Лекция 5. DHOG68 1/10/2024.

Замечание. ОТВЕТЫ НА ВОПРОСЫ НАЧИНАТЬ (И ПРОДОЛЖАТЬ) ВСЕГДА СО СЛОВА «ЕСЛИ…», ОГРАНИЧИВАЯ КОНТЕКСТ!!!! Например, если бы был вопрос: «Что такое файл»? То нужно начинать и продолжать так: «Если речь идёт о «бытовухе», то файл – это такая двусторонняя целлофановая хрень формата чуть больше А4, в которую вкладывают листочки, чтобы они не помялись. Если речь идёт о вычислительной технике и об используемой файловой системе, то под файлом понимается ПОИМЕНОВАННАЯ ОБЛАСТЬ НА каком-либо ВНЕШНЕМ ФИЗИЧЕСКОМ УСТРОЙСТВЕ (чаще всего на жестком диске). Если речь идёт о языках записи алгоритмов, то под файлом СЛЕДУЕТ ПОНИМАТЬ СТРУКТУРИРОВАННЫЙ ТИП ДАННЫХ (НЕ ПЕРЕМЕННУЮ типа файл, НЕ ФАЙЛ НА ВНЕШНЕМ НОСИТЕЛЕ).

А сегодняшняя Лекция 5 называется «DHOG68». И, как всегда, достаём листочки, подписываем и по ходу лекции отвечаем на вопросы, а в конце Лекции сдаём эти листочки.

**На одной из прошлых Лекций было у нас Толкование 29, в котором шла речь о двух основных устройствах ЭВМ – ПРОЦЕССОРЕ и ОПЕРАТИВНОЙ ПАМЯТИМ. Так что ВНИМАТЕЛЬНО ШТУДИРУЕМ ПРОДОЛЖЕНИЕ ТОЛКОВАНИЯ 29. Нужно понимать, что почти все компьютеры, существующие в мире, являются ЭВМ фон-неймановского ТИПА. Принцип, лежащий в основе данного типа ЭВМ предполагает то, ЧТО ПРОГРАММЫ И ДАННЫЕ НАХОДЯТСЯ В ОПЕРАТИВНОЙ ПАМЯТИ, а процессор осуществляет ВЫБОР КОМАНД ИЗ ОПЕРАТИВНОЙ ПАМЯТИ ПО ИХ АДРЕСУ, РАСШИФРОВЫВАЕТ ИХ, ИСПОЛНЯЕТ, используя АЛУ И РЕЗУЛЬТАТ ПОМЕЩАЕТ ЛИБО В РЕГИСТРЫ АЛУ либо В ОПЕРАТИВНУЮ ПАМЯТЬ.**

**Из всего сказанного и исходя из принципа «Разделяй и ВЛАСТВУЙ» следует В ПЕРВУЮ ОЧЕРЕДЬ обратить особое внимание, разобраться с ОПЕРАТИВНОЙ ПАМЯТЬЮ. Основные разборки можно найти в Лекциях Зубовича (что такое бит, байт, слово, двойное слово, числа с фиксированной точкой, числа с плавающей точкой и т.д.) НО ОСНОВНЫМ ПОНЯТИЕМ ЯВЛЯЕТСЯ ПОНЯТИЕ АДРЕСА!!!! Где под АДРЕСОМ БУДЕМ ПОНИМАТЬ НОМЕР БАЙТА ОПЕРАТИВНОЙ ПАМЯТИ, ВЫЧИСЛЯЕМЫЙ СПЕЦИАЛЬНЫМ ОБРАЗОМ, а ИМЕННО: АДРЕС = (СОДЕРЖИМОЕ СЕГМЕНТНОГО РЕГИСТРА) + СМЕЩЕНИЕ. ГДЕ ПОД СМЕЩЕНИЕМ ПОНИМАЕТСЯ ЧИСЛО, КОТОРОЕ ОПРЕДЕЛЯЕТ ТО, НА КАКОЕ КОЛИЧЕСТВО БАЙТОВ СЛЕДУЕТ ПЕРЕМЕСТИТСЯ, ЧТОБЫ ДОБРАТЬСЯ ДО БАЙТА С НУЖНЫМ АДРЕСОМ.**

**ЗАМЕЧАНИЕ! Подробно, не так как Вы любите, ИЗУЧИТЕ СЛЕДУЮЩЕЕ ДОМАШНЕЕ ЗАДАНИЕ.**

**Далее будет идти речь О ТАК НАЗЫВАЕМЫХ Memory Models и ОБ АРХИТЕКТУРЕ Intel 64 и IA-32 и соответствующем РУКОВОДСТВЕ ДЛЯ РАЗРАБОТЧИКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ. К СЛЕДУЮЩЕЙ ЛЕКЦИИ ПРИСЛАТЬ МНЕ НА ПОЧТУ ДОКАЗАТЕЛЬСТВО ТОГО, ЧТО НА ВАШЕМ ЛИЧНОМ КОМПЬЮТЕРЕ В СТРОГО ОПРЕДЕЛЕННОЙ ПАПКЕ НАХОДИТСЯ ФАЙЛ, ПРЕДСТАВЛЯЮЩИЙ СОБОЙ ФАЙЛ-ДОКУМЕНТ БОЛЬШОГО РАЗМЕРА (больше 20 мгб) – это самое руководство по архитектуре Intel 64 and IA-32 для РАЗРАБОТЧИКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ. Понятно, что это документация на английском языке и принадлежит правообладателю – корпорации INTEL. Но с сайта Intel его скачать не получится, но оно есть на других сайтах (заодно узнаете причину, почему с сайта Intel скачать не получится). То есть К следующей лекции СКАЧАТЬ БЕСПЛАТНО данный документ и прислать мне на почту ФОТО (не скриншот, а фото), из которого ВИДНО, ЧТО ЭТО ИМЕННО ВЫ И ЧТО ЭТОТ ДОКУМЕНТ НАХОДИТСЯ ИМЕННО НА ВАШЕМ КОМПЕ, а не на компе раба или рабовладельца, ВИДНО, в какой папке он у Вас находится, когда он там был создан и сколько «ОН ВЕСИТ».**

**Продолжаем толкование 29.**

Memory Models.

When employing the processor’s memory management facilities, programs do not directly address physical memory. Instead, they access memory using one of three memory models: flat, segmented, or real address mode:

• Flat memory model — Memory appears to a program as a single, continuous address space. This space is called a linear address space. Code, data, and stacks are all contained in this address space. Linear address space is byte addressable. An address for any byte in linear address space is called a linear address.

• Segmented memory model — Memory appears to a program as a group of independent address spaces called segments. Code, data, and stacks are typically contained in separate segments. To address a byte in a segment, a program issues a logical address. This consists of a segment selector and an offset (logical addresses are often referred to as far pointers). The segment selector identifies the segment to be accessed and the offset identifies a byte in the address space of the segment.. To access a memory location, the processor thus translates each logical address into a linear address. This translation is transparent to the application program. The primary reason for using segmented memory is to increase the reliability of programs and systems. For example, placing a program’s stack in a separate segment prevents the stack from growing into the code or data space and overwriting instructions or data, respectively.

• Real-address mode memory model — This is the memory model for the Intel 8086 processor. It is supported to provide compatibility with existing programs written to run on the Intel 8086 processor. The real-address mode uses a specific implementation of segmented memory in which the linear address space for the program and the operating system/executive consists of an array of segments of up to 64 KBytes in size each.   
  
  
Модели памяти.

При использовании средств управления памятью процессора программы не обращаются напрямую к физической памяти. Вместо этого они обращаются к памяти, используя одну из трех моделей памяти: плоскую, сегментированную или режим реального адреса:

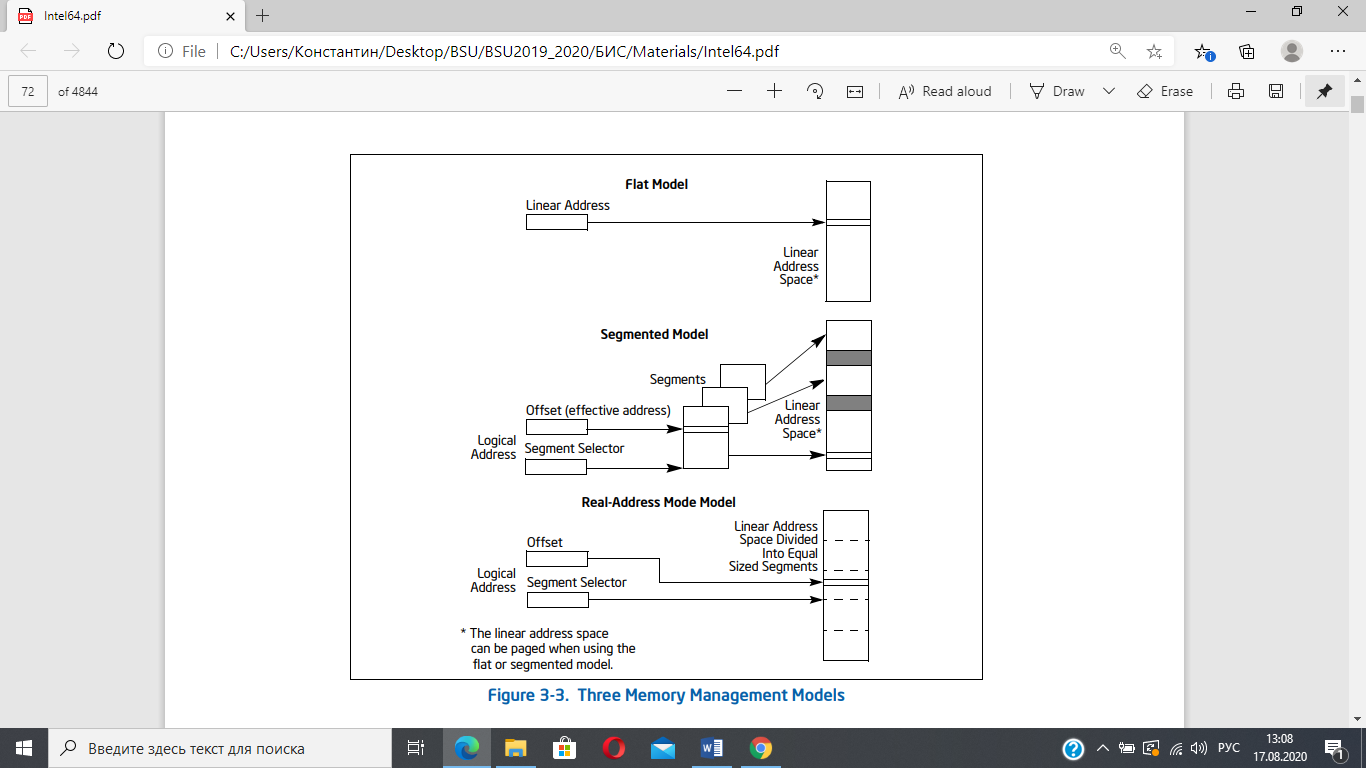
- Плоская модель памяти - память представляется программе как единое, непрерывное адресное пространство. Это пространство называется линейным адресным пространством. Код, данные и стеки содержатся в этом адресном пространстве. Линейное адресное пространство имеет байтовую адресацию. Адрес любого байта в линейном адресном пространстве называется линейным адресом.

- Сегментированная модель памяти - память представляется программе как группа независимых адресных пространств, называемых сегментами. Код, данные и стеки обычно содержатся в отдельных сегментах. Чтобы обратиться к байту в сегменте, программа выдает логический адрес. Он состоит из селектора сегмента и смещения (логические адреса часто называют дальними указателями). Селектор сегмента определяет сегмент, к которому нужно обратиться, а смещение - байт в адресном пространстве сегмента. Чтобы получить доступ к ячейке памяти, процессор переводит каждый логический адрес в линейный адрес. Этот перевод прозрачен для прикладной программы. Основная причина использования сегментированной памяти - повышение надежности программ и систем. Например, размещение стека программы в отдельном сегменте предотвращает его разрастание в пространство кода или данных и перезапись инструкций или данных, соответственно.

- Модель памяти в режиме реальных адресов - это модель памяти для процессора Intel 8086. Она поддерживается для обеспечения совместимости с существующими программами, написанными для работы на процессоре Intel 8086. В режиме реального адреса используется специальная реализация сегментированной памяти, в которой линейное адресное пространство для программы и операционной системы/исполнителя состоит из массива сегментов размером до 64 КБайт каждый.

***ЗАДАНИЕ №5.1. Перевести с английского на русский и записать на русском языке ПРЯМО НА ЗАНЯТИИ на листочках печатными буквами ПЕРЕВОД!!!!***

**Продолжение домашнего задания.** К фото ДОБАВИТЬ word-файл, в котором должен НАХОДИТЬСЯ ТЕКСТ НА РУССКОМ ЯЗЫКЕ, СООТВЕТСТВУЮЩИЙ ТОМУ, ЧТО БЫЛО НАПИСАНО О MEMORY MODELS на листочках. НЕ ИСПОЛЬЗОВАТЬ ПЕРЕВОДЧИКОВ а для незнакомых слов использовать 4-5 перевод из словарей. И добавить СКОПИРОВАННЫЙ ИЗ ОРИГИНАЛЬНОГО ДОКУМЕНТА рисунок 3.3., переведя все английские слова на русский язык. Например, «Flat Model» - «Квартирная модель».



Толкование 30. Для того, чтобы прикладная программа могла быть исполнена, ОПЕРАЦИОННАЯ СИСТЕМА (НЕ ОБОЛОЧКА Windows) должна как-то понять, что прикладная программа хочет попасть в оперативную память, и хочет, чтобы ЕЙ в какой-то момент было передано управление. Надеюсь, понятно, что ПРОГРАММА – это не то, что написано пером или набрано с помощью некоторого программного средства типа Microsoft Word и хранится в виде текстового файла на внешнем устройстве. То, что написано пером или «набрано с помощью некоторого программного средства типа Microsoft Word и хранится в виде текстового файла на внешнем устройстве» следует назвать «ИСХОДНЫЙ ТЕКСТ НА ЯЗЫКЕ ЗАПИСИ АЛГОРИТМОВ». ИСПОЛНИМЫЙ МОДУЛЬ представляет СОБОЙ ОФОРМЛЕННЫЙ ПО ПРАВИЛАМ ЛИБО ОПЕРАЦИОННОЙ СИСТЕМЫ (ЛИБО ОПЕРАЦИОННОЙ ОБОЛОЧКИ) ФАЙЛ, состоящий ИЗ КОМАНД ЛИБО ОПЕРАЦИОННОЙ СИТСЕМЫ (bat-файлы) ЛИБО из готовых к исполнению машинных команд, которые ПРЕДВАРЯЮТСЯ (либо не предваряются) так называемым ЗАГОЛОВКОМ. Например, ИСПОЛНИМЫЕ ФАЙЛЫ, называемые exe-файлами, которые как раз и состоят из заголовка, начинающегося с двух латинских букв MZ и имеющего строго определенные РАЗМЕР И СТРУКТУРУ, и НЕПОСРЕДСТВЕННО СОВОКУПНОСТИ МАШИННЫХ КОМАНД, полученных в результате компиляции (трансляции) исходного текста на языке записи алгоритмов и «сборки» (линковки). Если машинные команды в виде файла на внешнем устройстве НЕ ПРЕДВАРЯЮТСЯ ЗАГОЛОВКАМИ, ЭТОТ ФАЙЛ называется ИСПОЛНИМЫМ ФАЙЛОМ ТИПА COM. Из Лекции 5 ЗУБОВИЧА и лабораторных занятий 1-4 Вам понятно, что для получения подобных файлов необходимо транслировать исходный текст на языке записи алгоритмов низкого уровня (типа ассемблер), осуществить «линчевание» полученного в результате трансляции объектного модуля. Если же речь идёт о языках записи алгоритмов высокого уровня (типа Си), то необходимо компилировать исходный текст на языке записи алгоритмов, а затем «собрать» что-то там с получением исполнимого модуля типа exe.

Существует множество форматов исполняемых файлов для разных операционных оболочек, например COM, EXE (Windows, DOS), COFF, ELF (Unix). Обычно данные, хранящиеся в исполняемом файле, состоят из нескольких частей:

* заголовки (предполагаемые исполнители инструкций, настройки исполнителя инструкций и окружения, формат инструкций). Исполнители инструкций — это аппаратно-программный комплекс, способный исполнить инструкцию (процессоры, микроконтроллеры, интерпретаторы, виртуальные машины);
* инструкции (собственно код программы). Представлены либо машинными инструкциями, либо исходным кодом (для интерпретируемого языка), либо байт-кодом для виртуальной машины;
* дополнение (данные, используемые для отладки, список библиотек и используемых функций, иконки для ярлыков, тексты, изображения, звуки, видео и др.)

Можно прочитать о наиболее распространенных в наши дни исполняемых PE-файлов для 32-разрядных версий операционной оболочки Microsoft Windows.

### Структура формата PE

Формат PE (Portable Executable) — формат исполняемых файлов, динамически подключаемых библиотек dll, драйверов sys и некоторых других, используемый в 32- и 64-разрядных версиях операционной оболочки Windows. Формат обладает большим количеством полей, поэтому далее будут описаны только те, которые необходимы для реализации поставленной задачи.

В самом общем приближении исполняемый файл в формате PE состоит из набора заголовков и секций. Для того чтобы описать, какие именно данные находятся в секции, используются директории. В одной секции могут находиться данные разных типов. Общая структура PE формата изображена на рисунке 1. Отмечены только самые важные поля.

**IMAGE\_NT\_HEADER**

Signature

**IMAGE\_FILE\_HEADER**

NumberOfSections

SizeOfOptionalHeader

Characteristics

**IMAGE\_OPTIONAL\_HEADER**

Magic

ImageBase

AddressOfEntryPoint

SectionAligment

FileAligment

SizeOfHeaders

**IMAGE\_DATA\_DIRECTORY**

VirtualAddress

Size

**IMAGE\_DOS\_HEADER**

e\_magic

e\_lfanew

**IMAGE\_SECTION\_HEADER**

Name

VirtualSize

VirtualAddress

SizeOfRawData

PointerToRawData

Characteristics

**Секция**

**Секция**

*Рисунок 1. Общая структура PE формата*

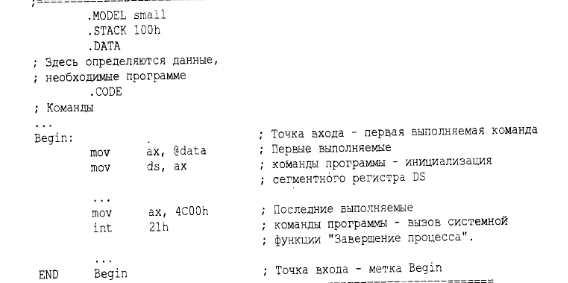
**Задание № 5.2. НА ЛИСТОЧКАХ: ПЕРЕРИСОВАТЬ Рисунок 1, переведя всё С АНГЛИЙСКОГО НА РУССКИЙ. Например, NumberOfSections записать в прямоугольнике как NumberOfSections – КоличествоСекций.**

**Задание № 5.3. НАРИСОВАТЬ СТРУКТУРУ ИСПОЛНИМОГО МОДУЛЯ ТИПА .COM.**

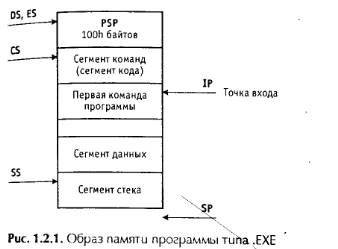
**То, о чём шла речь выше, подводит нас к вопросу о том, как Memory Model связано со структурой исполнимых модулей. Поэтому формулируем**

**Толкование 31. Структура ПРОГРАММ, получаемых в оперативной памяти, в результате ЗАГРУЗКИ под управлением операционной системы ИСПОЛНИМЫХ ФАЙЛОВ (МОДУЛЕЙ) типа .exe и .com.**

**В следующем фрагменте представлен один из возможных вариантов ОФОРМЛЕНИЯ ИСХОДНОГО ТЕКСТА НА ЯЗЫКЕ ЗАПИСИ АЛГОРИТМОВ assembler с возможностью получения в результате трансляции и линковки ИСПОЛНИМОГО ФАЙЛА ТИПА .EXE.**



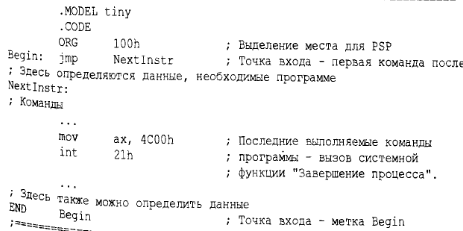
**Если соответствующим и правильным образом осуществить ТРАНСЛЯЦИЮ (tasm) и ЛИНКОВКУ (tlink) в рамках использования программного средства DosBox с получением файла .exe, а затем осуществить размещение уже ПРОГРАММЫ в оперативную память, ЧТО ВЫ И ДОЛЖНЫ БЫЛИ ДЕЛАТЬ ПРИ РЕШЕНИИ ЗАДАЧ ПРО ПАПУ, МАМУ и ТАБЛИЦУ, то в оперативной памяти «появится» такая «Картинка»:**



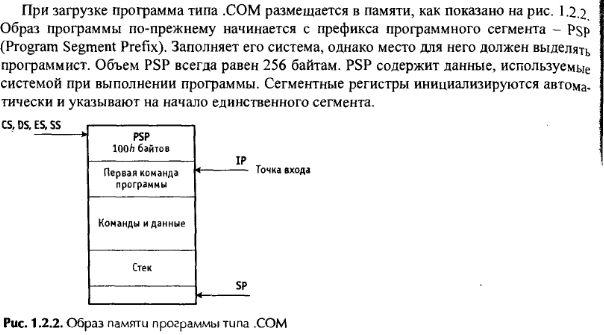
**ЗАДАНИЕ № 5.4. ЧТО ТАКОЕ PSP.**

**В конце лекции сдаём листочки.**

**Если же возникнет желание В РЕЗУЛЬТАТЕ ТРАНСЛЯЦИИ И ЛИНКОВКИ получить исполнимый файл ТИПА .COM, то исходный текст может быть оформлен следующим образом:**



**При этом указание, что должен быть получен файл типа .COM, делается с помощью указания при вызове TLINK ключика, если не обманываю, \t! В результате будет получена такая картинка в оперативной памяти:**



**Замечание. В зависимости от того, какой транслятор (tasm, masm, nasm, fasm, …. и какой линковщик используются, ОФОРМЛЕНИЕ ИСХОДНЫХ ТЕКСТОВ НА ЯЗЫКЕ ЗАПИСИ АЛГОРИТМОВ assembler МОЖЕТ РАЗЛИЧАТЬСЯ. Однако основным принципом является то, что должны быть как-то явно или неявно определены Сегменты и с каждым из них «связаны» сегментные регистры, в которые сначала транслятор, а затем линковщик с загрузчиком операционной системы, РАЗМЕЩАЮТСЯ АДРЕСА ОБЛАСТЕЙ ПАМЯТИ, в которых будут расположены соответствующие сегменты. Просто напоминание: АДРЕС – Номер байта ОПЕРАТИВНОЙ ПАМЯТИ!!!!**

**Из этих рисунков следует, что из исходного текста, являющегося исходным для исполнимого модуля типа Com, в сегментные регистры CS, DS, ES и SS помещается адрес специальной области памяти под названием PSP (не переводим) – Program Segment Prefix и размером в 100h байтов (в 256 байтов).**

**Толкование 32. Область Program Segment Prefix имеет строго определенную структуру и формируется операционной системой при ЗАГРУЗКЕ ИСПОЛНИМОГО ФАЙЛА В ОПЕРАТИВНУЮ ПАМЯТЬ!!! На помойке можно с легкостью найти кучу вариантов описаний этой области. Но с этими вариантами будем знакомиться по нужде – когда в этом возникнет необходимость при рассмотрении тех или иных вопросов, связанных с темой той или иной лекции.**

ПРОДОЛЖЕНИЕ (ОКОНЧАНИЕ) ДОМАШНЕГО ЗАДАНИЯ. После Рисунка 3.3. в сформированном word-файле НАПИСАТЬ (НАРИСОВАТЬ) ответ на следующий вопрос. Описанное распределение оперативной памяти с выделением PSP для загружаемых ИСПОЛНИМЫХ МОДУЛЕЙ ТИПА com и exe характерно для 32-разрядных ЭВМ. Для 64-разрядных ЭВМ PSP отсутствует. Но присутствует нечто другое, описание которого можно найти в той документации, которое Вы, надеюсь скачали. Вопрос: Указать, если таковой существует, на какой странице, в каком разделе дано описание аналога PSP для 64-разрядных ЭВМ, и если существует, указать аббревиатуру и её расшифровку, а также изобразить структуру этого аналога, скопировав её из документации.

ДОМАШНЕЕ ЗАДАНИЕ ОФОРМИТ В ФАЙЛЕ Word И ПРИСЛАТЬ МНЕ НА ПОЧТУ ДО 23-59 6.10.2024 года (до конца воскресенья). Удачи.