**УСТАНОВКА ПО И РАБОТА С ЭМУЛЯТОРОМ i8086**

**I. Установка NASM** (свободный ассемблер архитектуры х86)

1. Набрать: **NASM**.
2. Выбрать первую ссылку.
3. Выбрать последнюю версию:

Latest version .

Stable **2.15.05** .

1. Выбрать операционную систему:

**Win 64/**

1. Скачать инсталлятор, выбрав:

**nasm – 2.15.05 – installer – x64.exe**

1. Запустить инсталлятор, кликнув по файлу в левом нижнем углу.

Появится окно начала инсталляции. Следовать указаниям. Close

1. Проверить путь установки NASM. Он должен соответствовать пути, указанному в части II (Добавление пути к NASM).

Желательный (но не обязательный) путь к NASM:

**C:\Program Files\NASM\**

**II. Добавление пути к NASM**

ПУСК

1. Кнопка .
2. В окне поиска набрать: «переменные».
3. В открывшемся меню выбрать: «Изменение системных переменных среды».

Переменные среды

1. В окне «Свойства системы» нажать .
2. В окне «Переменные среды» в подокне «Системные переменные»

Изменить

выбрать позицию: «Path» и нажать .

1. В открывшемся окне «Изменить переменную среды» добавить путь к NASM

Создать

(Нажать и ввести с клавиатуры:

ОК

**C:\Program Files\NASM\** ).

**III. Написание кода**

1 способ (простейший)

1. Открыть блокнот, встроенный в Windows.
2. Составить нужный код и сохранить его в нужной папке не как **txt**-файл,

а как файл с расширением «**.asm»** (например: «commands.asm»).

Отсутствует удобный интерфейс.

2 способ (нужна установка дополнительного ПО (Notepad++))

1. Зайти в поисковик и набрать: **notepad++**
2. Выбрать **Downloads** (Загрузки).
3. Выбрать версию для скачивания, например:

**Notepad++ 8.3.3 (Make Apps, not war)**

1. Выбрать инсталлятор:

**Installer | GPG Signature** (Установщик | Подпись GPG)

и скачать его.

1. В системной папке **Downloads** (Загрузки) открыть файл:

**npp.8.3.3.Installer.x64.exe**

Следовать установщику.

1. Запустить Notepad++:

ПУСК

→ Программы → Notepad++

1. Создать ассемблерный файл:

**File** (Файл) → **New** (Новый) → **Language** (Синтаксисы) → **A** → **Assembly**

(можно выбрать язык).

1. Сохранение ассемблерного файла:

File → Save As… (будет предложено ассемблерное расширение «.asm»).

**IV. Компиляция написанного кода**

1. Войти в папку с составленными ранее ассемблерными кодами.
2. В поисковике набрать:

«**cmd**».

Появится окно:

**C:\Windows\System32\cmd.exe**

1. В командной строке окна, после названия текущей папки,

оканчивающегося значком « > », добавить название ассемблера (nasm)

и через пробел название ассемблерного файла (Program2.asm), например:

**D:\ Эмулятор i8086 (1810)\Programs>nasm Program2.asm**

При правильном наборе повторится предыдущая строка (до значка « > » –

без набранной с клавиатуры вставки), а в папке с ассемблерным кодом

появится откомпилированный файл Program2. Его название не должно

содержать пробелов. При ошибках в программе будет выведено сообщение.

**V. Работа с эмулятором i8086**

Запустив эмулятор (**i8086gui.exe**), найти требуемую

откомпилированную программу, например, Program2 (не Program2.asm):

**File** → **Open** → подокно **Folders** →

→ выбрав в просмотровщике папку, в подокне **Files** выбрать файл

(в машинных кодах) → откроется искомый файл.

Работа эмулятора i8086 в **пошаговом** режиме

Вторая слева кнопка в пиктографическом меню ( – Trace into)

заходит внутрь подпрограмм и пошагово их выполняет.

Третья слева кнопка ( – Step Over) пошагово выполняет всё, что

не является подпрограммой, а подпрограммы выполняет автоматически

(для максимального пошагового прохода).

Так как в данном случае нет подпрограмм, можно пользоваться любой

из двух кнопок.

Для программ, выполняемых в автоматическом режиме, есть возможность

задать **Update Interval** (ползунок справа от кнопок) – то, с какой скоростью

будет выполняться программа (в автоматическом режиме должен быть ≠ 0).

Работа эмулятора i8086 в **автоматическом** режиме

Четвёртая слева кнопка в пиктографическом меню ( – Run Program)

запускает автоматическое выполнение программы. При этом ползунок

«Update Interval» не должен находиться в крайнем левом положении

(скорость выполнения программы должна быть ≠ 0).

Кроме того, последняя команда программы должна быть зациклена сама

на себя: **end: jmp end**.

При нехватке выводимой на экран памяти воспользоваться прокруткой

в правом верхнем углу:

**˅**

– уменьшение адреса

**˄**

– увеличение адреса