

参考如下:

文档参考: [TI 实时操作系统SYS/BIOS使用总结](#)

```
/* BIOS/XDC Include Files. */
#include <xdc/std.h>
#include <xdc/cfg/global.h>
#include <xdc/runtime/IHeap.h>
#include <xdc/runtime/System.h>
#include <xdc/runtime/Error.h>
#include <xdc/runtime/Memory.h>
#include <ti/sysbios/BIOS.h>
#include <ti/sysbios/knl/Task.h>
#include <ti/sysbios/knl/Event.h>
#include <ti/sysbios/knl/Semaphore.h>
#include <ti/sysbios/knl/Clock.h>
#include <ti/sysbios/heaps/HeapBuf.h>
#include <ti/sysbios/heaps/HeapMem.h>
#include <ti/sysbios/knl/Event.h>
#include <ti/sysbios/family/c64p/Cache.h>
#include <ti/sysbios/family/c64p/Hwi.h>
#include <ti/sysbios/family/c64p/EventCombiner.h>
#include <ti/sysbios/utils/Load.h>
```

1.概述:

SYS/BIOS 是一个可扩展的实时的操作系统。具有非常快速的响应时间（在中断和任务切换时达到较短的延迟），响应时间的确定性，强壮的抢占系统，优化的内存分配和堆栈管理（尽量少的消耗和碎片）。能够实现系统的模块化并可裁剪。

最大的特点就是：实时。

2: XDCtools:

XDCtools 是SYS/BIOS引用的一种实时的软件组件技术（RTSC），它包括一些有用的工具，标准的API函数，静态配置文件和打包（packaging）操作。XDC最主要的亮点在于它标准化了传递过程，并简化了在应用程序中引用目标平台下其他包的过程。在工程创建的时候需要先创建一个基于RTST的平台，用来分配和管理内存和缓存，它可以自动的生成对应的cmd文件。

4: 四种线程

线程是一组在存储在存储器中的代码，一旦寄存器被正确初始化，CPU就会执行这些代码。线程可以是任何类型的，SYS/BIOS定义了四种线程类型：

硬件中断 (HWI) ,软件中断 (SWI) ,任务 (TASK) , 空闲 (Idle) 。

硬件中断：时间要求最为严格，要就实时响应的任务：一般为外设和内部的实时时间和定时器模块（定时器模块是通过硬件中断实现的，使用硬件的定时器）。

软件中断：对时间要求较为低的一种中断线程。（时钟模块是基于软件中断模块的定时器模块）

任务：CPU执行的主要线程，和软件中断的区别，可以被阻塞（可以等待）

空闲：优先级最低

各线程有隐式和显示的优先级：

隐式优先级是由线程的类型决定的，显示优先级由软件的编程者决定，并且允许抢占的：就是高优先级的线程能够抢线程占低优先级的线程，优先执行，这种高于优先级的调度管理机制保证了最高优先级的线程能在第一时间被执行。