

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

дисциплина: Архитектура компьютера

Студент: Татур Платон Андреевич

Группа: НКАбд-05-25

МОСКВА

2025 г

4.1. Цель работы

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

4.3. Порядок выполнения лабораторной работы

4.3.1. Программа Hello world!

```
patatur@platonpc:~$ mkdir -p ~/work/arch-pc/lab04
patatur@platonpc:~$ cd ~/work/arch-pc/lab04
patatur@platonpc:~/work/arch-pc/lab04$ 
patatur@platonpc:~/work/arch-pc/lab04$ touch hello.asm
patatur@platonpc:~/work/arch-pc/lab04$ gedit hello.asm
```

Рис. 1

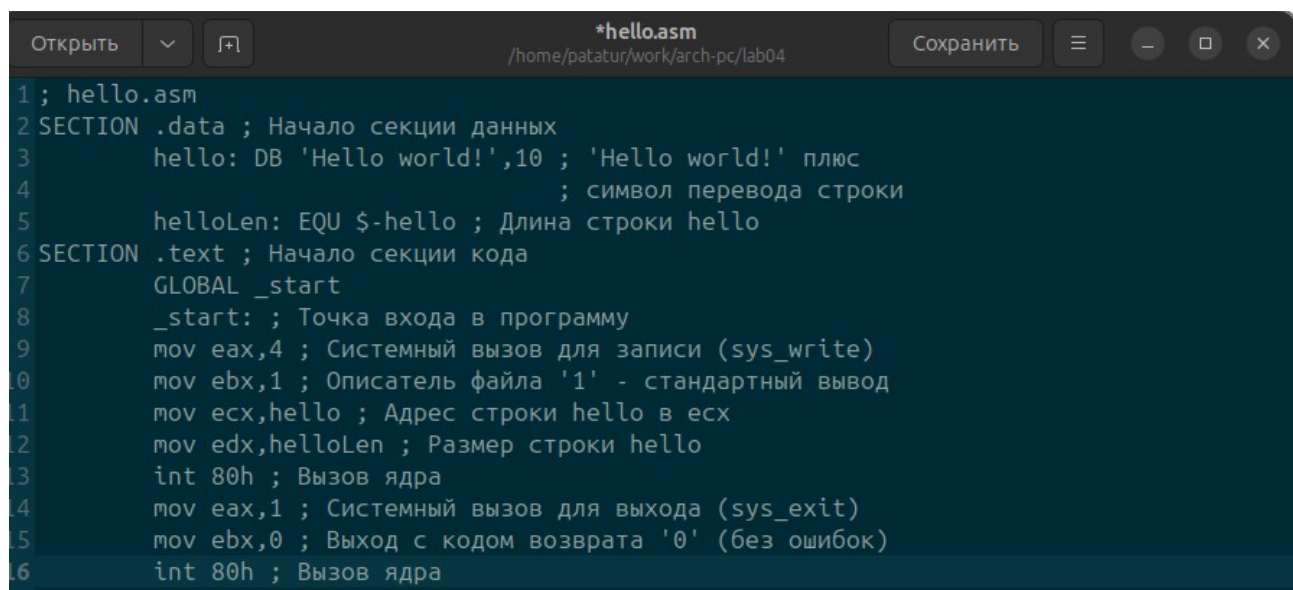
1) Создал каталог для работы с программами на языке ассемблера NASM

2) Перешел в созданный каталог

3) Создал текстовый файл с именем hello.asm

4) Открыл файл с помощью текстового редактора gedit

(см Рис. 1)



The screenshot shows a text editor window titled '*hello.asm' with the file path '/home/patatur/work/arch-pc/lab04'. The editor contains the following assembly code:

```
1; hello.asm
2SECTION .data ; Начало секции данных
3    hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4                ; символ перевода строки
5    helloLen: EQU $-hello ; Длина строки hello
6SECTION .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. 2

5) Ввел в него текст программы

(см Рис. 2)

4.3.2. Транслятор NASM

```
patatur@platonpc:~/work/arch-pc/lab04$ nasm -f elf hello.asm
patatur@platonpc:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
```

Рис. 3

- 1) С помощью NASM, преобразовал текст программы из файла hello.asm в объектный код, который записался в файл hello.o
- 2) С помощью команды ls проверил, что объектный файл был создан, объектный файл имеет название «hello.o»

(см Рис. 3)

4.3.3. Расширенный синтаксис командной строки NASM

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

Рис. 4

- 1) Скомпилировал исходный файл hello.asm в obj. При этом формат выходного файла elf, и в него включены символы для отладки, создал файл листинга list.lst
- 2) Проверил с помощью ls, что файлы были созданы

(см Рис. 4)

4.4. Компоновщик LD

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

Рис. 5

- 1) Передал объектный файл на обработку компоновщику
- 2) Проверил с помощью ls, что исполняемый файл hello был создан

(см Рис. 5)

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

Рис. 6

- 1) Исполняемый файл имеет имя “main”
- 2) Объектный файл из которого собран исполняемый файл “main” имеет название “obj.o”

(см Рис. 6)

4.4.1. Запуск исполняемого файла



```
patatur@platonpc:~/work/arch-pc/lab04$ ./hello
Hello world!
```

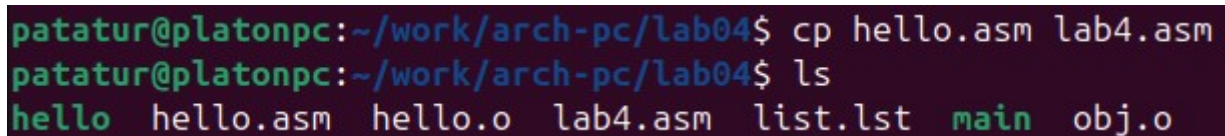
Рис. 7

1) Запустил на выполнение созданный исполняемый файл

(см Рис. 7)

4.5. Задание для самостоятельной работы

1. В каталоге `~/work/arch-pc/lab04` с помощью команды `cp` создайте копию файла `hello.asm` с именем `lab4.asm`



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

Рис. 8

1) Создал копию файла "hello.asm" с названием "lab4.asm"

2) Проверил с помощью `ls`, что файл был скопирован

(см Рис. 8)

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

```
patatur@platonpc:~/work/arch-pc/lab04$ gedit lab4.asm
```

Рис. 9.1

```
; hello.asm
SECTION .data ; Начало секции данных
    hello: DB 'Tatur Platon',10 ; 'Tatur Platon' плюс
                                ; символ перевода строки
    helloLen: EQU $-hello ; Длина строки hello
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,hello ; Адрес строки hello в ecx
    mov edx,helloLen ; Размер строки hello
    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
```

Рис. 9.2

- 1)Открыл файл “lab4.asm” в редакторе (см Рис. 9.1)
- 2)Изменил текст “Hello world” на своё имя и фамилию (см Рис. 9.2)

3. Оттранслируйте полученный текст программы *lab4.asm* в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.

```
patatur@platonpc:~/work/arch-pc/lab04$ nasm -o lab4.o -f elf -g -l listlab4.lst lab4.asm
```

Рис. 10.1

```
patatur@platonpc:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
patatur@platonpc:~/work/arch-pc/lab04$ ./lab4
Tatur Platon
```

Рис. 10.2

- 1)Оттранслировал полученный текст программы *lab4.asm* в объектный файл. (см Рис. 10.1)
- 2)Выполнил компоновку объектного файла и запустил получившийся исполняемый файл (см Рис. 10.2)

4. Скопируйте файлы *hello.asm* и *lab4.asm* в Ваш локальный репозиторий в каталог *~/work/study/2025-2026/"Архитектура компьютера"/arch-pc/labs/lab04/*.

Загрузите файлы на Github.

```
patatur@platonpc:~/work/arch-pc/lab04$ cp hello.asm ~/work/study/2025-2026/"Архитектура компьютера"/arch-pc/labs/lab04
patatur@platonpc:~/work/arch-pc/lab04$ cp lab4.asm ~/work/study/2025-2026/"Архитектура компьютера"/arch-pc/labs/lab04
```

Рис. 11.1

```
patatur@platonpc:~/work/arch-pc/lab04$ cd ~/work/study/2025-2026/"Архитектура компьютера"/arch-pc/labs/lab04
```

Рис. 11.2

```
patatur@platonpc:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab04$ git add .
patatur@platonpc:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab04$ git commit -am "add files"
[master 8443169] add files
 2 files changed, 32 insertions(+)
 create mode 100644 labs/lab04/hello.asm
 create mode 100644 labs/lab04/lab4.asm
patatur@platonpc:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab04$ git push
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 947 байтов | 947.00 КиБ/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:patatur4/study_2025-2026_arh-pc.git
 e06544c..8443169 master -> master
```

Рис. 11.3

1)Скопировал файлы hello.asm и lab4.asm в локальный репозиторий в каталог ~/work/study/2025-2026/"Архитектура компьютера"/arch-pc/labs/lab04 (см Рис. 11.1)

2)Загрузил файлы на github (см Рис. 11.2 и Рис. 11.3)

Вывод

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

