### 一、堆栈和内存模型



模型A：带有内核保护的堆栈模型，分内核堆栈和用户堆栈，可对MPU提供良好支持。

1. 每个任务一个内核栈，一个用户栈，所有任务内核栈大小一致，用户栈可通过任务创建接口配置。
2. 提供“进入内核模式(特权级)”的内核请求服务，在内核函数的入口处调用该请求。
3. 提供“返回用户模式(用户级)”的内核请求服务，在内核函数的返回处调用该请求。
4. 内存划分为内核数据(全局变量)，内核堆(kmalloc)，用户数据(全局变量)，用户堆(malloc)。
5. 需要MPU提供保护，Kernel .data和heap配置为特权级可写，用户级不可写（仅特权级可写）。
6. 需要MPU提供保护，User .data和heap配置为用户级可写(用户级和特权级都可写)。
7. 需要MPU提供保护，内核堆栈仅允许特权级可写，用户堆栈仅允许当前任务可写。

模型B：不带内核保护的堆栈模型，用户代码和内核代码均运行在内核模式。

1. 每个任务只有一个用户栈，内核代码和用户代码均使用用户栈。
2. 不划分内核模式和用户模式。
3. 不需要MPU提供保护。

### 二、系统结构



以上列表中所列模块和功能并不完全，应当在实际开发过程中必要时进行补充。

**内核功能：**

Object: 对象管理，涉及内核对象的分配、释放，通过链表完成不同类型的对象管理，对象类型包括Semaphore、Mutex、Event、Task、File、Device、Stack等等。

KMemory：核心内存管理，对于模型A，所有内核对象通过kmalloc从核心内存中分配，核心内存的容量由BOARD中的Build config提供配置。对于模型B，kmalloc直接调用lite heap分配。

Clock：提供一个与SysTick同步的时钟计数器，并提供相关接口。

Schedule：提供调度管理，根据任务优先级和活动任务选择最高优先级活动任务进行调度，并提供相同优先级的时间片轮转。完整的任务调度流程参考第三部分-关键流程。

Task：提供任务对象管理功能，例如：任务对象的初始化，任务链表的节点处理、任务的激活和挂起、所有任务的公共入口函数等。

Idle：系统空闲任务，看门狗。

Tls/Errno：线程局部存储空间(参考pthread\_key\_create和TlsAlloc)和错误码处理。

Syscall Service：提供系统调用服务，例如：任务切换、进入和退出特权级等等。

Exception handler：异常处理，提供系统异常处理，例如非法内存访问、硬件故障等（属于ARCH还是kernel需要讨论）。

IRQ：中断管理，

Exception