GDT и LDT — таблицы глобальных и локальных дескрипторов соответственно. Это таблицы восьмибайтных структур, называемых дескрипторами сегментов, в которых и находится начальный адрес сегмента вместе с другой важной информацией:

слово 3 (старшее):
биты 15 – 8: биты 31 – 24 базы
бит 7: бит гранулярности (0 — лимит в байтах, 1 — лимит в 4-килобайтных единицах)
бит 6: бит разрядности (0/1 — 16-битный/32-битный сегмент)
бит 5: 0
бит 4: зарезервировано для операционной системы
биты 3 – 0: биты 19 – 16 лимита

слово 2:
бит 15: бит присутствия сегмента
биты 14 – 13: уровень привилегий дескриптора (DPL)
бит 12: тип дескриптора (0 — системный, 1 — обычный)
биты 11 – 8: тип сегмента
биты 7 – 0: биты 23 – 16 базы

слово 1: биты 15 – 0 базы
слово 0 (младшее): биты 15 – 0 лимита

Два основных поля в этой структуре — база и лимит сегмента. База представляет линейный 32-битный адрес начала сегмента, а лимит — это 20-битное число, которое равно размеру сегмента в байтах (от 1 байта до 1 мегабайта), если бит гранулярности сброшен в ноль, или в единицах по 4096 байт (от 4 Кб до 4 Гб), если бит гранулярности установлен в 1. Числа отсчитываются от нуля, так что лимит 0 соответствует сегменту длиной 1 байт, точно так же, как база 0 соответствует первому байту памяти.

Остальные элементы дескриптора выполняют следующие функции:

Бит разрядности: для сегмента кода этот бит указывает на разрядность операндов и адресов по умолчанию. То есть в сегменте с этим битом, установленным в 1, все команды будут интерпретироваться как 32-битные, а префиксы изменения разрядности адреса или операнда будут превращать их в 16-битные, и наоборот. Для сегментов данных этот бит управляет тем, какой регистр (SP или ESP) используют команды, работающие с этим сегментом данных как со стеком.

Поле DPL определяет уровень привилегий сегмента.

Бит присутствия указывает, что сегмент реально есть в памяти. Операционная система может выгрузить содержимое сегмента из памяти на диск и сбросить этот бит, а когда программа попытается к нему обратиться, произойдет исключение, обработчик которого снова загрузит содержимое этого сегмента в память.

Бит типа дескриптора — если он равен 1, сегмент является обычным сегментом кода или данных. Если этот бит — 0, дескриптор является одним из 16 возможных видов, определяемых полем типа сегмента.

Тип сегмента: для системных регистров в этом поле находится число от 0 до 15, определяющее тип сегментов (LDT, TSS, различные шлюзы). Для обычных сегментов кода и данных эти четыре бита (биты 11 – 8) выполняют следующие функции:
бит 11: 0 — сегмент данных, 1 — сегмент кода
бит 10: для данных — бит направления роста сегмента
для кода — бит подчинения
бит 9: для данных — бит разрешения записи
для кода — бит разрешения чтения
бит 8: бит обращения

Бит обращения устанавливается в 1 при загрузке селектора этого сегмента в регистр.

Бит разрешения чтения/записи выбирает разрешаемые операции с сегментом — для сегмента кода это могут быть выполнение или выполнение/чтение, а для сегмента данных — чтение или чтение/запись.

Бит подчинения указывает, что данный сегмент кода является подчиненным. Это значит, что программа с низким уровнем привилегий может передать управление в этот сегмент и текущий уровень привилегий не изменится.

Бит направления роста сегмента обращает смысл лимита сегмента. В сегментах с этим битом, сброшенным в ноль, допустимы все смещения от 0 до лимита, а если этот бит — 1, то допустимы все смещения, кроме смещений от 0 до лимита. Про такой сегмент говорят, что он растет сверху вниз, так как если лимит, например, равен –100, допустимы смещения от –100 до 0, а если лимит увеличить, станут допустимыми еще меньшие смещения.