

Отчёта по лабораторной работе №4

Архитектура компьютера

Еремина Оксана Андреевна НКАбд-02-23

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
	Список литературы	11

Список таблиц

Список иллюстраций

4.1	рис.1	8
4.2	рис.2	8
4.3	рис.3	9
4.4	рис.4	9
4.5	рис.5	9
4.6	рис.6	9
4.7	рис.7	10
4.8	рис.8	10

1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM

2 Задание

4.3.1. Программа Hello world! 4.3.2. Транслятор NASM 4.3.3. Расширенный синтаксис командной строки NASM 4.4. компоновщик LD 4.4.1. Запуск исполняемого файла 4.5. Задание для самостоятельной работы

3 Теоретическое введение

Основными функциональными элементами любой электронно-вычислительной машины (ЭВМ) являются центральный процессор, память и периферийные устройства.

Язык ассемблера (assembly language, сокращённо asm) — машинно-ориентированный язык низкого уровня. Можно считать, что он больше любых других языков приближен к архитектуре ЭВМ и её аппаратным возможностям, что позволяет получить к ним более полный доступ, нежели в языках высокого уровня, таких как C/C++, Perl, Python и пр.

4 Выполнение лабораторной работы

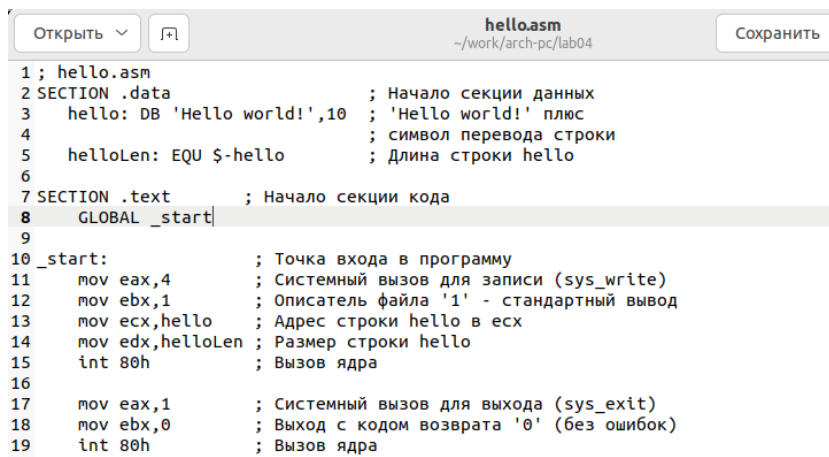
4.3.1. Программа Hello world!

Создаю каталог для работы на языке ассемблера NASM, перехожу в него и создаю текстовый файл, открываю текстовый файл с помощью текстового редактора (рис.1)

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

Рис. 4.1: рис.1

Ввожу текст для вывода приветственного сообщения. (рис.2)



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

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

Рис. 4.2: рис.2

4.3.2. Транслятор NASM

С помощью команды превращаю текст программы в объектный код, и проверяю наличие, появившегося файла с помощью команды ls

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

Рис. 4.3: рис.3

4.3.3. Расширенный синтаксис командной строки NASM

С помощью команды компелирую файл hello.asm в obj.o и проверяю, что файлы создались (рис.4)

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

Рис. 4.4: рис.4

4.4. Компоновщик LD

Передаю объектный файл на обработку компоновщику и проверяю, что файл был создан (рис.5)

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

Рис. 4.5: рис.5

4.4.1. Запуск исполняемого файла

Запускаю файл и смотрю, что он выведет в терминале (рис.6)

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

Рис. 4.6: рис.6

4.5. Задание для самостоятельной работы

С помощью cp создаю копию файла hello.asm с именем lab4.asm(рис.7)

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

Рис. 4.7: рис.7

Открываю файл в текстовом редакторе и вношу изменения в программу так, чтобы она выводила мои имя и фамилию (рис.8)

```
1 ; lab4.asm
2 SECTION .data          ; Начало секции данных
3     hello: DB 'Eremina Oksana',10 ; 'Hello world!' плюс
4                                     ; символ перевода строки
5     helloLen: EQU $-lab4      ; 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,lab4        ; Адрес строки hello в ecx
14     mov edx,lab4Len      ; Размер строки hello
15     int 80h            ; Вызов ядра
16
17     mov eax,1           ; Системный вызов для выхода (sys_exit)
18     mov ebx,0           ; Выход с кодом возврата '0' (без ошибок)
19     int 80h            ; Вызов ядра
```

Рис. 4.8: рис.8

Компилирую текст программы в объектный файл, проверяю, что файл создан. Передаю объектный файл на обработку компоновщику и запускаю программу (рис.9)

```
oaeremina@oaeremina:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
oaeremina@oaeremina:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o
oaeremina@oaeremina:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
oaeremina@oaeremina:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o
oaeremina@oaeremina:~/work/arch-pc/lab04$ ./lab4
Eremina Oksana
```

Выводы

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

Список литературы

Архитектура ЭВМ