

Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютера

Машков Илья Евгеньевич

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Программа Hello world!	6
3.2	Транслятор NASM	7
3.3	Расширенный синтаксис командной строки NASM	7
3.4	Компоновщик LD	8
3.5	Запуск исполняемого файла	8
3.6	Выполнение заданий для самостоятельной работы	8
4	Выводы	11
5	Список литературы	12

Список иллюстраций

3.1	Создание папки для работы и .asm файла в ней.	6
3.2	Программа “Hello world!”.	7
3.3	Компиляция программы в файл hello.o.	7
3.4	Компиляция программы в файлы obj.o и list.lst.	8
3.5	Получение исполняемой программы и объектного файла.	8
3.6	Запуск программы.	8
3.7	Создание нового .asm файла для работы.	9
3.8	Работа с файлом.	9
3.9	Трансляция в объектный файл.	9
3.10	Компоновка и запуск программы.	9
3.11	Перемещение .asm файлов в локальный репозиторий.	10
3.12	Загрузка файлов на Github.	10

1 Цель работы

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

2 Задание

1. Программа Hello world!;
2. Транслятор NASM;
3. Расширенный синтаксис командной строки NASM;
4. Компоновщик LD;
5. Запуск исполняемого файла;
6. Выполнение заданий для самостоятельной работы.

3 Выполнение лабораторной работы

3.1 Программа Hello world!

Создаю папку lab4 в директории arch-pc и создаю файл hello.asm с помощью команды “touch” (рис. [3.1]).

```
lenashkov@lenashkov: ~/work/study/2023-2024/Архитектура компьютера/arch-pc$ mkdir lab04
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ ls
CHANGELOG.md  config  COURSE  lab04  labs  LICENSE  Makefile  prepare  presentation  README.en.md  README.git-flow.md  README.md  template
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ cd lab04
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab04$ touch hello.asm
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab04$ ls
hello.asm
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab04$
```

Рис. 3.1: Создание папки для работы и .asm файла в ней.

Далее открываю файл, копирую в него пример программы из лабораторной работы на ТУИСе и внимательно изучаю её структуру (Рис. [3.2]).

```

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     mov eax,1 ; Системный вызов для выхода (sys_exit)
17     mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
18     int 80h ; Вызов ядра

```

Рис. 3.2: Программа “Hello world!”.

3.2 Транслятор NASM

Чтобы скомпилировать текст из программы “Hello world!” в файл hello.o, использую команду “nasm -f elf hello.asm” и проверяю правильность проделанных действий с помощью команды “ls” (Рис. [3.3]).

```

lenashkov@lenashkov: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab04$ nasm -f elf hello.asm
lenashkov@lenashkov: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab04$ ls
hello.asm  hello.o

```

Рис. 3.3: Компиляция программы в файл hello.o.

3.3 Расширенный синтаксис командной строки NASM

Чтобы скомпилировать исходный файл hello.asm в файл obj.o и в листинг list.lst, использую команду “nasm -o obj.o -f elf -g -l list.lst hello.asm” и проверяю правильность проделанных действий с помощью команды “ls” (Рис. [3.4]).

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

Рис. 3.4: Компиляция программы в файлы obj.o и list.lst.

3.4 Компоновщик LD

Далее мне нужно получить исполняемую программу и объектный файл, чтобы это сделать мне необходимо передать файлы hello.o и obj.o, для чего я буду использовать команды “ld -m elf_i386 hello.o -o hello” и “ld -m elf_i386 obj.o -o main” соответственно (ключ -o используется для объектных файлов) (Рис. [3.5]).

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

Рис. 3.5: Получение исполняемой программы и объектного файла.

3.5 Запуск исполняемого файла

Теперь я запускаю исполняемый файл с помощью команды “./hello”, если всё было сделано правильно, то программа выведет надпись “Hello world!” (Рис. [3.6]).

```
lenashkov@lenashkov: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab04$ ./hello
Hello world!
lenashkov@lenashkov: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab04$
```

Рис. 3.6: Запуск программы.

3.6 Выполнение заданий для самостоятельной работы

1. Находясь в этом же каталоге, создаю копию файла hello.asm с именем lab4.asm с помощью команды “cp” (Рис. [3.7]).


```
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab04$ cp hello.asm lab4.asm
```

Рис. 3.7: Создание нового .asm файла для работы.

2. Теперь открываю lab4.asm в текстовом редакторе, заменяю выводимую фразу на свои имя и фамилию, а также везде заменяю слово “hello”, т.к. имя файла теперь lab4 (Рис. [3.8]).

```
; lab4.asm
SECTION .data ; Начало секции данных
lab4: DB 'Mashkov Ilya',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 lab4
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. 3.8: Работа с файлом.

3. Затем транслирую текст файла в объектный файл (Рис. [3.9]).

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

Рис. 3.9: Трансляция в объектный файл.

Далее компоную и запускаю программу (Рис. [3.10]).

```
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab04$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab04$ ./lab4
Mashkov Ilya
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab04$
```

Рис. 3.10: Компоновка и запуск программы.

4. Теперь копирую файлы “hello.asm” и “lab4.asm” в папку labs/lab04 (Рис. [3.12]).

```
lenashkov@lenashkov: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ cp hello.asm lab4.asm ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ cd
lenashkov@lenashkov: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ cd
lenashkov@lenashkov: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ cd
lenashkov@lenashkov: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello.asm  lab4.asm  presentation_report
lenashkov@lenashkov: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 3.11: Перемещение .asm файлов в локальный репозиторий.

После чего отправляю файлы на сервер (Рис. [3.12]).

```
lenashkov@lenashkov: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ git add .
lenashkov@lenashkov: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ git commit -am 'Add .asm files'
[master 5ce922e] Add .asm files
2 files changed, 35 insertions(+)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
lenashkov@lenashkov: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ git push
Перечисление объектов: 100% (9/9), 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), 1014 байтов | 1014.00 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:757eVe7N7/study_2023-2024_arh-pc.git
eb20473..5ce922e master -> master
```

Рис. 3.12: Загрузка файлов на Github.

4 Выводы

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

5 Список литературы

Архитектура ЭВМ