

Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютера

Наговицын Арсений Владимирович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Программа Hello world!	9
4.2	Работа с транслятором NASM.	10
4.3	Работа с расширенным синтаксисом командной строки NASM. . .	10
4.4	Работа с компоновщиком LD.	11
4.5	Запуск исполняемого файла.	11
4.6	Задание для самостоятельной работы.	11
5	Выводы	15

Список иллюстраций

4.1	Создание каталога	9
4.2	Создание файла	9
4.3	Заполнение файла	10
4.4	Компиляция текста программы	10
4.5	Компиляция текста программы	11
4.6	Компиляция текста программы	11
4.7	Запуск исполняемого файла	11
4.8	Создание копии файла	12
4.9	Изменение программы	12
4.10	Компиляция текста программы	12
4.11	Компиляция текста программы	13
4.12	Компиляция текста программы	13
4.13	Запуск исполняемого файла	13
4.14	Копирование файлов	13
4.15	Файлы в каталоге	14

Список таблиц

1 Цель работы

Цель данной работы: освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

2 Задание

1. Программа Hello world!
2. Работа с транслятором NASM.
3. Работа с расширенным синтаксисом командной строки NASM.
4. Работа с компоновщиком LD.
5. Запуск исполняемого файла.
6. Задание для самостоятельной работы.

3 Теоретическое введение

Основными функциональными элементами любой ЭВМ являются центральный процессор, память и периферийные устройства. Эти устройства взаимодействуют друг с другом через общую шину, которая соединяет их. В современных компьютерах проводники шины представлены в виде электропроводящих дорожек на материнской плате. Центральный процессор выполняет обработку информации и координирует работу всех узлов компьютера. Он состоит из нескольких устройств, включая арифметико-логическое устройство (АЛУ), которое выполняет логические и арифметические операции над данными, и устройство управления (УУ), которое обеспечивает управление и контроль всех устройств компьютера. Также в состав процессора входят регистры - сверхбыстрая оперативная память небольшого объема, используемая для хранения промежуточных результатов выполнения инструкций.

Для написания программ на ассемблере необходимо знать, какие регистры процессора существуют и как их использовать. Примерами основных регистров общего назначения в архитектуре x86 являются RAX, RCX, RDX, RBX, RSI, RDI (64-битные), EAX, ECX, EDX, EBX, ESI, EDI (32-битные), AX, CX, DX, BX, SI, DI (16-битные) и AH, AL, CH, CL, DH, DL, BH, BL (8-битные). Оперативное запоминающее устройство (ОЗУ) является важным элементом ЭВМ. Оно предназначено для хранения программ и данных, с которыми процессор работает в текущий момент. ОЗУ состоит из ячеек памяти с уникальными номерами, которые служат адресами для хранящихся данных.

Периферийные устройства включают устройства внешней памяти, предна-

значенные для хранения больших объемов данных, и устройства ввода-вывода, которые обеспечивают взаимодействие с внешней средой.

Вычислительный процесс ЭВМ основан на программном управлении, где задача решается последовательностью действий, записанных в программе. Коды команд представляют собой двоичные комбинации, где операционная часть содержит код выполняемой команды, а адресная часть содержит данные или адреса данных, необходимых для выполнения операции. При выполнении команды процессор выполняет командный цикл, который включает формирование адреса команды, считывание кода команды, выполнение команды и переход к следующей команде.

Язык ассемблера (assembly language) является машинно-ориентированным языком низкого уровня. NASM (Netwide Assembler) - это проект ассемблера, который поддерживает Intel-синтаксис и инструкции x86-64. NASM доступен для различных операционных систем и позволяет получать объектные файлы для этих систем.

4 Выполнение лабораторной работы

4.1 Программа Hello world!

Перехожу в каталог, с помощью утилиты `cd`, и создаю папку `lab04` (рис. 4.1).

```
(avnagovicihn@avnagovicihn)-[~]  
$ cd RUDN/study/2023-2024/Архитектура\ коьютера/arch-pc  
(avnagovicihn@avnagovicihn)-[~/../study/2023-2024/Архитектура коьютера/arch-pc]  
$ mkdir lab04  
(avnagovicihn@avnagovicihn)-[~/../study/2023-2024/Архитектура коьютера/arch-pc]  
$ ls  
lab04  labs
```

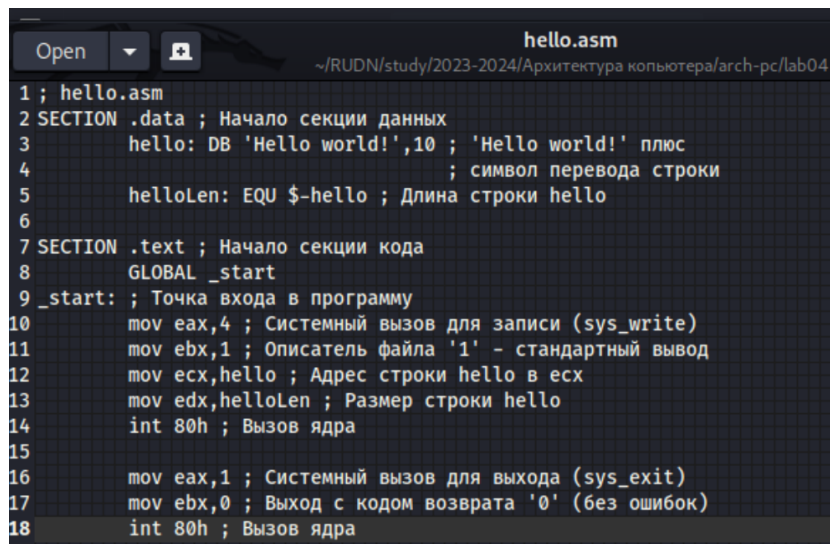
Рис. 4.1: Создание каталога

Создаю в текущем каталоге файл и открываю файл в текстовом редакторе (рис. 4.2).

```
(avnagovicihn@avnagovicihn)-[~/../2023-2024/Архитектура коьютера/arch-pc/lab04]  
$ touch hello.asm  
(avnagovicihn@avnagovicihn)-[~/../2023-2024/Архитектура коьютера/arch-pc/lab04]  
$ ls  
hello.asm  
(avnagovicihn@avnagovicihn)-[~/../2023-2024/Архитектура коьютера/arch-pc/lab04]  
$ gedit hello.asm
```

Рис. 4.2: Создание файла

Заполняю файл, вставляя в него программу для вывода “Hello world!” (рис. 4.3).



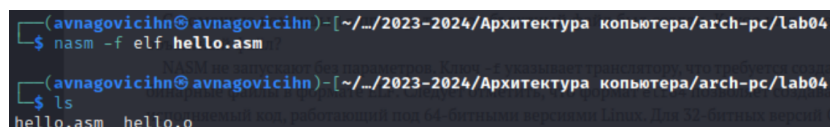
```
hello.asm
~/RUDN/study/2023-2024/Архитектура копыютера/arch-pc/lab04

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 _start: ; Точка входа в программу
10    mov eax,4 ; Системный вызов для записи (sys_write)
11    mov ebx,1 ; Описатель файла '1' - стандартный вывод
12    mov ecx,hello ; Адрес строки hello в ecx
13    mov edx,helloLen ; Размер строки hello
14    int 80h ; Вызов ядра
15
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. 4.3: Заполнение файла

4.2 Работа с транслятором NASM.

Превращаю текст программы в объектный код с помощью транслятора NASM (рис. 4.4).



```
(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура копыютера/arch-pc/lab04]
$ nasm -f elf hello.asm

(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура копыютера/arch-pc/lab04]
$ ls
hello.asm  hello.o
```

Рис. 4.4: Компиляция текста программы

4.3 Работа с расширенным синтаксисом командной строки NASM.

Ввожу команду, которая скомпилирует файл Hello.asm в файл obj.o (рис. 4.5).

```

(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура компьютера/arch-pc/lab04]
$ nasm -o obj.o -f elf -g -l list.lst hello.asm

(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура компьютера/arch-pc/lab04]
$ ls
hello.asm  hello.o  list.lst  obj.o

```

Рис. 4.5: Компиляция текста программы

4.4 Работа с компоновщиком LD.

Передаю объектный файл hello.o на обработку компоновщику ld (рис. 4.6).

```

(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура компьютера/arch-pc/lab04]
$ ld -m elf_i386 hello.o -o hello

(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура компьютера/arch-pc/lab04]
$ ls
hello  hello.asm  hello.o  list.lst  obj.o

```

Рис. 4.6: Компиляция текста программы

4.5 Запуск исполняемого файла.

Запускаю на выполнение созданный исполняемый файл hello (рис. 4.7).

```

(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура компьютера/arch-pc/lab04]
$ ./hello
Hello world!

```

Рис. 4.7: Запуск исполняемого файла

4.6 Задание для самостоятельной работы.

Создаю в текущем каталоге копию файла hello.asm с именем lab4.asm и запускаю в текстовом редакторе (рис. 4.8).

```

(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура компьютера/arch-pc/lab04]
$ cp hello.asm lab4.asm

(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура компьютера/arch-pc/lab04]
$ ls
hello hello.asm hello.o lab4.asm list.lst obj.o

(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура компьютера/arch-pc/lab04]
$ gedit lab4.asm

```

Рис. 4.8: Создание копии файла

Вношу изменения в программу, чтобы она выводила мое имя и фамилию (рис. 4.9).

```

lab4.asm
~/RUDN/study/2023-2024/Архитектура компьютера/arch-pc/lab04

1 ; lab4.asm
2 SECTION .data ; Начало секции данных
3     lab4: DB 'Nagovitsyn Arseniy ',10
4
5     lab4Len: EQU $-lab4
6
7 SECTION .text ; Начало секции кода
8     GLOBAL _start
9 _start: ; Точка входа в программу
10    mov eax,4 ; Системный вызов для записи (sys_write)
11    mov ebx,1 ; Описатель файла '1' - стандартный вывод
12    mov ecx,lab4 ; Адрес строки hello в ecx
13    mov edx,lab4Len ; Размер строки hello
14    int 80h ; Вызов ядра
15
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. 4.9: Изменение программы

Компилирую текст программы в объектный код с помощью транслятора NASM (рис. 4.10).

```

(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура компьютера/arch-pc/lab04]
$ nasm -f elf lab4.asm

(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура компьютера/arch-pc/lab04]
$ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst obj.o

```

Рис. 4.10: Компиляция текста программы

Ввожу команду, которая скомпилирует файл lab4.asm в файл lab4.o (рис. 4.11).

```

(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура компьютера/arch-pc/lab04]
$ nasm -o lab4.o -f elf -g -l list.lst lab4.asm

(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура компьютера/arch-pc/lab04]
$ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst obj.o

```

Рис. 4.11: Компиляция текста программы

Передаю объектный файл lab4.o на обработку компоновщику LD (рис. 4.12).

```

(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура компьютера/arch-pc/lab04]
$ ld -m elf_i386 lab4.o -o lab4

(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура компьютера/arch-pc/lab04]
$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst obj.o

```

Рис. 4.12: Компиляция текста программы

Запускаю на выполнение созданный исполняемый файл lab4 (рис. 4.13).

```

(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура компьютера/arch-pc/lab04]
$ ./lab4
Nagovitsyn Arseniy

```

Рис. 4.13: Запуск исполняемого файла

Копирую файлы hello.asm и lab4.asm в каталог (рис. 4.14).

```

(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура компьютера/arch-pc/lab04]
$ cp hello.asm lab4.asm ~/RUDN/study/2023-2024/Архитектура\ компьютера/arch-pc/labs/lab04

(avnagovicihn@avnagovicihn)-[~/2023-2024/Архитектура компьютера/arch-pc/lab04]

```

Рис. 4.14: Копирование файлов

Проверяю нахождение файлов в каталоге (рис. 4.15).

```
(avnagovicihn@ avnagovicihn)-[~/../2023-2024/Архитектура копьютера/arch-  
c/lab04]  
$ cd ..  
  
(avnagovicihn@ avnagovicihn)-[~/../study/2023-2024/Архитектура копьютера/  
arch-pc]  
$ cd labs/lab04  
  
(avnagovicihn@ avnagovicihn)-[~/../Архитектура копьютера/arch-pc/labs/lab  
04]  
$ ls  
hello.asm lab4.asm presentation report
```

Рис. 4.15: Файлы в каталоге

5 Выводы

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