

Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютера

Гаврилейко Алина Александровна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	10
4.1	Транслятор NASM	11
4.2	Расширенный синтаксис командной строки NASM	11
4.3	Компоновщик LD	12
4.4	Запуск исполняемого файла	13
4.5	Задания для самостоятельной работы	13
5	Выводы	15
6	Список литературы	16

Список иллюстраций

4.1	Создание рабочей директории	10
4.2	Создание .asm файла	10
4.3	Редактирование файла	11
4.4	Компиляция программы	11
4.5	Возможности синтаксиса NASM	12
4.6	Отправка файла компоновщику	12
4.7	Создание исполняемого файла	12
4.8	Запуск программы	13
4.9	Создание копии	13
4.10	Редактирование копии	13
4.11	Проверка работоспособности скомпонованной программы	14
4.12	Отправка файлов в локальный репозиторий	14
4.13	Загрузка изменений	14

Список таблиц

1 Цель работы

Цель данной лабораторной работы - освоить процедуры компиляции и сборки программ, написанных на ассемблере NASM.

2 Задание

1. Создание программы 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. Выполнение заданий для самостоятельной работы.

3 Теоретическое введение

Основными функциональными элементами любой ЭВМ являются центральный процессор, память и периферийные устройства. Взаимодействие этих устройств осуществляется через общую шину, к которой они подключены. Физически шина представляет собой большое количество проводников, соединяющих устройства друг с другом. В современных компьютерах проводники выполнены в виде электропроводящих дорожек на материнской плате. Основной задачей процессора является обработка информации, а также организация координации всех узлов компьютера. В состав центрального процессора входят следующие устройства: - арифметико-логическое устройство (АЛУ) — выполняет логические и арифметические действия, необходимые для обработки информации, хранящейся в памяти; - устройство управления (УУ) — обеспечивает управление и контроль всех устройств компьютера; - регистры — сверхбыстрая оперативная память небольшого объёма, входящая в состав процессора, для временного хранения промежуточных результатов выполнения инструкций; регистры процессора делятся на два типа: регистры общего назначения и специальные регистры. Для того, чтобы писать программы на ассемблере, необходимо знать, какие регистры процессора существуют и как их можно использовать. Большинство команд в программах написанных на ассемблере используют регистры в качестве операндов. Практически все команды представляют собой преобразование данных хранящихся в регистрах процессора, это например пересылка данных между регистрами или между регистрами и памятью, преобразование (арифметические или логические

операции) данных хранящихся в регистрах. Доступ к регистрам осуществляется не по адресам, как к основной памяти, а по именам. Каждый регистр процессора архитектуры x86 имеет свое название, состоящее из 2 или 3 букв латинского алфавита. В качестве примера приведем названия основных регистров общего назначения (именно эти регистры чаще всего используются при написании программ): - RAX, RCX, RDX, RBX, RSI, RDI — 64-битные - EAX, ECX, EDX, EBX, ESI, EDI — 32-битные - AX, CX, DX, BX, SI, DI — 16-битные - AH, AL, CH, CL, DH, DL, BH, BL — 8-битные

Другим важным узлом ЭВМ является оперативное запоминающее устройство (ОЗУ). ОЗУ — это быстродействующее энергозависимое запоминающее устройство, которое напрямую взаимодействует с узлами процессора, предназначенное для хранения программ и данных, с которыми процессор непосредственно работает в текущий момент. ОЗУ состоит из одинаковых пронумерованных ячеек памяти. Номер ячейки памяти — это адрес хранящихся в ней данных. Периферийные устройства в составе ЭВМ: - устройства внешней памяти, которые предназначены для долговременного хранения больших объёмов данных. - устройства ввода-вывода, которые обеспечивают взаимодействие ЦП с внешней средой.

В основе вычислительного процесса ЭВМ лежит принцип программного управления. Это означает, что компьютер решает поставленную задачу как последовательность действий, записанных в виде программы.

Коды команд представляют собой многоразрядные двоичные комбинации из 0 и 1. В коде машинной команды можно выделить две части: операционную и адресную. В операционной части хранится код команды, которую необходимо выполнить. В адресной части хранятся данные или адреса данных, которые участвуют в выполнении данной операции. При выполнении каждой команды процессор выполняет определённую последовательность стандартных действий, которая называется командным циклом процессора. Он заключается в следующем: 1. формирование адреса в памяти очередной команды; 2. считывание кода команды из памяти и её дешифрация; 3. выполнение команды; 4. переход к

следующей команде.

Язык ассемблера (assembly language, сокращённо asm) — машинно-ориентированный язык низкого уровня. NASM — это открытый проект ассемблера, версии которого доступны под различные операционные системы и который позволяет получать объектные файлы для этих систем. В NASM используется Intel-синтаксис и поддерживаются инструкции x86-64.

4 Выполнение лабораторной работы

##Програма Hello world!

В домашней директории создаю каталог, в котором буду хранить файлы для текущей лабораторной работы. (рис. 4.1)

```
alina@gavrileykoalina: $ mkdir -p ~/work/arch-pc/lab04  
alina@gavrileykoalina: $ cd ~/work/arch-pc/lab04
```

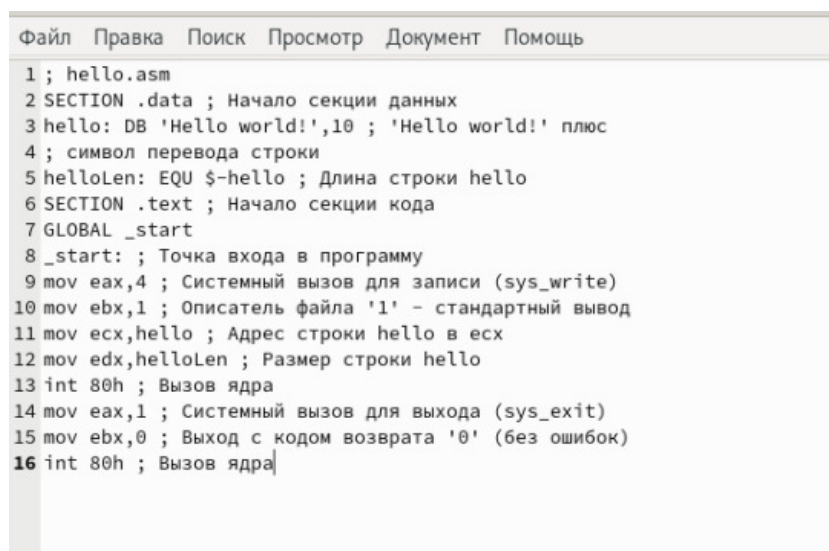
Рис. 4.1: Создание рабочей директории

Создаю в нем файл hello.asm, в котором буду писать программу на языке ассемблера. (рис. 4.2)

```
alina@gavrileykoalina: $ mkdir -p ~/work/arch-pc/lab04  
alina@gavrileykoalina: $ cd ~/work/arch-pc/lab04  
alina@gavrileykoalina: ~/work/arch-pc/lab04$ touch hello.asm  
alina@gavrileykoalina: ~/work/arch-pc/lab04$ mousepad hello.asm
```

Рис. 4.2: Создание .asm файла

С помощью редактора пишу программу в созданном файле. (рис. 4.3)

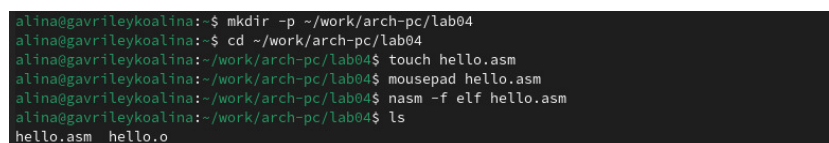


```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4 ; символ перевода строки
5 helloLen: EQU $-hello ; Длина строки hello
6 SECTION .text ; Начало секции кода
7 GLOBAL _start
8 _start: ; Точка входа в программу
9 mov eax,4 ; Системный вызов для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

Рис. 4.3: Редактирование файла

4.1 Транслятор NASM

Компилирую с помощью NASM свою программу. (рис. 4.4)



```
alina@gavrileykoalina:~$ mkdir -p ~/work/arch-pc/lab04
alina@gavrileykoalina:~$ cd ~/work/arch-pc/lab04
alina@gavrileykoalina:~/work/arch-pc/lab04$ touch hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ mousepad hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ nasm -f elf hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
```

Рис. 4.4: Компиляция программы

4.2 Расширенный синтаксис командной строки NASM

Выполняя команду, указанную на (рис. 4.5), она скомпилировала исходный файл hello.asm в obj.o, расширение .o говорит о том, что файл - объектный, помимо него флаги -g -l подготовят файл отладки и листинга соответственно.

```

alina@gavrileykoalina:~$ mkdir -p ~/work/arch-pc/lab04
alina@gavrileykoalina:~$ cd ~/work/arch-pc/lab04
alina@gavrileykoalina:~/work/arch-pc/lab04$ touch hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ mousepad hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ nasm -f elf hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
alina@gavrileykoalina:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o

```

Рис. 4.5: Возможности синтаксиса NASM

4.3 Компоновщик LD

Затем мне необходимо передать объектный файл компоновщику, делаю это с помощью команды `ld`. (рис. 4.6)

```

alina@gavrileykoalina:~$ mkdir -p ~/work/arch-pc/lab04
alina@gavrileykoalina:~$ cd ~/work/arch-pc/lab04
alina@gavrileykoalina:~/work/arch-pc/lab04$ touch hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ mousepad hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ nasm -f elf hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
alina@gavrileykoalina:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
alina@gavrileykoalina:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  obj.o

```

Рис. 4.6: Отправка файла компоновщику

Выполняю следующую команду ..., результатом исполнения команды будет созданный файл `main`, скомпонованный из объектного файла `obj.o`. (рис. 4.7)

```

alina@gavrileykoalina:~$ mkdir -p ~/work/arch-pc/lab04
alina@gavrileykoalina:~$ cd ~/work/arch-pc/lab04
alina@gavrileykoalina:~/work/arch-pc/lab04$ touch hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ mousepad hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ nasm -f elf hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
alina@gavrileykoalina:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
alina@gavrileykoalina:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  obj.o
alina@gavrileykoalina:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
alina@gavrileykoalina:~/work/arch-pc/lab04$ ./hello
Hello world!
alina@gavrileykoalina:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o

```

Рис. 4.7: Создание исполняемого файла

4.4 Запуск исполняемого файла

Запускаю исполняемый файл из текущего каталога. (рис. 4.8)

```
alina@gavrileykoalina:~$ mkdir -p ~/work/arch-pc/lab04
alina@gavrileykoalina:~$ cd ~/work/arch-pc/lab04
alina@gavrileykoalina:~/work/arch-pc/lab04$ touch hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ mousepad hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ nasm -f elf hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
alina@gavrileykoalina:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
alina@gavrileykoalina:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  obj.o
alina@gavrileykoalina:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
alina@gavrileykoalina:~/work/arch-pc/lab04$ ./hello
Hello world!
```

Рис. 4.8: Запуск программы

4.5 Задания для самостоятельной работы

Создаю копию файла для последующей работы с ней. (рис. 4.9)

```
alina@gavrileykoalina:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
alina@gavrileykoalina:~/work/arch-pc/lab04$ mousepad lab4.asm
```

Рис. 4.9: Создание копии

Редактирую копию файла, заменив текст на свое имя и фамилию. (рис. 4.10)

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

Рис. 4.10: Редактирование копии

Транслирую копию файла в объектный файл, компоную и запускаю. (рис. 4.11)

```

alina@gavrileykoalina:~$ mkdir -p ~/work/arch-pc/lab04
alina@gavrileykoalina:~$ cd ~/work/arch-pc/lab04
alina@gavrileykoalina:~/work/arch-pc/lab04$ touch hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ mousepad hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ nasm -f elf hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
alina@gavrileykoalina:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
alina@gavrileykoalina:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  obj.o
alina@gavrileykoalina:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
alina@gavrileykoalina:~/work/arch-pc/lab04$ ./hello
Hello world!
alina@gavrileykoalina:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
alina@gavrileykoalina:~/work/arch-pc/lab04$ mousepad lab4.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
alina@gavrileykoalina:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
alina@gavrileykoalina:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4  lab4.asm  lab4.o  list.lst  main  obj.o
alina@gavrileykoalina:~/work/arch-pc/lab04$ ./lab4
Gavrileyko Alina

```

Рис. 4.11: Проверка работоспособности скомпонованной программы

Убедившись в корректности работы программы, копирую рабочие файлы в свой локальный репозиторий. (рис. 4.12)

```

alina@gavrileykoalina:~/work/arch-pc/lab04$ cp hello.asm lab4.asm ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab04/

```

Рис. 4.12: Отправка файлов в локальный репозиторий

Загрузка изменений на удаленный репозиторий на GitHub. (рис. 4.13)

```

alina@gavrileykoalina:~/work/arch-pc/lab04$ cd ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab04/
alina@gavrileykoalina:~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab04$ ls
hello.asm  lab4.asm  presentation  report
alina@gavrileykoalina:~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab04$ git add .
alina@gavrileykoalina:~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab04$ git commit -m "feat(main): add files for lab04"

```

Рис. 4.13: Загрузка изменений

5 Выводы

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

6 Список литературы

1. Курс на ТУИС
2. Лабораторная работа №4
3. Программирование на языке ассемблера NASM Столяров А. В.