基本概念和应用：

是一个随身携带的计算设备，典型的移动设备有一个带有触控的屏幕或者迷你键盘。



应用举例：Moblin->Meego；Symbian；Windows CE；Android；……

4种嵌入式体系结构：

* 轮转结构(round-robin)
  + - * 最简单的一种结构
      * 主循环依次检查每个I/O设备，并为需要服务的设备提供服务。
      * 不存在中断
      * 优点

结构简单

* + - * 缺点

最坏响应时间

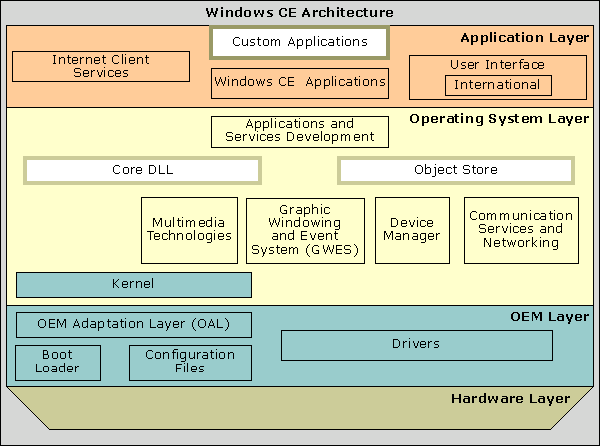
缺乏优先级

结构缺乏可扩展性

* 带中断的轮转结构
  + 中断程序处理硬件特别紧急的需求，然后设置标记；
  + 主循环轮询这些标记，然后根据这些需求进行后续的处理。
  + 优点
    - 为硬件操作提供优先级。实质上，中断程序中的所有操作拥有比主程序中任务代码更高的优先级。
  + 缺点
    - 结构带来一定的复杂度
    - 共享数据问题
    - 优先级机制不彻底！所有任务代码以同样的优先级来执行。
      * 改进方法？
    - 任务的最坏响应时间：
* 函数队列调度结构
  + 更复杂精细的结构
  + 中断程序在一个函数指针队列中添加一个函数指针，以供主程序调用。
  + 主程序仅需要从该队列中读取相应的指针并且调用相关函数。
  + 引进任务优先级：通过对函数指针排队实现

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 结构 | 允许优先级 | 任务代码的最坏响应时间 | 代码改变时响应时间的稳定性 | 简单性 |
| 轮转结构 | 不允许 | 所有任务代码的总和 | 差 | 很简单 |
| 带中断的轮转结构 | 中断程序有优先级次序，所有任务代码在同一个优先级上 | 所有任务代码的执行时间的总和(加上中断程序的执行时间) | 中断程序响应时间的稳定性好；任务代码响应时间的稳定性差 | 必须处理中断程序和任务代码的共享数据 |
| 函数队列调度结构 | 中断程序有优先级次序，任务代码也有优先级次序 | 最长函数的执行时间(加上中断程序的执行时间) | 相对较好 | 必须处理共享数据，并且要编写函数排队代码 |
| 实时操作系统结构 | 中断程序有优先级次序，任务代码也有优先级次序 | 0(加上中断程序的执行时间)(对优先级最高的任务来说) | 很好 | 最复杂（尽管多数复杂部分是在操作系统内部） |

WinCE体系结构：



KITL-Kernel Independent Transport Layer，用于调试的时候的通信，是在开发ＰＣ和Windows Embedded CE 启动的设备之间。提供remote tools

特点：

是一个强实时性的OS，并且不是从桌面Windows移植过来，而是从头开始设计，专门为嵌入式设备设计。

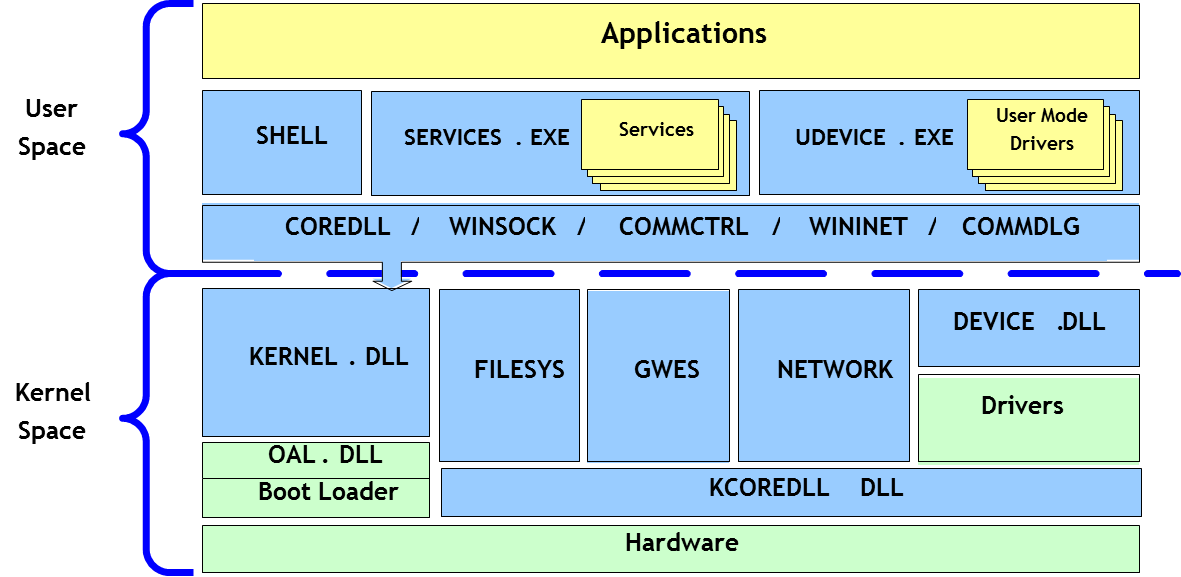
主要是为32位处理器设计，使用4G的虚拟内存存储；2G虚拟内存是给内核的地址空间，2G是给用户内存地址空间；最多可以打开32k个processes。

两种权限：用户态和内核态

Shell标准或者自制接口给设备

Service.exe管理n个服务

UDevice.exe管理n个用户态设备驱动



内核大小只有400K到40M，得益于OS生成的时候的特性

将5.0的exe转化到6.0的dll主要是为了提高系统的性能

使用的是桌面win API的子集，因此不是所有的程序都能在模拟器上跑，但是所有模拟器的程序在桌面os上

Windows CE 6.0使用的是优先级为基础的时间片分割法进行线程调度的（256）

微内核特点：

只提供基本的系统服务，比如进程调度，进程管理，系统IO管理，存储管理，其他服务，比如说文件管理，网络支持等都是通过接口连到微内核；

具有很好的拓展性，减少磁盘和存储器的要求

厂家可以很容易的将微内核移植到其他处理器的平台上

Windows CE Components：

* Board Support Package
* Core OS Services
* Multimedia
* Connection Services & Networking
* Internationalization
* WinCE应用程序开发与Windows桌面应用程序开发的主要不同：
  + 在开发桌面应用程序时，应用程序同时在桌面操作系统中运行;
  + 在开发WinCE应用程序时，需要将WinCE应用程序download到WinCE目标设备上运行，并且调试的情况也是一样。
* 2GB user space, 2GB kernel space
* CE 5.0
  + 32 processes; 32MB each process.
* CE 6.0
  + 32000 processes, 2GB each process.
* XIP：Execute In Place（本地执行）
  + 允许程序代码不进入RAM，直接在ROM/Flash中执行
* 优点：
  + 代码段不必先加载到物理内存中，节省内存。（在Windows CE中，OS分配虚拟地址空间给代码段，并将其映射到ROM/Flash中）
* 缺点：
  + 只支持允许线性访问的ROM/Flash（e.g. NOR Flash）
  + 执行速度相对较慢；不适合实时性要求高的场合。
* 系统定制与开发的整体流程
* Build System – 4 steps
  + 与Build有关的文件：Source，DIRS，Makefile

与系统初始化有关的文件：.bib, .reg, .dat, .db

BSP：Board Support Package，板级支持包

Bootloader，Drivers，Configuration files（定义了驱动的一些参数），OEM（OAL）

进程和线程的概念：

进程是动态的程序创建执行结束的完整生命周期

包含两个东西：一个是内核对象

另一个是地址空间；

线程是winCE执行调度的最小单元

进程之间的同步通信采用了互斥量（Mutex），事件（Event），信号量（Semaphore）：Signaled 或者Non-signaled。