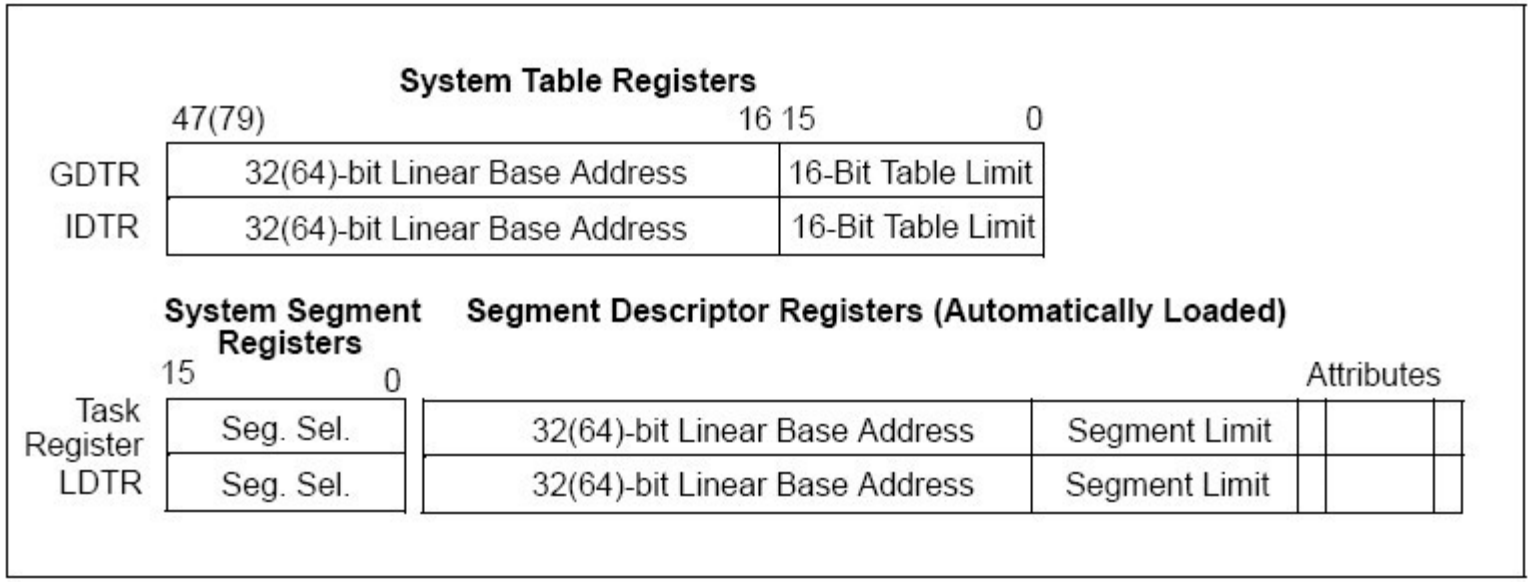
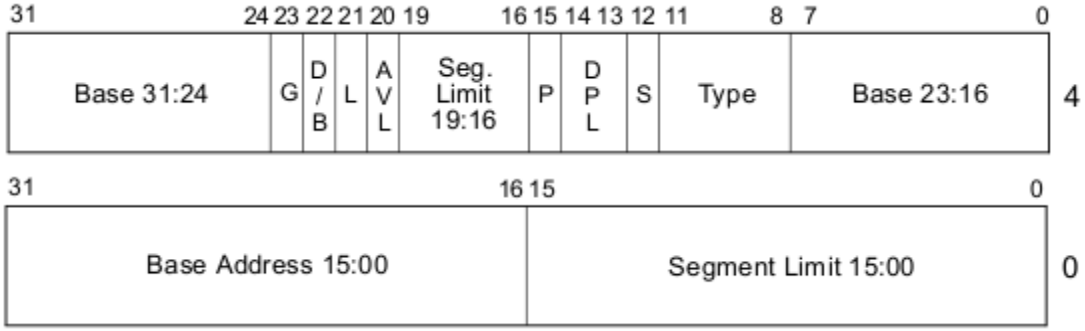
Введение в защищённый режим.
Итак, PM значительно отличается от всем привычного ещё со времён DOS'а реального режима (RM). Теперь придётся привыкать: здесь нет статичных, 64 килобайтных сегментов, таблицы прерываний в 1'ом килобайте, адресов баз сегментов в сегментных регистрах, в общем совершенно новый мир.
Теперь сегменты описываются в **Global Descriptor Table (GDT)**. Сия таблица может быть только в одном экземпляре. Она структура в памяти. Не сегмент! Может располагаться в памяти где угодно, но её адрес и лимит записываются в регистр GDTR. Вот его структура:



Сама таблица состоит из записей следующей структуры (кстати нулевая запись пустая. Это важно. При обращении к памяти, 'описываемой' нулевым дескриптором, получите #GP – General Protection Fault):
Давайте рассмотрим эту структуру повнимательней.



1. Segment Limit:
Назначение этого поля понятно по названию, но есть тонкость. Собака зарыта в бите G (Granularity).
Если он неустановлен, то память ‘отсчитывается’ в байтах. В таком случае размер сегмента может варьироваться от 1 байта до 1 мегабайта на размер в 1 байт.
Если установим его в 1, то будет введена страничная адресация памяти. Тогда мы сможем адресовать от 4 килобайт до 4 гигабайт оперативки с изменением размера на 4 килобайта (размер страницы). Вообще страничная адресация предпочтительней (сравните (1Мб+64Кб-16байт) и 4Гб). Давайте в этом посте поговорим только о сегментной адресации. Paging заслуживает отдельного разговора.

2. Base Address:
Здесь указываем физический адрес базы.

3. Type field:
Комбинации битов определяют тип сегмента:

№	Поле Тип				Тип дескриптора	Описание
	11	10 E	9 W	8 A		
0	0	0	0	0	Данные	Только для чтения
1	0	0	0	1	Данные	Только для чтения, доступен
2	0	0	1	0	Данные	Для чтения/записи
3	0	0	1	1	Данные	Для чтения/записи, доступен
4	0	1	0	0	Данные	Только для чтения, растет вниз
5	0	1	0	1	Данные	Только для чтения, растет вниз, доступен
6	0	1	1	0	Данные	Для чтения/записи, растет вниз
7	0	1	1	1	Данные	Для чтения/записи, растет вниз, доступен
		C	R	A		
8	1	0	0	0	Код	Только для исполнения
9	1	0	0	1	Код	Только для исполнения, доступен
10	1	0	1	0	Код	Для исполнения/чтения
11	1	0	1	1	Код	Для исполнения/чтения, доступен
12	1	1	0	0	Код	Только для исполнения, подчинен
13	1	1	0	1	Код	Только для исполнения, подчинен, доступен
14	1	1	1	0	Код	Для исполнения/чтения, подчинен
15	1	1	1	1	Код	Для исполнения/чтения, подчинен, доступен

4. S (descriptor type):
В документации интеловской сказано, что если этот бит не установлен, то этот дескриптор для системного сегмента, иначе – кода или данных. Под системным подразумевается LDT, TSS, Interrupt Gates и иже с ними (о них позже).

5. DPL (Descriptor Privilege Level):
Привилегии описываемого сегмента. Всем знакомые Rings.

6. P (segment present):
Если этот бит установлен, то процессор ‘знает’, что сегмент в уже памяти (хотя лучше сказать валидный). Если загрузите в сегментный регистр селектор дескриптора с неустановленным битом P, то произойдёт исключение #NP (not present). Вообще смысл этой витиеватой фразы объясню чуть позже.

7. D/B:
Для сегментов разного типа по-разному трактуется.
1. Для сегментов кода:
32 или 16 битная длина эффективного адреса и размерность операндов.

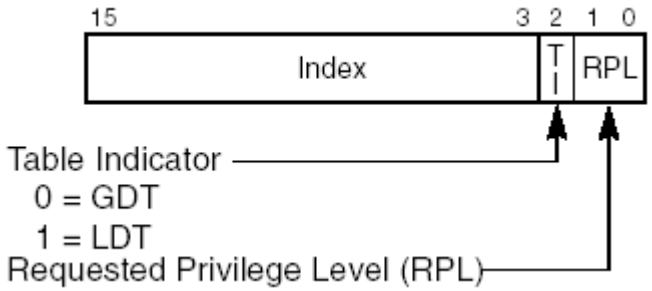
(1-32; 0-16);
2. Для стека:
Указатель стека 32 или 16 битный. (1-32; 0-16);

8. G:
Влияет на то, в каких единицах (байты, страницы) измеряется лимит сегмента. Вообще Paging можно включить при переходе в PM, установив 31 бит регистра CR0.

Ещё немного информации:
Догадываемся, что слово Global поставили не напрасно. Значит есть ещё какая-то табличка. Верно, есть также **Local Descriptor Table**. Их может быть великое множество. К примеру они могут использоваться в реализации задач и.т.д. А вот **LDT** уже представляет собой сегмент! Так что привыкайте к фразам типа ‘дескриптор сегмента локальной таблички дескрипторов’.

После того, как мы описали таблицу, нужно ей загрузить в регистр **GDTR**. Это делается далеко не mov’ом. **GDTR** заполняется командой **lgdt fword (значение)**. То есть надо сформировать самостоятельно эту структуру и загрузить в вышеупомянутый регистр. Есть ещё команды работы с этим регистром, но мы несёмся галопом по Европам.

Ещё один момент. В PM в сегментных регистрах хранятся не базовые адреса сегментов (как в RM), а специально обученные штуки, под названием **селекторы**. Их структура такова:



Здесь Index – порядковый номер дескриптора в таблице.
TI показывает где искать дескриптор (в **GDT** или **LDT**).

Теперь, когда уже понятно как строить таблицу, поговорим о том, как перейти в PM (замечу, это можно сделать только из RM). Вообще … нужно всего установить бит 0 управляющего регистра CR0. Хотя вру. Для начала нужно запретить все прерывания (**NMI** (**Non Maskable Interrupts**) в том числе), открыть адресную линию **A20** (чтобы была доступна 32-битная адресация), загрузить **GDTR**, и прыгнуть на метку – старт.