm

Вносим изменение в текстовый файл так, чтобы на экран выводилась строка с нашим именем и фамилией (рис. [-@fig:013])



```
1 SECTION .data
2  hello: DB 'Olga Kalashnikova|',10
3  helloLen: EQU $-hello
4 SECTION .text
5  GLOBAL _start
6 _start:
7  mov eax,4
8  mov ebx,1
9  mov ecx,hello
10 mov edx,helloLen
11 int 80h
12 mov eax,1
13 mov ebx,0
14 int 80h
```

Рис. 13: Изменения в файле

Компилируем текст программы «Olga Kalashnikova» (`nasm -f elf lab4.asm`) и проверяем (`ls`) (рис. [-@fig:014])

```
oskalashnikova@dk6n62:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
oskalashnikova@dk6n62:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o
oskalashnikova@dk6n62:~/work/arch-pc/lab04$
```

Рис. 14: Компиляция и проверка

Компилируем исходный файл (`nasm -o obj4.o -f elf -g -l list4.lst lab4.asm`) и проверяем (`ls`) (рис. [-@fig:015])

```
oskalashnikova@dk6n62:~/work/arch-pc/lab04$ nasm -o obj4.o -f elf -g -l list4.lst lab4.asm
oskalashnikova@dk6n62:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm lab4.o list4.lst list.lst main obj4.o obj.o
oskalashnikova@dk6n62:~/work/arch-pc/lab04$
```

Рис. 15: Компиляция и проверка

Передаём объектный файл «lab.o» на обработку компоновщику (`ld -m elf_i386 lab4.o -o lab4`) и проверяем (`ls`) (рис. [-@fig:016])

```
oskalashnikova@dk6n62:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
oskalashnikova@dk6n62:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list4.lst list.lst main obj4.o obj.o
oskalashnikova@dk6n62:~/work/arch-pc/lab04$
```

Рис. 16: Выполнение компоновки и проверка 1

Передаём объектный файл «obj4.o» на обработку компоновщику с помощью (`ld -m elf_i386 obj4.o -o main4`) и проверяем (`ls`) (рис. [-@fig:017])

```
oskalashnikova@dk6n62:~/work/arch-pc/lab04$ ld -m elf_i386 obj4.o -o main4
oskalashnikova@dk6n62:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list4.lst list.lst main main4 obj4.o
oskalashnikova@dk6n62:~/work/arch-pc/lab04$
```

Рис. 17: Выполнение компоновки и проверка 2

Запускаем на выполнение созданный исполняемый файл, находящийся в текущем каталоге (`./lab4`) (рис. [-@fig:018])



```
oskalashnikova@dk6n62:~/work/arch-pc/lab04$ ./lab4
Olga Kalashnikova
oskalashnikova@dk6n62:~/work/arch-pc/lab04$ █
```

Рис. 18: Запуск «lab4»

Скопируем файлы hello.asm и lab4.asm в локальный репозиторий в каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/ (ср) и проверяем (ls) (рис. [-@fig:019])

```
oskalashnikova@dk6n62:~/work/arch-pc/lab04$ cp lab4.asm ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/
oskalashnikova@dk6n62:~/work/arch-pc/lab04$ cp hello.asm ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/
oskalashnikova@dk6n62:~/work/arch-pc/lab04$ cd
oskalashnikova@dk6n62:~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/
oskalashnikova@dk6n62:~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04$ ls
hello.asm lab4.asm presentation report
```

Рис. 19: Копирование файлов в локальный репозиторий и проверка

Загружаем файлы на github (рис. [-@fig:020])

```
oskalashnikova@dk6n62:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git add .
oskalashnikova@dk6n62:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit -am 'add hello.asm and lab4.asm'
[master 4ffcc81] add hello.asm and lab4.asm
3 files changed, 29 insertions(+), 21 deletions(-)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
oskalashnikova@dk6n62:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 13, готово.
Подсчет объектов: 100% (13/13), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов: 100% (8/8), готово.
Запись объектов: 100% (8/8), 888 байтов | 222.00 Киб/с, готово.
Всего 8 (изменений 5), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (5/5), completed with 4 local objects.
To github.com:lacrimell/study_2023-2024_arh-pc.git
2541478..4ffcc81 master -> master
oskalashnikova@dk6n62:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ █
```

Рис. 20: Загрузка файлов на github

## Выводы

В ходе выполнения лабораторной работы мы освоили процедуры компиляции и сборки программ, написанных на ассемблере NASM.