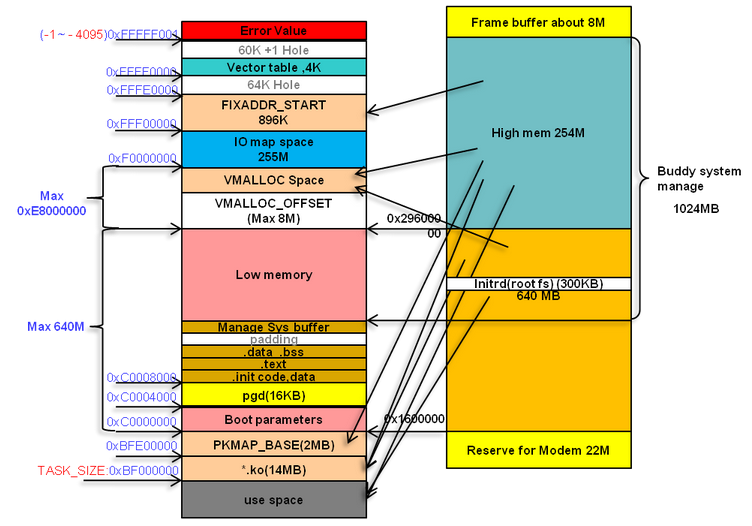
**kernel space**

在分析KE前，你要了解kernel内存布局，才知道哪些地址用来做什么，可能会是什么问题。目前智能机已进入64bit，因此就存在32bit布局和64bit布局，下面一一讲解。

**ARM32bit kernel布局**

这是一张示意图（有些地址可能会有差异）



整个地址空间是4G，kernel被配置为1G，程序占3G。

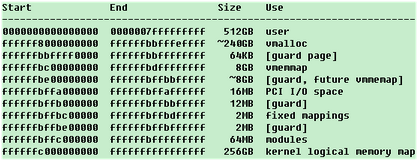
任何程序都有TEXT（可执行代码）,RW（数据段）,ZI段（未初始化数据段），kernel也有，对应的是.text,.data,.bss。而内核代码开始的地址是0xC0008000，前面放页表（起始地址为0xC0004000），如果支持模块（\*.ko）那么地址在0xBF000000。

由于kernel没办法将所有内存都映射进来，毕竟kernel自己只占1G，如果RAM超过1G，就无法全部映射。怎么办呢？只能先映射一部分了，这部分叫low memory。其他的就按需映射，VMALLOC区域就是用于按需映射的。

ARM的外设寄存器和内存一样，都统一地址编码，因此0xF0000000以上的一段空间用于映射外设寄存器，便于操作硬件模块。

0xFFFF0000是特殊地址，CPU用于存放异常向量表，kernel异常绝大部分都是CPU异常（MMU发出的abort/undef inst.等异常）。

**ARM64bit kernel布局**



这是39bit的kernel空间，由于多达512GB的空间，因此完全可以将整个RAM映射进来，0xFFFFFFC000000000之后就是一一映射了，就无所谓high memory了。

vmalloc还是存在，因为可以将不连续的物理内存拼接成连续的虚拟内存，可以解决部分内存碎片问题。而且外设寄存器也直接映射到vmalloc了，就没有32bit布局里的IO map space了。

modules对应的就是\*.ko内核模块了。

以上是粗略的说明，还需查看代码获取完整的分析信息（内核在不停演进，有些部分可能还会变化）。