

Лабораторная работа №5 Создание и процесс обработки программ на языке ассемблера NASM

**Факультет физико-математических и естественных наук Кафедра
прикладной информатики и теории вероятностей. Дисциплина:
Архитектура ЭВМ**

Осокин Георгий Иванович. НММбд-02-22

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Программа “Hello world”	6
2.1.1	Создадим файл <code>hello.asm</code>	6
2.1.2	Воспользуемся транслятором NASM	7
2.1.3	Слинкуем <code>obj.o</code> спомощью утилиты <code>ld</code>	8
2.1.4	Откроем исполняемый файл через дизассемблер	8
2.1.5	Слинкуем в файл с другим именем с помощью опции <code>-o</code> . .	9
2.1.6	Удалим лишние файлы	9
2.1.7	Запустим исполняемый файл	9
2.2	Задания для самостоятельной работы	10
2.2.1	Создадим файл <code>lab05.asm</code>	10
2.2.2	Странслируем в объектный файл <code>obj-lab05.o</code>	10
2.2.3	Создадим исполняемый файл с помощью <code>ld</code>	11
2.2.4	Запустим <code>./lab05</code>	11
2.3	Скопируем папку в рабочее пространство и отправим на GitHub .	11
3	Выводы	13

Список иллюстраций

2.1	Переход в директорию lab05	6
2.2	Создание lab05.asm	6
2.3	Открытие файла через текстовый редактор Emacs	7
2.4	Код на ассемблере введенный в файл	7
2.5	Трансляция в объектный файл	8
2.6	Создание объектного файла с другим именем и создание листинга	8
2.7	Линковка с помощью ld	8
2.8	Просмотр исполняемого файла через дизассемблер radare2	9
2.9	Создание исполняемого файла с именем main	9
2.10	Удаление лишних файлов	9
2.11	Исполнение ./main	9
2.12	Создание файла lab05.asm	10
2.13	Содержимое файла lab05.asm	10
2.14	Трансляция lab05	10
2.15	Создание исполняемого файла main-lab5	11
2.16	Исполнение lab05	11
2.17	Копирование в рабочее пространство	11
2.18	Создание коммита и отправка файлов на GitHub	12

Список таблиц

1 Цель работы

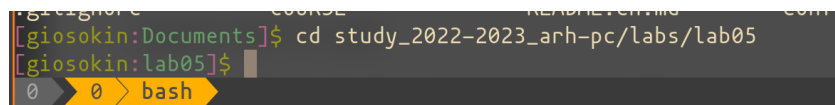
Освоить процедуры компиляции и сборки программ, написанных на ассемблере NASM.

2 Выполнение лабораторной работы

2.1 Программа “Hello world”

2.1.1 Создадим файл `hello.asm`

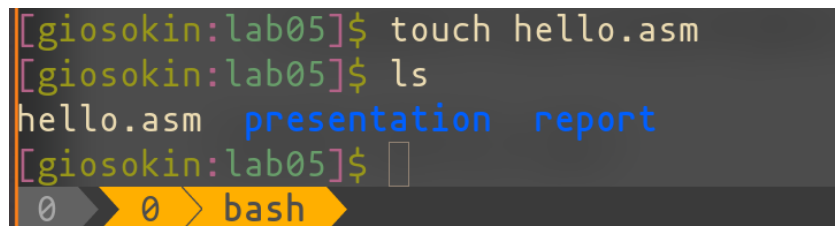
Перейдем в директорию `lab05`

A terminal window showing the command `cd study_2022-2023_arh-pc/labs/lab05` being executed. The prompt changes from `[giosokin:Documents]$` to `[giosokin:lab05]$`.

```
[giosokin:Documents]$ cd study_2022-2023_arh-pc/labs/lab05
[giosokin:lab05]$
```

Рис. 2.1: Переход в директорию `lab05`

Создадим файл `lab05.asm` и откроем его в текстовом редакторе *Emacs*

A terminal window showing the command `touch hello.asm` being executed, followed by `ls`. The output of `ls` is `hello.asm presentation report`.

```
[giosokin:lab05]$ touch hello.asm
[giosokin:lab05]$ ls
hello.asm presentation report
[giosokin:lab05]$
```

Рис. 2.2: Создание `lab05.asm`

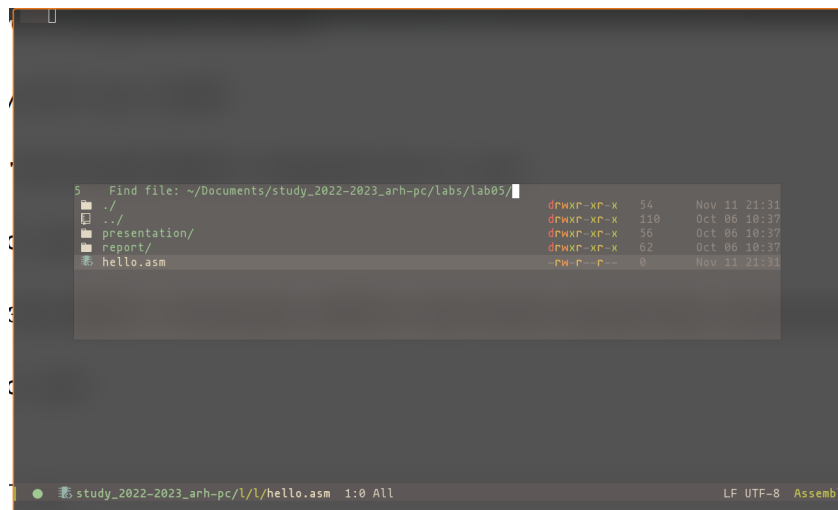


Рис. 2.3: Открытие файла через текстовый редактор Emacs

Введем исходный текст в файл

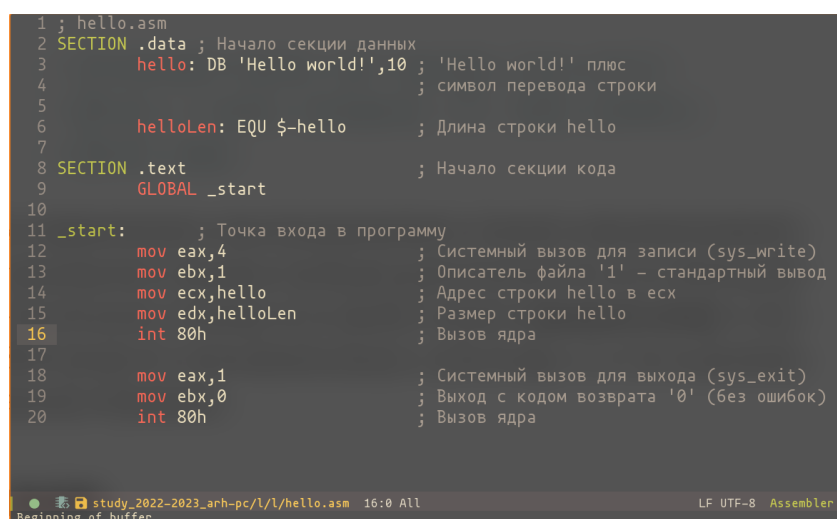


Рис. 2.4: Код на ассемлере введенный в файл

2.1.2 Воспользуемся транслятором NASM

Странслируем исходный код в объектный файл с помощью команды `nasm -f elf hello.asm`

```
[giosokin:lab05]$ nasm -f elf hello.asm
[giosokin:lab05]$ ls
hello.asm hello.asm~ hello.o presentation report
[giosokin:lab05]$
```

Рис. 2.5: Трансляция в объектный файл

Создадим объектный файл с другим именем, с помощью опции -o и сгенерируем листинг, с помощью опции -l

```
[giosokin:lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[giosokin:lab05]$ ls
#hello.asm# hello.asm hello.o list.lst obj.o presentation report
[giosokin:lab05]$
```

Рис. 2.6: Создание объектного файла с другим именем и создание листинга

2.1.3 Слинкуем obj.o с помощью утилиты ld

Исполним команду `ld -m elf_i386 hello.o -o hello`, что б получить исполняемый файл `hello`

```
[giosokin:lab05]$ ld -m elf_i386 hello.o -o hello
[giosokin:lab05]$ ls
#hello.asm# hello.asm hello.o obj.o report
hello hello.asm~ list.lst presentation
[giosokin:lab05]$
```

Рис. 2.7: Линковка с помощью ld

2.1.4 Откроем исполняемый файл через дизассемблер

Мы получили исполняемый файл. Из любопытства, откроем его через дизассемблер *Radare2*



Рис. 2.8: Просмотр исполняемого файла через дизасемблер radare2

Заметим, что код на ассемблере представленный здесь, очень похож на наш изначальный и даже сохранились названия некоторых “переменных”

2.1.5 Слинкуем в файл с другим именем с помощью опции -o

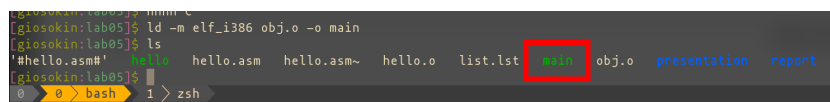


Рис. 2.9: Создание исполняемого файла с именем main

2.1.6 Удалим лишние файлы



Рис. 2.10: Удаление лишних файлов

2.1.7 Запустим исполняемый файл

Наберем в консоли ./main и увидим вывод.

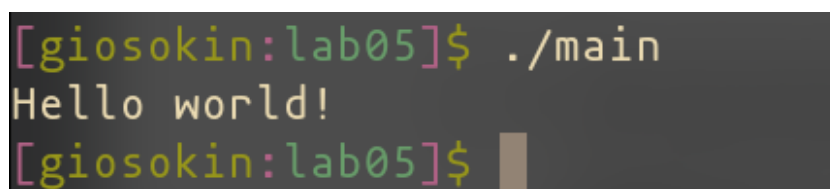


Рис. 2.11: Исполнение ./main

2.2 Задания для самостоятельной работы

2.2.1 Создадим файл lab05.asm

```
[giosokin:lab05]$ touch lab5.asm
[giosokin:lab05]$ 
0 > bash 1 > zsh
```

Рис. 2.12: Создание файла lab05.asm

Скопируем исходный код из hello.asm и модифицируем его через текстовый редактор *Emacs*

```
1 ; lab5.asm
2
3 SECTION .data ; Начало секции данных
4     string: DB 'Осокин Георгий',10 ; 'Осокин Георгий' плюс
5                                     ; символ перевода строки
6
7     stringLen: EQU $-string ; Длина строки string
8
9 SECTION .text ; Начало секции кода
10    GLOBAL _start
11
12    _start: ; Точка входа в программу
13        mov eax,4 ; Системный вызов для записи (sys_write)
14        mov ebx,1 ; Описатель файла '1' - стандартный вывод
15        mov ecx,string ; Адрес строки string в ecx
16        mov edx,stringLen ; Размер строки string
17        int 80h ; Вызов ядра
18
19        mov eax,1 ; Системный вызов для выхода (sys_exit)
20        mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
21        int 80h ; Вызов ядра
```

Рис. 2.13: Содержимое файла lab05.asm

2.2.2 Транслируем в объектный файл obj-lab05.o

Воспользуемся транслятором NASM что бы получить объектный файл

```
[giosokin:lab05]$ nasm -o obj-lab5.o -f elf -g -l list-lab5.lst lab5.asm
[giosokin:lab05]$ ls
#hello.asm#  hello.asm~  lab5.asm~  list.lst  obj-lab5.o  presentation
hello.asm    lab5.asm    list-lab5.lst  main      obj.o       report
[giosokin:lab05]$ 
0 > bash 1 > zsh
```

Рис. 2.14: Трансляция lab05

2.2.3 Создадим исполняемый файл с помощью ld

```
hello.asm  lab5.asm  list=lab5.lst  obj.o  report
[giosokin:lab05]$ ld -m elf_i386 obj-lab5.o -o main-lab5
[giosokin:lab05]$ ls
#hello.asm#  hello.asm~  lab5.asm~  list.lst  main-lab5  obj.o  report
hello.asm  lab5.asm  list-lab5.lst  main-lab5  presentation
[giosokin:lab05]$
```

Рис. 2.15: Создание исполняемого файла main-lab5

2.2.4 Запустим ./lab05

```
hello.asm  lab5.asm  list-
[giosokin:lab05]$ ./main-lab5
Осокин Георгий
[giosokin:lab05]$
```

Рис. 2.16: Исполнение lab05

Как видим, в консоль выводится строка, которую мы задали (фамилия с именем)

2.3 Скопируем папку в рабочее пространство и отправим на GitHub

```
[giosokin:lab05]$ sudo cp -r ../lab05 /home/giosokin/work/study/2022-2023/Архитектура\ компьютера/study_2022-2023_arh-pc/labs/
[giosokin:lab05]$
```

Рис. 2.17: Копирование в рабочее пространство

Создадим коммит с сообщением “start lab05” и запустим.

```

[giosokin:lab05]$ git add .
[giosokin:lab05]$ git commit -m "start lab05"
[master 9214e6e] start lab05
25 files changed, 80 insertions(+)
create mode 100644 labs/lab05/hello.asm
create mode 100644 labs/lab05/lab5.asm
create mode 100644 labs/lab05/list-lab5.lst
create mode 100644 labs/lab05/list.lst
create mode 100755 labs/lab05/main
create mode 100755 labs/lab05/main-lab5
create mode 100644 labs/lab05/obj-lab5.o
create mode 100644 labs/lab05/obj.o
create mode 100644 labs/lab05/report/image/1.png
create mode 100644 labs/lab05/report/image/10.png
create mode 100644 labs/lab05/report/image/11.png
create mode 100644 labs/lab05/report/image/12.png
create mode 100644 labs/lab05/report/image/13.png
create mode 100644 labs/lab05/report/image/14.png
create mode 100644 labs/lab05/report/image/15.png
create mode 100644 labs/lab05/report/image/16.png
create mode 100644 labs/lab05/report/image/17.png
create mode 100644 labs/lab05/report/image/2.png
create mode 100644 labs/lab05/report/image/3.png
create mode 100644 labs/lab05/report/image/4.png
create mode 100644 labs/lab05/report/image/5.png
create mode 100644 labs/lab05/report/image/6.png
create mode 100644 labs/lab05/report/image/7.png
create mode 100644 labs/lab05/report/image/8.png
create mode 100644 labs/lab05/report/image/9.png
[giosokin:lab05]$ git push
Enumerating objects: 36, done.
Counting objects: 100% (36/36), done.
Delta compression using up to 8 threads
Compressing objects: 100% (31/31), done.
0 > 0 > bash 1 > [tmux] 2 > zsh

```

Рис. 2.18: Создание коммита и отправка файлов на GitHub

3 Выводы

В ходе данной лабораторной работы мы научились переводить программы на языке ассемблера NASM в исполняемый файл с помощью трансляции и последующей линковки. Мы написали программу на NASM, которая выводит в консоль нашу фамилию и имя. А также проверили, что в исполняемом файле в дизасемблированном виде находится очень похожий на наш изначальный код на языке ассемблера.