z-stack代码分析--osalInitTasks函数

```
//任务初始化函数
/*********************************************************************
* @fn osalInitTasks
*
* @brief This function invokes the initialization function for each task.
*
* @param void
*
* @return none
*/
void osalInitTasks( void )
{
uint8 taskID = 0;

//osal_mem_alloc为当前OSAL中各任务分配存储空间，函数返回指向任务缓冲区的指针
//因此tasksEvents 指向该任务数组
tasksEvents = (uint16 *)osal_mem_alloc( sizeof( uint16 ) * tasksCnt);

//把开辟的内存全部设置为0；sizeof( uint16 )为4个字节，即一个任务的长度，乘以任务数量tasksCnt，为全部内存空间
osal_memset( tasksEvents, 0, (sizeof( uint16 ) * tasksCnt));

//初始化MAC层任务，mac_taskID=0
macTaskInit( taskID++ );

//初始化网络层任务，nwk_taskID=1
nwk_init( taskID++ );

//初始化硬件任务，hal_taskID=2
Hal_Init( taskID++ );

//初始化MT层任务
//MT层：实现通过串口可控制各层，并与各层进行直接交互
#if defined( MT_TASK )
MT_TaskInit( taskID++ );
#endif

//初始化APS层任务
//应用层由三个部分组成,APS 子层,ZDO(包含 ZDO 管理平台) 和制造商定义的应用对象
//APS:提供NWK层和APL层之间的接口，又名应用支持子层
APS_Init( taskID++ );

//是否已定义分包传输
```

```
//初始化化APSF层任务
//APSF层是啥东西，还不明白？？
#if defined ( ZIGBEE_FRAGMENTATION )
APSF_Init( taskID++ );
#endif

//初始化ZDO应用层任务
ZDApp_Init( taskID++ );

//初始化网络管理任务
#if defined ( ZIGBEE_FREQ_AGILITY ) || defined ( ZIGBEE_PANID_CONFLICT )
ZDNwkMgr_Init( taskID++ );
#endif

//自定义任务初始化
SampleApp_Init( taskID );
}
```

说明：

1、任务初始化，就是为系统的各个任务分配存储空间,当然,这个空间初始化时为全0(NULL),然后为各任务分配taskID;这里的顺序要注意.

系统主循环函数里tasksEvents[idx]和tasksArr[idx]的idx与这里taskID是一一对应关系。

2、指针数组tasksEvents[ ]里面最终分别指向的是各任务存储空间

指针数组tasksArr[ ]里面最终分别指向的是各任务事件处理函数

这两个指针数组里面各元素的顺序要一一对应,因为后面需要相应任务调用相应事件处理函数.