# Global Descriptor Table (GDT)

In protected mode, most of the memory management and Interrupt Service Routines (ISR) are controlled through tables of descriptors. Each descriptor stores information about a single object (e.g. a service routine, a task, a chunk of code or data) the CPU might need at some time. Intel defined 3 types of tables: the Interrupt Descriptor Table (replaces IVT), the Global Descriptor Table (GDT) and the Local Descriptor Table. Each table is defined as a (size, linear address) to the CPU through the LIDT, LGDT, LLDT instructions respectively.

## Glossary

### Segment

A logically contiguous chunk of memory with consistent properties (CPU’s speaking)

### Segment Register

A register of your CPU that refers to a segment for special use (e.g. SS, CS, DS …)

### Selector

A reference to a descriptor you can load into a segment register. It is an offset of a descriptor table entry. These entries are 8 bytes long. Bits 3 and up only declare the descriptor table entry offset, while bit 2 specifies if this selector is a GDT or LDT selector (LDT – bit set, GDT – bit cleared), and bits 0 – 1 declare the ring level that needs to correspond to the descriptor table entry’s DPL field. If it doesn’t, a General Protection Fault occurs; if it does correspond then the CPL level of the selector used changed accordingly.

### Descriptor

A memory structure (part of table) that tells the CPU the attributes of a given segment

## What should I put in my GDT?

### Basics

For sanity purpose, you should always have these items stored in your GDT:

* The null descriptor which is never referenced by the processor. Some emulators will complain about limit exceptions if you do not have one present. Some use this descriptor to store a pointer to the GDT itself (to use with the LGDT instruction). The null descriptor is 8 bytes wide and the pointer is 6 bytes wide, so it might just be the perfect place for this.
* A code segment descriptor (for your kernel, it should have type = 0x9a)
* A data segment descriptor (you can’t write to a CS, so add this with type = 0x92
* A TSS (Task Stack Segment) descriptor.

## Structure