**1.核心结构体**

* **efuse\_req 结构体** ：用于管理请求。包含请求的元数据（如 in 和 out 结构体）、标志位（如 FR\_BACKGROUND 和 FR\_INTERRUPTED）以及引用计数。

**2.核心函数**

* **efuse\_process\_init\_reply 函数** ：该函数用于处理初始化回复。与原始的 FUSE 初始化处理函数相比，它在处理逻辑上有一些差异。例如，它会根据回复中的信息来设置 fuse\_conn 结构体中的各种参数，如 max\_background 和 congestion\_threshold，并且会根据协议版本的不同进行相应的处理。它还会设置文件系统的时间粒度、读取页面等参数，并最终决定初始化是否成功。
* **efuse\_send\_init 函数** ：用于发送初始化请求。它会创建一个 efuse\_req 结构体，设置初始化请求的相关参数，如 major、minor、max\_readahead 和 flags 等，然后将其通过 efuse\_simple\_background 函数发送。与原始的 FUSE 初始化发送函数相比，它在构造请求的方式和使用的新结构体 efuse\_req 上有所不同。
* **efuse\_statfs 函数** ：用于获取文件系统统计信息。它与原始的 FUSE statfs 函数类似，但实现上有所区别。它会创建一个 efuse\_req 结构体来发送 FUSE\_STATFS 请求，并在收到回复后将统计信息转换为 kstatfs 结构体。转换时，它会设置文件系统类型为 EFUSE\_SUPER\_MAGIC，而不是原始的 FUSE\_SUPER\_MAGIC。
* **efuse\_send\_destroy 函数** ：用于发送销毁请求。它会创建一个 efuse\_req 结构体，设置销毁请求的相关参数，如 opcode 为 FUSE\_DESTROY，然后通过 efuse\_simple\_request 函数发送。与原始的 FUSE 销毁发送函数相比，它同样在使用新结构体 efuse\_req 和具体的实现逻辑上有所不同。