

Отчёт по лабораторной работе 4

Архитектура компьютера

Довлетов Довлет

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Программа «Hello, World!»	6
2.2	Использование транслятора NASM	7
2.3	Компоновка с помощью LD	8
2.4	Выполнение заданий для самостоятельной работы	9
3	Выводы	11

Список иллюстраций

2.1	Создание каталога и файла	6
2.2	Код программы hello.asm	7
2.3	Трансляция hello.asm в объектный файл	7
2.4	Трансляция hello.asm с дополнительными опциями	8
2.5	Линковка и создание исполняемого файла	8
2.6	Линковка для создания исполняемого файла main	8
2.7	Запуск исполняемых файлов	9
2.8	Измененный код программы в файле lab4.asm	10
2.9	Запуск измененной программы lab4.asm	10

Список таблиц

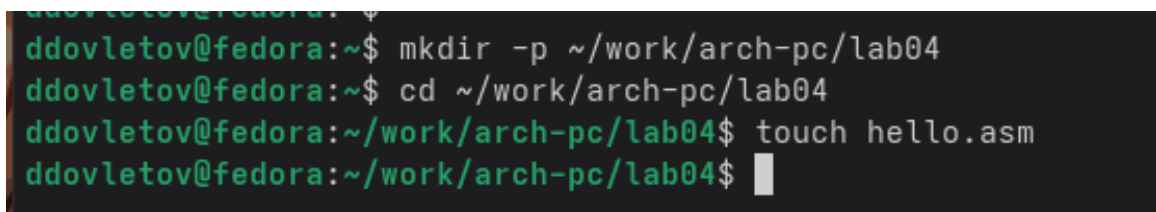
1 Цель работы

Целью работы является освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

2 Выполнение лабораторной работы

2.1 Программа «Hello, World!»

В первую очередь создаю каталог lab04 с помощью команды `mkdir`, после чего перехожу в него с помощью `cd` и создаю файл с именем `hello.asm`. (рис. 2.1)



```
ddovletov@fedora:~$ mkdir -p ~/work/arch-pc/lab04
ddovletov@fedora:~$ cd ~/work/arch-pc/lab04
ddovletov@fedora:~/work/arch-pc/lab04$ touch hello.asm
ddovletov@fedora:~/work/arch-pc/lab04$
```

Рисунок 2.1: Создание каталога и файла

Затем открываю созданный файл и реализую код программы согласно заданию. (рис. 2.2)



```
hello.asm
~/work/arch-pc/lab04

SECTION .data
    hello:      db "Hello, world!",0xa
               helloLen: equ $ - hello

SECTION .text
    global _start

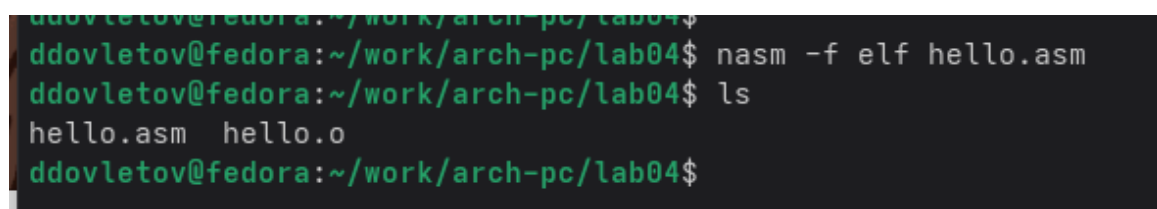
_start:
    mov eax, 4
    mov ebx, 1
    mov ecx, hello
    mov edx, helloLen
    int 0x80

    mov eax, 1
    mov ebx, 0
    int 0x80
```

Рисунок 2.2: Код программы hello.asm

2.2 Использование транслятора NASM

Для трансляции файла использую команду `nasm`, в результате чего получаю объектный файл `hello.o`. (рис. 2.3)



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

Рисунок 2.3: Трансляция hello.asm в объектный файл

Далее провожу трансляцию с дополнительными параметрами, что приводит к созданию файла листинга `list.lst`, объектного файла `obj.o`, а также к добавлению отладочной информации в программу. (рис. 2.4)

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

Рисунок 2.4: Трансляция `hello.asm` с дополнительными опциями

2.3 Компоновка с помощью LD

Для компоновки использую команду `ld`, в результате чего получается исполняемый файл. (рис. 2.5)

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

Рисунок 2.5: Линковка и создание исполняемого файла

Повторяю линковку для объектного файла `obj.o`, в результате чего создается исполняемый файл с именем `main`. (рис. 2.6)

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

Рисунок 2.6: Линковка для создания исполняемого файла `main`

Запускаю созданные исполняемые файлы для проверки их работы. (рис. 2.7)

```
ddovletov@fedora:~/work/arch-pc/lab04$  
ddovletov@fedora:~/work/arch-pc/lab04$ ./hello  
Hello, world!  
ddovletov@fedora:~/work/arch-pc/lab04$ ./main  
Hello, world!  
ddovletov@fedora:~/work/arch-pc/lab04$ █
```

Рисунок 2.7: Запуск исполняемых файлов

2.4 Выполнение заданий для самостоятельной работы

Копирую исходный код программы в новый файл.

Изменяю текст «Hello, World!» на свое имя (рис. 2.8) и запускаю измененную программу. (рис. 2.9)

```

SECTION .data
    hello:      db "Довлетов Довлет",0xa
                helloLen: equ $ - hello

SECTION .text
    global _start

_start:
    mov eax, 4
    mov ebx, 1
    mov ecx, hello
    mov edx, helloLen
    int 0x80

    mov eax, 1
    mov ebx, 0
    int 0x80

```

Рисунок 2.8: Измененный код программы в файле lab4.asm

```

ddovletov@fedora:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
ddovletov@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
ddovletov@fedora:~/work/arch-pc/lab04$ ./lab4
Довлетов Довлет
ddovletov@fedora:~/work/arch-pc/lab04$

```

Рисунок 2.9: Запуск измененной программы lab4.asm

3 Выводы

При выполнении данной лабораторной работы я освоил процесс компиляции и сборки программ, написанных на ассемблере `nasm`.