## Лабораторная работа №4

Дисциплина: Архитектура компьютера

Шония Ника Гигловна

## Содержание

Цель работы	5
Теоретическое введение	6
Выполнение лабораторной работы	8
Список литературы	10

## Список таблиц

# Список иллюстраций

### Цель работы

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

#### Теоретическое введение

AH, AL, CH, CL, DH, DL, BH, BL — 8-битные

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 де устройство управления (УУ) — обеспечивает управление и контроль всех устройств корегистры — сверхбыстрая оперативная память небольшого объёма, входящая в состав гах, RCX, RDX, RBX, RSI, RDI — 64-битные EAX, ECX, EDX, EBX, ESI, EDI — 32-битные AX, CX, DX, BX, SI, DI — 16-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 бол 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) — машинноориентированный 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.

#### Выполнение лабораторной работы

1. Создание программы Hello world! Создаю каталог для работы с программа-

nikashoniya@nikashoniya-VirtualBox: ~/work/arch-pc/lab04 nikashoniya@nikashoniya-VirtualBox:~\$ mkdir -p ~/work/arch-pc/lab04 nikashoniya@nikashoniya-VirtualBox:~\$ cd ~/work/arch-pc/lab04 ми на языке ассемблера NASM.

Создаю текстовый файл с именем hello.asm и открытаю его с помощью тек-\$ nano hello.asm

стового редактора nano.

```
GNU nano 6.2
      hello.asm
                                       : Начало секции данных
                            'Hello world!',10 ; 'Hello world!' плюс
     символ перевода строки

elloLen: EQU $-hello ; Длина строки hello

ECTION .text ; Начало секции кода
start: ; Точка входа в программу
mov eax, 4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,hello ; Адрес строки hello в есх
mov edx,helloLen ; Размер строки hello
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
 int 80h ; Вызов ядра
```

Заполнению файл текстом.

2. Работа с транслятором NASM Для компиляции текста программы «Hello nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab04\$ nasm -f elf hello.asm nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab04\$ ls World» пишу команду. hello.asm hello.o

3. Работа с расширенным синтаксисом командной строки NASM Ввожу команду. Данная команда скомпилирует исходный файл hello.asm в obj.o

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

4. Работа с компоновщиком LD Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику.

```
ikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o
nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab04$ ls
ello hello.asm hello.o list.lst obj.o
                                                                            Выполняю коман-
```

ду.Исполняемый файл будет иметь имя main, т.к. после ключа - о было задано значение main. Объектный файл, из которого собран этот исполняемый

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

- 5. Запуск исполняемого файла Набираю в командной строке ./hello nikashoniya@nikashoniya-VirtualBox:~/work/arch-pc/lab04\$ ./hello Hello world!
- 6. Выполнение заданий для самостоятельной работы С помощью команды ср создайте копию файла hello.asm с именем lab4.asm nikashoniya-VirtualBox:-/work/arch-pc/laboas ср hello.asm lab4.asm С помощью текстового редактора вношу изменения в текст программы в файле lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с моим фа-

```
GNU nano 6.2 lab4.asm *
; hello.asm
SECTION .data; Начало секции данных
hello: DB 'Shoniya Nika',10; 'Shoniya Nika' плюс
; символ перевода строки
helloLen: EQU $-hello; Длина строки hello
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,hello; Адрес строки hello в есх
mov edx,helloLen; Размер строки hello
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

МИЛИЕЙ И ИМЕНЕМ.
```

Компилирую текст программы в объектный файл и проверяю его наличие.

ntkashontya@ntkashontya-VirtualBox:~/work/arch-pc/lab04\$ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o

Передаю объектный файл lab4.o на обработку компоновщику LD, чтобы получить исполняntkashontya@ntkashontya-VirtualBox:-/work/arch-pc/lab04\$ ls
ntkashontya@ntkashontya-VirtualBox:-/work/arch-pc/lab04\$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o

Запускаю исполняемый файл lab4, на экран действительно выводятся мои

Запускаю исполняемый файл lab4, на экран действительно выводятся мож ntkashontya@ntkashontya-VirtualBox:~/work/arch-pc/lab04\$ ./lab4 ммя и фамилия

Выводы

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

## Список литературы

Архитектура компьютера

Ссылка на мой репозиторий: https://github.com/NikaShoniya/study\_2023-2024\_arch-pc