

Отчёта по лабораторной работе №4

**Создание и процесс обработки программ на языке ассемблера
NASM**

Джаллох Ишмаил

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Программа Hello world!	7
3.2	Транслятор NASM	8
3.3	Расширенный синтаксис командной строки NASM	8
3.4	Компоновщик LD	9
3.5	Запуск исполняемого файла	9
3.6	Задание для самостоятельной работы	10
4	Выводы	12

Список иллюстраций

3.1	Создаем каталоги с помощью команды <code>mkdir</code>	7
3.2	Переходим в каталог с помощью команды <code>cd</code>	7
3.3	Создаем текстовый файл <code>hello.asm</code>	7
3.4	Открываем файл и заполняем его по примеру	8
3.5	Используем команду <code>nasm</code>	8
3.6	Проверяем работу команды	8
3.7	Преобразуем файл <code>hello.asm</code> в <code>obj.o</code>	8
3.8	Проверяем создание файла командой <code>ls</code>	8
3.9	Используем команду <code>ld</code>	9
3.10	Используем команду <code>ls</code>	9
3.11	Используем команду <code>ld</code> , создавая файл <code>main</code>	9
3.12	Используем команду <code>ls</code>	9
3.13	Используем команду <code>./hello</code>	9
3.14	Используем команду <code>cp</code>	10
3.15	Открываем файл в текстовом редакторе	10
3.16	Редактируем файл для своего имени и фамилии	10
3.17	Прописываем команды для работы файла и запускаем программу	10
3.18	Копируем файлы в каталог с ЛР4	10
3.19	Загружаем файлы	11

Список таблиц

1 Цель работы

Освоить процедуры компиляции и сборки программ, познакомиться с языком ассемблера NASM.

2 Задание

Написать 2 программы(Hello world, lab4(Имя Фамилия))

3 Выполнение лабораторной работы

3.1 Программа Hello world!

Создаем каталог для работы с программами на языке ассемблера NASM (рис. fig. 3.1).

A terminal window with a dark background. The prompt is 'jalloh_ishmail@vbox:~'. The user enters the command 'mkdir -p ~/work/arch-pc/lab04'. The prompt changes to 'jalloh_ishmail@vbox:~\$' and a cursor is visible.

```
jalloh_ishmail@vbox:~$ mkdir -p ~/work/arch-pc/lab04
jalloh_ishmail@vbox:~$
```

Рис. 3.1: Создаем каталоги с помощью команды mkdir

Переходим в созданный каталог (рис. fig. 3.2).

A terminal window with a dark background. The prompt is 'jalloh_ishmail@vbox:~'. The user enters the command 'cd ~/work/arch-pc/lab04'. The prompt changes to 'jalloh_ishmail@vbox:~/work/arch-pc/lab04\$' and a cursor is visible.

```
jalloh_ishmail@vbox:~$ cd ~/work/arch-pc/lab04
jalloh_ishmail@vbox:~/work/arch-pc/lab04$
```

Рис. 3.2: Переходим в каталог с помощью команды cd

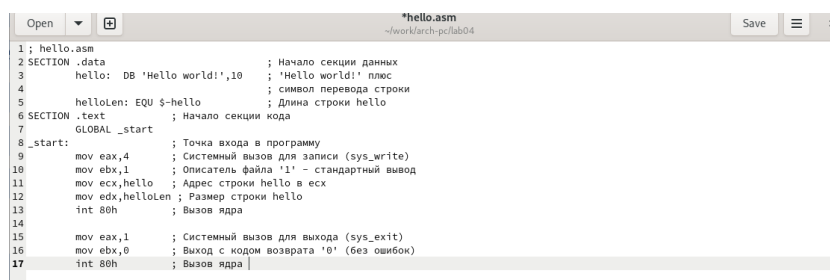
Создаем текстовый файл (рис. fig. 3.3).

A terminal window with a dark background. The prompt is 'jalloh_ishmail@vbox:~/work/arch-pc/lab04\$'. The user enters the command 'touch hello.asm'. The prompt changes to 'jalloh_ishmail@vbox:~/work/arch-pc/lab04\$' and a cursor is visible.

```
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ touch hello.asm
jalloh_ishmail@vbox:~/work/arch-pc/lab04$
```

Рис. 3.3: Создаем текстовый файл hello.asm

Открываем данный файл в текстовом редакторе (рис. fig. 3.4).

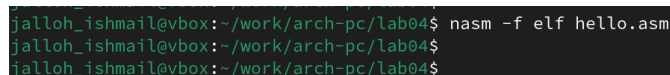


```
1; hello.asm
2SECTION .data                ; Начало секции данных
3    hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4                                ; символ перевода строки
5    hellolen: EQU $-hello      ; Длина строки hello
6SECTION .text                ; Начало секции кода
7    GLOBAL _start
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 ехх
12   mov edx,hellolen           ; Размер строки hello
13   int 80h                    ; Вызов ядра
14
15   mov eax,1                  ; Системный вызов для выхода (sys_exit)
16   mov ebx,0                  ; Выход с кодом возврата '0' (без ошибок)
17   int 80h                    ; Вызов ядра
```

Рис. 3.4: Открываем файл и заполняем его по примеру

3.2 Транслятор NASM

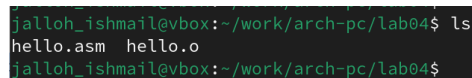
Преобразуем текст программы в объектный код (рис. fig. 3.5).



```
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ nasm -f elf hello.asm
jalloh_ishmail@vbox:~/work/arch-pc/lab04$
```

Рис. 3.5: Используем команду nasm

Проверяем создан ли объектный файл с помощью команды ls (рис. fig. 3.6).

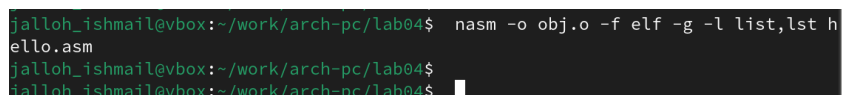


```
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ ls
hello.asm hello.o
jalloh_ishmail@vbox:~/work/arch-pc/lab04$
```

Рис. 3.6: Проверяем работу команды

3.3 Расширенный синтаксис командной строки NASM

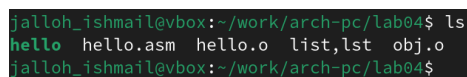
Компилируем исходный файл (рис. fig. 3.7).



```
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
jalloh_ishmail@vbox:~/work/arch-pc/lab04$
```

Рис. 3.7: Преобразуем файл hello.asm в obj.o

Проверяем, как сработала команда (рис. fig. 3.8).



```
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ ls
hello.asm hello.o list.lst obj.o
jalloh_ishmail@vbox:~/work/arch-pc/lab04$
```

Рис. 3.8: Проверяем создание файла командой ls

3.4 Компоновщик LD

Передаем объектный файл на обработку компоновщику (рис. fig. 3.9).

```
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
jalloh_ishmail@vbox:~/work/arch-pc/lab04$
```

Рис. 3.9: Используем команду ld

Проверяем созданся ли исполняемый файл hello (рис. fig. 3.10).

```
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ ls
hello.asm hello.o list.lst obj.o
jalloh_ishmail@vbox:~/work/arch-pc/lab04$
jalloh_ishmail@vbox:~/work/arch-pc/lab04$
```

Рис. 3.10: Используем команду ls

Передаем объектный файл на обработку компоновщику (рис. fig. 3.11).

```
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
jalloh_ishmail@vbox:~/work/arch-pc/lab04$
```

Рис. 3.11: Используем команду ld, создавая файл main

Проверяем созданся ли исполняемый файл hello (рис. fig. 3.12).

```
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o list.lst main obj.o
jalloh_ishmail@vbox:~/work/arch-pc/lab04$
```

Рис. 3.12: Используем команду ls

3.5 Запуск исполняемого файла

Запускаем на выполнение созданный исполняемый файл (рис. fig. 3.13).

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

Рис. 3.13: Используем команду ./hello

3.6 Задание для самостоятельной работы

Создаем копию файла hello.asm (рис. fig. 3.14).

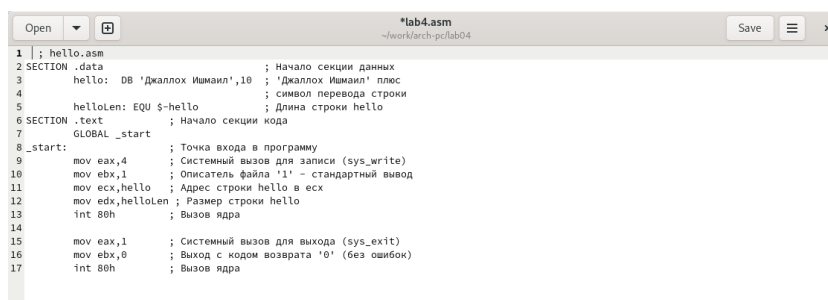
```
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
jalloh_ishmail@vbox:~/work/arch-pc/lab04$
```

Рис. 3.14: Используем команду cp

Открываем файл и редактируем его (рис. fig. 3.15).

```
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ gedit lab4.asm
```

Рис. 3.15: Открываем файл в текстовом редакторе



```
*lab4.asm
~/.work/arch-pc/lab04
1 | ; hello.asm
2 SECTION .data
3     hello: DB 'Джаллох Ишмаил',10 ; 'Джаллох Ишмаил' плюс
4                                     ; символ перевода строки
5     hellolen: EQU $-hello ; Длина строки hello
6 SECTION .text
7     GLOBAL _start ; Начало секции кода
8 _start:
9     mov eax,4 ; Точка входа в программу
10    mov ebx,1 ; Системный вызов для записи (sys_write)
11    mov ecx,hello ; Описатель файла '1' - стандартный вывод
12    mov edx,hellolen ; Адрес строки hello в ecx
13    int 80h ; Размер строки hello
14    ; Вызов ядра
15    mov eax,1 ; Системный вызов для выхода (sys_exit)
16    mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
17    int 80h ; Вызов ядра
```

Рис. 3.16: Редактируем файл для своего имени и фамилии

Прописываем те же команды, что и с первой программой (рис. fig. 3.17).

```
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst lab4.asm
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o hello
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ ./hello
Джаллох Ишмаил
jalloh_ishmail@vbox:~/work/arch-pc/lab04$
```

Рис. 3.17: Прописываем команды для работы файла и запускаем программу

Копируем файлы в локальный репозиторий (рис. fig. 3.18).

```
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ cp hello.asm ~/work/study/2024-2025/"Архитектура Компьютера"/study_2024-2025_arh-pc/labs/lab04/
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ cp lab4.asm ~/work/study/2024-2025/"Архитектура Компьютера"/study_2024-2025_arh-pc/labs/lab04/
jalloh_ishmail@vbox:~/work/arch-pc/lab04$
jalloh_ishmail@vbox:~/work/arch-pc/lab04$
```

Рис. 3.18: Копируем файлы в каталог с ЛР4

Переходим в каталог лабораторных работ и загружаем файлы на Github (рис. fig. 3.19).

```
jalloh_ishmail@vbox:~/work/arch-pc/lab04$ cd ~/work/study/2024-2025/"Архитектура
Компьютера"/study_2024-2025_arh-pc
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-202
5_arh-pc$ git add .
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-202
5_arh-pc$ git commit -am 'feat(main): add files lad-4'
[master 44dbc32] feat(main): add files lad-4
2 files changed, 34 insertions(+)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-202
5_arh-pc$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1002 bytes | 501.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:Isho-Jah/study_2024-2025_arh-pc.git
74e3f79..44dbc32 master -> master
jalloh_ishmail@vbox:~/work/study/2024-2025/Архитектура Компьютера/study_2024-202
5_arh-pc$
```

Рис. 3.19: Загружаем файлы

4 Выводы

Мы познакомились с языком ассемблера NASM и создали две работающих программы.