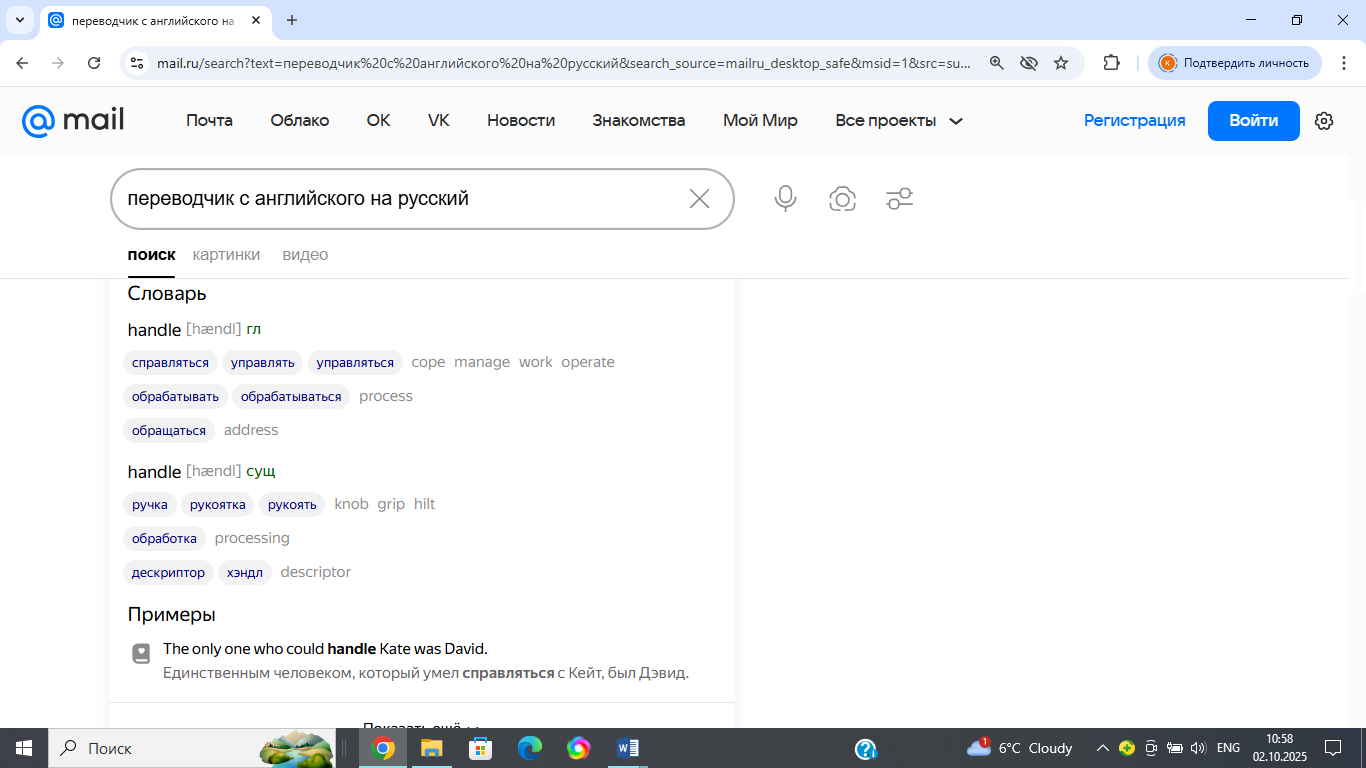
Лекция 5. Снова неадекватная.

Прошлая лекция-контрольная тоже была неадекватной и она Вам не понравилась – 95% присутствовавших на лекции записали на листочках «Нет» - не надо нам таких лекций. А 5% получат дополнительно 16 баллов в лекционный рейтинг за то, что ответили «Да» на вопрос: «Нужны ли такие лекции?». Сегодня ударимся в другую крайность. Данную лекцию ЧИТАЕМ и просто ДУРКуем! На листочках ничего не пишем, занимаемся чем хотим (я буду ходить и просто смотреть, кто чем занимается), а в начале лекции я просто отмечу присутствующих. Однако на следующей лекции Вы будете письменно спрошены, что у Вас осталось после сегодняшней лекции. И что такое «ДУРКовать»?

***Толкование 25. Для того, чтобы в чём-то разобраться, кроме правил «3-х компасов», «Разделяй и властвуй», «Ограничение контекста», «Хорошего рассказывания» следует пользоваться ещё тремя принципами, которые могут сделать из Вас хотя бы простых кодировщиков, если не разработчиков программного обеспечения: 1. «Делать всё головой» - Думать, а не стучать пальцами по клавиатуре и лазить по помойке, а Учиться. 2. Не говорить, не слушать, не писать, а РИСОВАТЬ!!! 3. «Копать» отсюда и до обеда, и после обеда, и после ужина. То есть получается принцип: ДУРКовать!***

***Замечание 1. В дальнейшем, так как в рамках курса БИС разрешено использовать только информацию с сайтов правообладателей тех или иных продуктов, придётся много переводить с английского на русский. Если Вы будете использовать какие-либо «переводчики» или англо-русские словари, договариваемся, что при переводе НУЖНО БРАТЬ 4-5 вариант перевода слова за основу, а не первое попавшееся, и конечно же, «ограничивать контекст». Например, в дальнейшем будем иметь дело со словом handle. Если просто залезть в «переводчик Гугел», то получится перевод:***



***И станет понятно, что в контексте вычислительной техники переводить слово «handle» (существительное) нужно как «дескриптор», но никак не «ручка». А это как раз 5-ый (ПЯТЫЙ) вариант перевода.***

***Очень похоже, что я Вас обманул в предыдущих абзацах. Догадаетесь, где обманул?***

***Замечание 2. Ещё раз. Словосочетание «Загрузка операционной системы» само по себе БЕССМЫСЛЕННО, так как непонятно, во-первых, куда что «грузится», во-вторых, как «аппаратные средства, являющиеся составной частью ОС» могут быть куда-либо загружены! Поэтому просьба, говоря о том, что обычно подразумевается под «загрузкой операционной системы», ГОВОРИТЬ ТОЛЬКО СЛЕДУЮЩИМ ОБРАЗОМ: «Размещение В ОПЕРАТИВНОЙ ПАМЯТИ ПРОГРАММНЫХ СРЕДСТВ, являющихся составной частью ОС». Договорились? А чтобы сие замечание усилить, читаем***

***Толкование 26.***

<http://on-line-teaching.com/IBM-computer/24_OS.html>

**Процесс подготовки компьютера к работе**

1. **Нажатие кнопки включения питания.**

При включении кнопки Power на элементы материнской платы поступает питающее напряжение; по сигналу Power Good запускается тактовый генератор; исполняются команды, зашитые в аппаратной части BIOS.

1. **Проверка BIOS.**

Так называемая «контрольная сумма» пересчитывается и сравнивается с эталонным значением.

1. **Идентификация процессора.**

Материнская плата предусматривает возможность установки различных моделей процессора. Аппаратная часть BIOS подает запрос на идентификацию процессора и по полученному ответу определяет тип процессора, частоту, напряжение и проч.

1. **Настройка базовых элементов.**

Инициализируются и тестируются базовые компоненты системной платы: блок прямого доступа к памяти, таймер, блок аппаратных прерываний.

1. **Тестирование ОЗУ.**

Определяется тип модулей памяти, их объем, организация; тестируются первые 64 Кб оперативной памяти.

1. **Организация рабочих структур ОЗУ.**

Выделяется область под программную составляющую BIOS, настраиваются прерывания.

1. **Проверка CMOC-памяти и батарейки.** ???????

**И что это за CMOC?** Вопрос в пустоту.

1. **Инициализация устройств материнской платы.**

Производится поиск и настройка загрузочных устройств (жесткий диск, привод CD, FDD), средств управления процессом загрузки (клавиатура, мышь), устройств ввода-вывода (COM, LPT). Устройствам [выделяются соответствующие линии прерывания](http://on-line-teaching.com/IBM-computer/22_IRQ_CMOS.html).

1. **PnP.**

Идентифицируются устройства, подключенные через системные разъемы. Устройствам выделяются ресурсы и прерывания.

1. **Включение видеосистемы.**

Запускается Video BIOS, который настраивает видеоконтроллер на режим VGA или EGA, которые поддерживают все видеоконтроллеры. После этого видеоконтроллер готов к работе.

1. **Выдача сообщения на экран монитора.**

На экране монитора появляется первое сообщение: фирма-производитель BIOS, тип и частота процессора, тип и объем ОЗУ.

1. **Тестирование ОЗУ.**

Производится выборочная проверка незадействованной оперативной памяти.

1. **Инициализация контроллера дисководов.**
2. **Инициализация контроллера жестких дисков.**
3. **Инициализация клавиатуры.**

Включается контроллер клавиатуры, производится тест матрицы контактов, устанавливаются временные параметры опроса клавиш и режим NumLock. Клавиатура готова к работе. На экран выводится сообщение о возможности использования программы BIOS Setup (обычно для этого используется клавиша Del).

1. **Поиск устройств с собственным BIOS.**

Если таковые устройства найдены, то управление передается BIOS-программам этих устройств и происходит их инициализация.

1. **Передача управления загрузчику программной части ОС.**

По программному прерыванию Int 19h на дисковых накопителях ищется загрузчик программной части ОС (Boot Record). Он должен находиться на одном из устройств (HDD, CD, FDD, SCSI). Местоположение загрузчика везде одинаково. После того, как загрузчик программной части ОС найден, управление передается ему.

***Толкование 27. Всё, что может ЭВМ, каким-то образом связано с процессором и оперативной памятью. Причём распределением памяти (управлением просто устройством ЭВМ) заведует «операционная система» (я не знаю, как называются операционные системы в нынешних ЭВМ, поэтому будем считать их безымянными).***

**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 (АРИФМЕТИКО-ЛОГИЧЕСКОЕ УСТРОЙСТВО) 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 СЛЕДУЕТ ПОНИМАТЬ НОМЕР БАЙТА ОПЕРАТИВНОЙ ПАМЯТИ, ВЫЧИСЛЯЕМЫЙ СПЕЦИАЛЬНЫМ ОБРАЗОМ, а ИМЕННО: АДРЕС = (СОДЕРЖИМОЕ СЕГМЕНТНОГО РЕГИСТРА) \*16 + СМЕЩЕНИЕ. ГДЕ ПОД СМЕЩЕНИЕМ ПОНИМАЕТСЯ ЧИСЛО, КОТОРОЕ ОПРЕДЕЛЯЕТ ТО, НА КАКОЕ КОЛИЧЕСТВО БАЙТОВ СЛЕДУЕТ ПЕРЕМЕСТИТЬСЯ, ЧТОБЫ ДОБРАТЬСЯ ДО БАЙТА С НУЖНЫМ АДРЕСОМ.**

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

**Далее будет идти речь О ТАК НАЗЫВАЕМЫХ Memory Models и 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). К следующей лекции СКАЧАТЬ БЕСПЛАТНО данный документ и прислать мне на почту ФОТО (не скриншот, а фото (селфи)), 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», а также дата и время, когда сделано фото.**

**Толкование 28.**

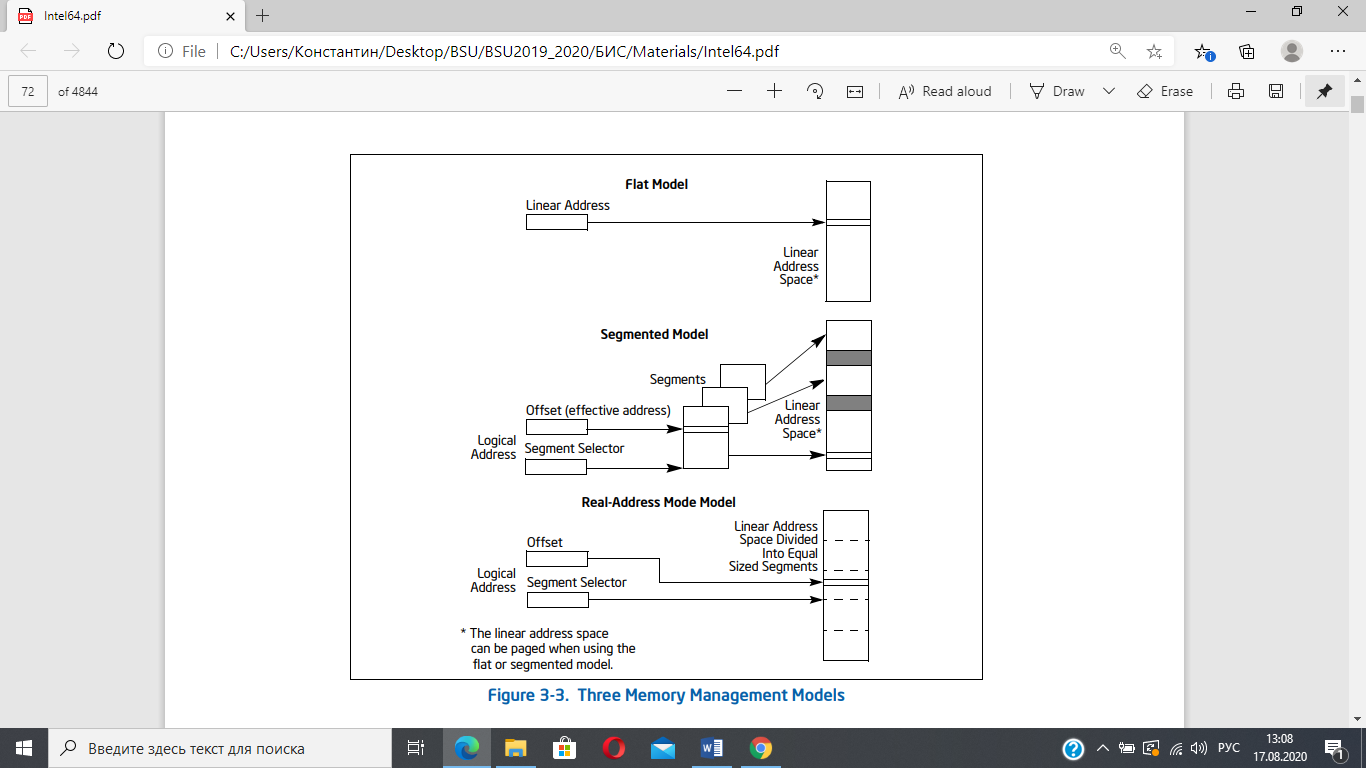
Memory Models.

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.

**Продолжение домашнего задания.** К фото ДОБАВИТЬ word-файл, в котором должен НАХОДИТЬСЯ ТЕКСТ НА РУССКОМ ЯЗЫКЕ, СООТВЕТСТВУЮЩИЙ ТОМУ, ЧТО НАПИСАНО выше на английском языке О MEMORY MODELS. НЕ ИСПОЛЬЗОВАТЬ ПЕРЕВОДЧИКОВ, а для незнакомых слов использовать 4-5 перевод из словарей (см. ранее записанное замечание). 

Толкование 29. 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 текст программы на машинном языке). Представлены машинными инструкциями;
* 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 основные.

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**

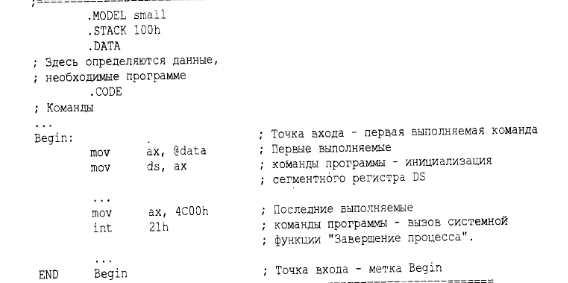
*Рисунок 1. Общая структура PE формата*

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

**Толкование 30. Структура ПРОГРАММ, получаемых в оперативной памяти, в результате ЗАГРУЗКИ под управлением операционной системы ИСПОЛНИМЫХ ФАЙЛОВ (МОДУЛЕЙ) типа .exe и .com.**

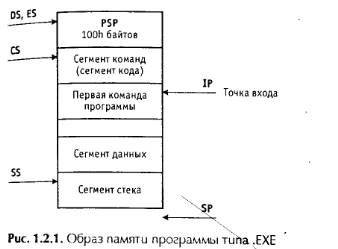
**Замечание. ОЧЕНЬ ВНИМАТЕЛЬНО НУЖНО СМОТРЕТЬ НА ТЕКСТ на языке записи алгоритмов ассемблер и на картинки!!! ОСОБОЕ ВНИМАНИЕ УДЕЛИТЬ ТОМУ, КУДА НА КАРТИНКАХ УКАЗЫВАЮТ СТРЕЛОЧКИ И ЧТО НА ЭТИХ СТРЕЛОЧКАХ НАРИСОВАНО (КАКИЕ РЕГИСТРЫ). Очень даже может быть, что рисуночки, взятые из известной Вам книги, Вас обманывают.**

**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. И именно так был оформлен текст на языке записи алгоритмов ассемблер в задаче про папу с мамой и рамой.**

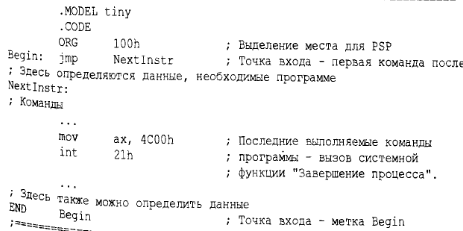


**Знакомо?**

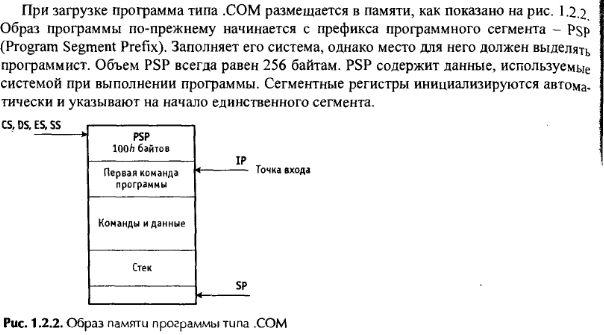
**Если соответствующим и правильным образом осуществить ТРАНСЛЯЦИЮ (tasm) и ЛИНКОВКУ (tlink) в рамках использования программного средства DosBox с получением файла .exe, а затем осуществить размещение уже ПРОГРАММЫ в оперативную память, ЧТО ВЫ И ДОЛЖНЫ БЫЛИ ДЕЛАТЬ ПРИ РЕШЕНИИ ЗАДАЧ ПРО ПАПУ, МАМУ и РАМУ,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



**Если же возникнет желание В РЕЗУЛЬТАТЕ ТРАНСЛЯЦИИ И ЛИНКОВКИ получить исполнимый файл ТИПА .COM, то исходный текст может быть оформлен следующим образом (так нужно было оформлять текст на языке записи алгоритмов ассемблер для решения задачи про 36 байтов):**



**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 транслятором, а затем линковщиком 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 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 размером в 100h байтов (в 256 байтов).**

**Толкование 31. 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. В сформированном word-файле перевода с английского на русский 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? (Обманул?). Но присутствует нечто другое, описание которого можно найти в той документации, которое Вы, надеюсь скачали.

**Вопрос:** Указать, 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.

После ответа на этот вопрос ЗАПИСАТЬ БОЛЬШИМИ БУКВАМИ (ТОЧНО ТАКИМИ ЖЕ КАК ЗДЕСЬ) одно из двух слов – либо «ДА», либо «НЕТ» в качестве ответа на ещё один вопрос: «Стоит ли в дальнейшем практиковать подобные лекции?».

Замечание. Таким образом, к следующей лекции Вы должны прислать мне на почту: 1. Селфи с компом и файлом про Intel. 2. Word-файл с домашним заданием. ВНИМАТЕЛЬНО ЧИТАЙТЕ УСЛОВИЯ!!!!! Квалификация разработчика программного обеспечения определяется уровнем соответствия решения задачи её постановке.