

# **Отчёт по лабораторной работе №4**

**Дисциплина: архитектура компьютера**

Сидельников Андрей Владимирович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Программа Hello world! . . . . .	6
2.2	Работа с транслятором NASM . . . . .	7
2.3	Работа с расширенным синтаксисом командной строки NASM . .	7
2.4	Работа с компоновщиком LD . . . . .	8
2.5	Запуск исполняемого файла . . . . .	8
<b>3</b>	<b>Выполнение заданий для самостоятельной работы</b>	<b>9</b>
<b>4</b>	<b>Выводы</b>	<b>12</b>

## **Список таблиц**

# Список иллюстраций

2.1	Создание файла hello.asm . . . . .	6
2.2	Заполнение файла . . . . .	7
2.3	Компиляция текста программы . . . . .	7
2.4	Компиляция текста программы . . . . .	8
2.5	Передача объектного файла на обработку компоновщику . . . . .	8
figno:1.	. . . . .	8
2.6	Передача объектного файла на обработку компоновщику . . . . .	8
2.7	Запуск исполняемого файла . . . . .	8
3.1	Создание копии файла . . . . .	9
3.2	Изменение программы . . . . .	9
3.3	Компиляция текста программы . . . . .	10
3.4	Передача объектного файла на обработку компоновщику . . . . .	10
3.5	Запуск исполняемого файла . . . . .	10
figno:2.	. . . . .	11
3.6	Копирование файлов . . . . .	11
3.7	Удаление файлов . . . . .	11
3.8	Отправка файлов . . . . .	11

# 1 Цель работы

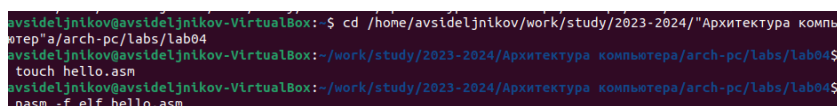
Освоить процедуры компиляции и сборки программ, написанных на ассемблере NASM.

## 2 Выполнение лабораторной работы

### 2.1 Программа Hello world!

Так как я изначально выполнял не в той директории, то создание нужной директории будет в конце.

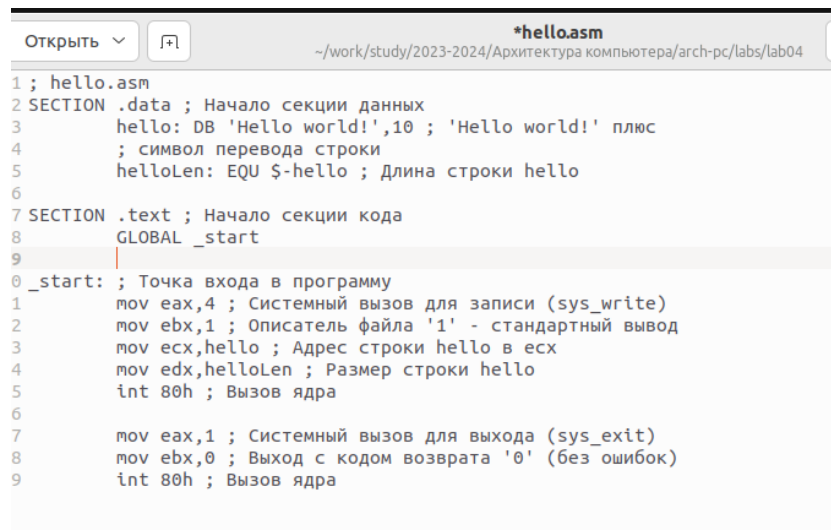
Перемещаюсь в каталог, в котором буду работать и создаю пустой файл hello.asm (рис. 2.1).



```
avsideljnikov@avsideljnikov-VirtualBox: ~$ cd /home/avsideljnikov/work/study/2023-2024/"Архитектура компьютер"
avsideljnikov@avsideljnikov-VirtualBox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ touch hello.asm
avsideljnikov@avsideljnikov-VirtualBox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ nasm -f elf hello.asm
```

Рис. 2.1: Создание файла hello.asm

Открываю созданный файл и заполняю файл, вставляя в него программу для вывода “Hello word!” (рис. 2.2).

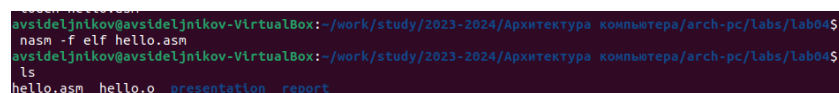


```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3     hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4             ; символ перевода строки
5     helloLen: EQU $-hello ; Длина строки hello
6
7 SECTION .text ; Начало секции кода
8     GLOBAL _start
9
10 _start: ; Точка входа в программу
11     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
```

Рис. 2.2: Заполнение файла

## 2.2 Работа с транслятором NASM

Превращаю текст программы для вывода “Hello world!” в объектный код с помощью транслятора NASM, используя команду `nasm -f elf hello.asm`: действительно, создан файл “hello.o” (рис. 2.3).



```
avsidejnikov@avsidejnikov-VirtualBox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
nasm -f elf hello.asm
avsidejnikov@avsidejnikov-VirtualBox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
ls
hello.asm hello.o presentation report
```

Рис. 2.3: Компиляция текста программы

## 2.3 Работа с расширенным синтаксисом командной строки NASM

Ввожу команду, которая скомпилирует файл `hello.asm` в файл `obj.o` и с помощью утилиты `ls` правильность выполнения команды (рис. 2.4).

```
avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
nasm -o obj.o -f elf -g -l list.lst hello.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
ls
hello.asm hello.o list.lst obj.o presentation report
```

Рис. 2.4: Компиляция текста программы

## 2.4 Работа с компоновщиком LD

Передаю объектный файл hello.o на обработку компоновщику LD, чтобы получить исполняемый файл hello. Далее проверяю с помощью утилиты ls правильность выполнения команды (рис. 2.5).

```
avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
ld -m elf_i386 hello.o -o hello
avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
ls
hello hello.asm hello.o list.lst obj.o presentation report
```

Рис. 2.5: Передача объектного файла на обработку компоновщику

Выполняю команду **ld -m elf\_i386 obj.o -o main**. Исполняемый файл будет иметь имя main, т.к. после ключа -o было задано значение main. Объектный файл, из которого собран этот исполняемый файл, имеет имя obj.o (рис. 2.6).

```
avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
ld -m elf_i386 obj.o -o main

avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
ls
hello hello.asm hello.o list.lst main obj.o presentation report
```

Рис. 2.6: Передача объектного файла на обработку компоновщику

## 2.5 Запуск исполняемого файла

Запускаю на выполнение созданный исполняемый файл hello (рис. 2.7).

```
avsideljnikov@avsideljnikov-VirtualBox:~$
./hello
Hello world!
```

Рис. 2.7: Запуск исполняемого файла



### 3 Выполнение заданий для самостоятельной работы

С помощью утилиты `cp` создаю в текущем каталоге копию файла `hello.asm` с именем `lab4.asm` (рис. 3.1).

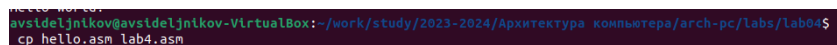


Рис. 3.1: Создание копии файла

Открываю файл `lab4.asm` и вношу изменения в программу так, чтобы она выводила мои имя и фамилию (рис. 3.2).

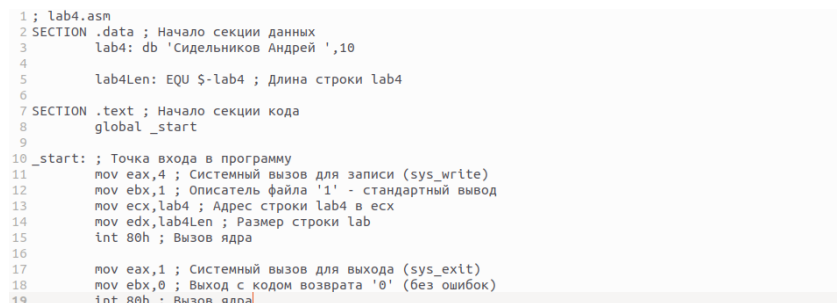


Рис. 3.2: Изменение программы

“; lab4.asm SECTION .data ; Начало секции данных lab4: db ‘Сидельников Андрей’,10

lab4Len: EQU \$-lab4 ; Длина строки lab4

SECTION .text ; Начало секции кода global \_start

\_start: ; Точка входа в программу mov eax,4 ; Системный вызов для записи (sys\_write) mov ebx,1 ; Описатель файла '1' - стандартный вывод mov ecx,lab4 ; Адрес строки lab4 в ecx mov edx,lab4Len ; Размер строки lab int 80h ; Вызов ядра

mov eax,1 ; Системный вызов для выхода (sys\_exit)

mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)

int 80h ; Вызов ядра ``

Компилирую текст программы в объектный файл и проверяю с помощью утилиты ls, что файл lab4.o создан (рис. 3.3).

```
avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
nasm -f elf lab4.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o presentation report
```

Рис. 3.3: Компиляция текста программы

Передаю объектный файл lab4.o на обработку компоновщику LD, чтобы получить исполняемый файл lab4 (рис. 3.4).

```
avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
ld -m elf_i386 lab4.o -o lab4
avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o presentation report
```

Рис. 3.4: Передача объектного файла на обработку компоновщику

Запускаю исполняемый файл lab4, на экран действительно выводятся мои имя и фамилия (рис. 3.5).

```
avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
./lab4
Сидельников Андрей
```

Рис. 3.5: Запуск исполняемого файла

Так как я делал не в той директории, то исправлю данный момент.

Создаю каталог /work/arch-pc/lab04, затем копирую в него всё из /work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04, используя '\*', и с помощью команды ls проверяю успешное копирование (рис. 3.6)

```
avsideljnikov@avsideljnikov-VirtualBox:~$ mkdir -p ~/work/arch-pc/lab04

avsideljnikov@avsideljnikov-VirtualBox:~$ cd ~/work/arch-pc/lab04
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab04$ cd ~/work/study/2023-2024/Архитектура комп
ьютера/arch-pc/labs/lab04
avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
$ cp * ~/work/arch-pc/lab04
cp: не указан -r; пропускается каталог 'presentation'
cp: не указан -r; пропускается каталог 'report'
avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
$ cd ~/work/arch-pc/lab04
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4  lab4.asm  lab4.o  list.lst  main  obj.o
```

Рис. 3.6: Копирование файлов

Удаляю ненужные файлы в каталоге /work/study/2023-2024/Архитектура комп-  
ьютера/arch-pc/labs/lab04 с помощью команды `rm` (рис. 3.7)

```
avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
$ rm hello hello.o lab4 lab4.o list.lst main obj.o
avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
$ ls
hello.asm  lab4.asm  presentation  report
```

Рис. 3.7: Удаление файлов

Отправляю на Github (рис. 3.8)

```
avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git add .
avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit
-m ""
Отмена коммита из-за пустого сообщения коммита.
avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit
-m "d"
[master 96fb65c] d
2 files changed, 38 insertions(+)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 1.03 КиБ | 1.03 МБ/с, готово.
Всего 6 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:Avsideljnikov/study_2023-2024_arch-pc.git
fd4db12..96fb65c master -> master
```

Рис. 3.8: Отправка файлов

## 4 Выводы

При выполнении данной лабораторной работы я освоил процедуры компиляции и сборки программ, написанных на ассемблере NASM.