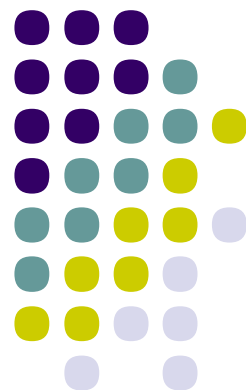


# 嵌入式操作系统基础

电子电气工程学院  
计算机系  
王益涵





# 主要内容

- 1 操作系统概述
- 2 常用的操作系统术语
- 3 嵌入式操作系统概述
- 4 常见的嵌入式操作系统



# 嵌入式操作系统基础

- 1 操作系统概述
  - 定义
  - 功能
  - 特征
  - 模式



# 嵌入式操作系统基础

- 1 操作系统概述

- 定义

从广义的角度（作为程序），操作系统是一直在计算机上运行的**程序**，是计算机最基本的程序。

从用户的角度（作为虚拟机），操作系统是用户与计算机交互的**接口**。

从系统的角度（作为资源管理器），操作系统是计算机硬件和软件资源的**管理者**。



# 嵌入式操作系统基础

## ● 1 操作系统概述

### ● 功能

#### ● 处理机管理

- 进程控制
- 进程调度
- 进程通信
- 进程同步

程序与进程(程序的一次执行):

——**程序**: 静态, 只包含代码

——**进程**: 动态, 至少包含代码, 数据及上下文环境(进程的状态)

1. 进程是资源(CPU, 内存, 文件和外设...)分配的基本单位
2. 进程使程序(如I/O读写和计算程序...)能独立并发运行



# 嵌入式操作系统基础

- 1 操作系统概述
  - 功能
    - 存储器管理：
      - 内存分配
        - 静态分配
        - 动态分配
      - 内存保护
        - 上界界限寄存器
        - 下界界限寄存器



# 嵌入式操作系统基础

## ● 1 操作系统概述

### ● 功能

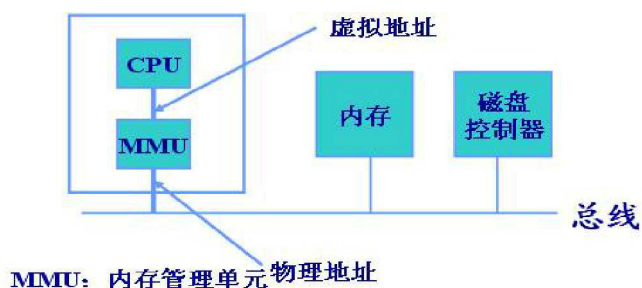
#### ● 存储器管理：

##### ▪ 地址映射

——逻辑地址：程序中的其他地址是相对编译后的起始地址计算

——物理地址：内存中限定的地址

##### ▪ 虚拟内存





# 嵌入式操作系统基础

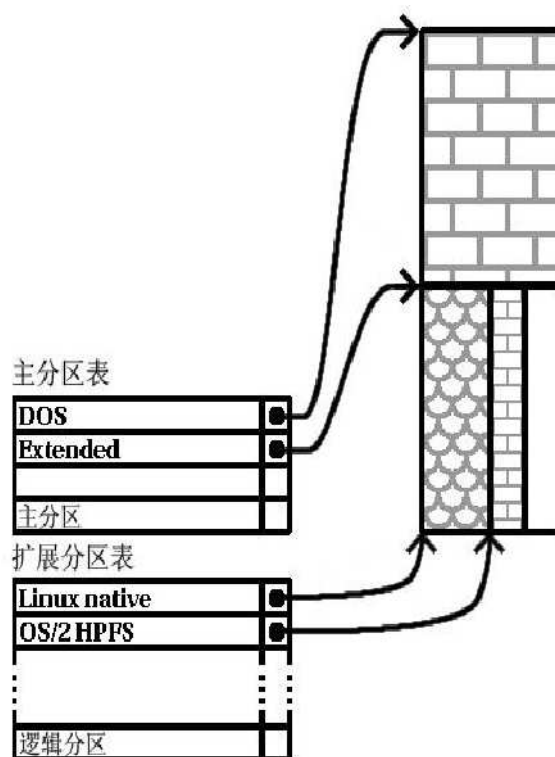
- 1 操作系统概述
  - 功能
    - 文件管理：
      - 存储结构和空间管理
      - 目录管理
      - 读写管理
      - 权限管理





# 文件管理

- 硬盘分区
  - 往硬盘哪个区域写信息
  - 主分区[1...4]
  - 扩展分区
    - 逻辑分区1[5]、逻辑分区2[6]...
- 文件系统
  - 按哪种格式写信息
  - EXT2/3, SWAP(分区非文件)
  - FAT16/32, NTFS





# 嵌入式操作系统基础

- 1 操作系统概述
  - 功能
    - 设备管理：
      - 设备分配
      - 设备处理（设备驱动）
      - 缓冲管理
      - 虚拟设备



# 嵌入式操作系统基础

## ● 1 操作系统概述

### ● 特征

#### ● 并发

引入进程使程序能够并发执行（如I/O程序和计算程序）

——并发：两个或多个事件在同一时间间隔内发生

——并行：两个或多个事件在同一时刻发生

#### ● 共享

系统中的资源可供内存中多个并发进程共同使用

——同时共享：磁盘

——互斥共享：打印机（临界资源）



# 嵌入式操作系统基础

- 1 操作系统概述

- 特征

- 并发与共享之间的关系

**并发**和**共享**是操作系统两个最基本的特征，且互为条件

——资源共享以程序的并发执行为条件（独占）

——程序并发执行以对资源共享有效管理为条件（死锁）



# 嵌入式操作系统基础

## ● 1 操作系统概述

### ● 特征

#### ● 虚拟

把一个物理实体映射为一个或多个逻辑实体

#### ● 异步

进程并非一气呵成而是走走停停（过程不确定结果确定）

- 程序执行顺序的不确定性：两个进程可能先入内存后完成
- 程序执行时间的不确定性：同一程序两次运行时间不同
- 程序执行结果的确定性：同一程序多次运行结果相同



# 嵌入式操作系统基础

## ● 1 操作系统概述

### ● 模式：

#### ● 用户模式

用户程序的运行模式

不包含处理器的特权指令

**特权指令：  
影响处理器状态的指令**

#### ● 系统模式

操作系统内核的运行模式

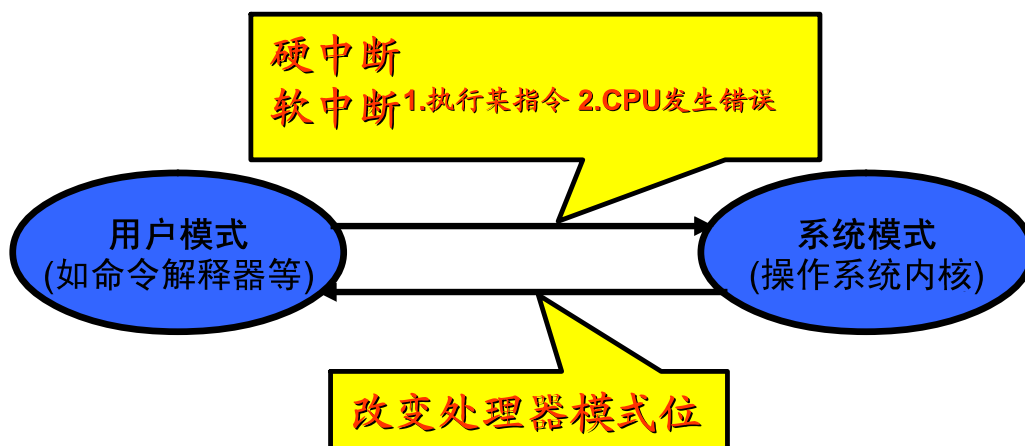
包含处理器指令系统全集

**系统复位时处于的状态**



# 嵌入式操作系统基础

- 1 操作系统概述
  - 模式





# 嵌入式操作系统基础

- 1 操作系统概述
  - 定义<sup>3</sup>
  - 功能<sup>4</sup>
  - 特征<sup>4</sup>
  - 模式<sup>2</sup>





# 嵌入式操作系统基础

- 2 常用的操作系统术语

- 任务相关

- 内核相关

- 中断相关

- 资源相关



# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 任务

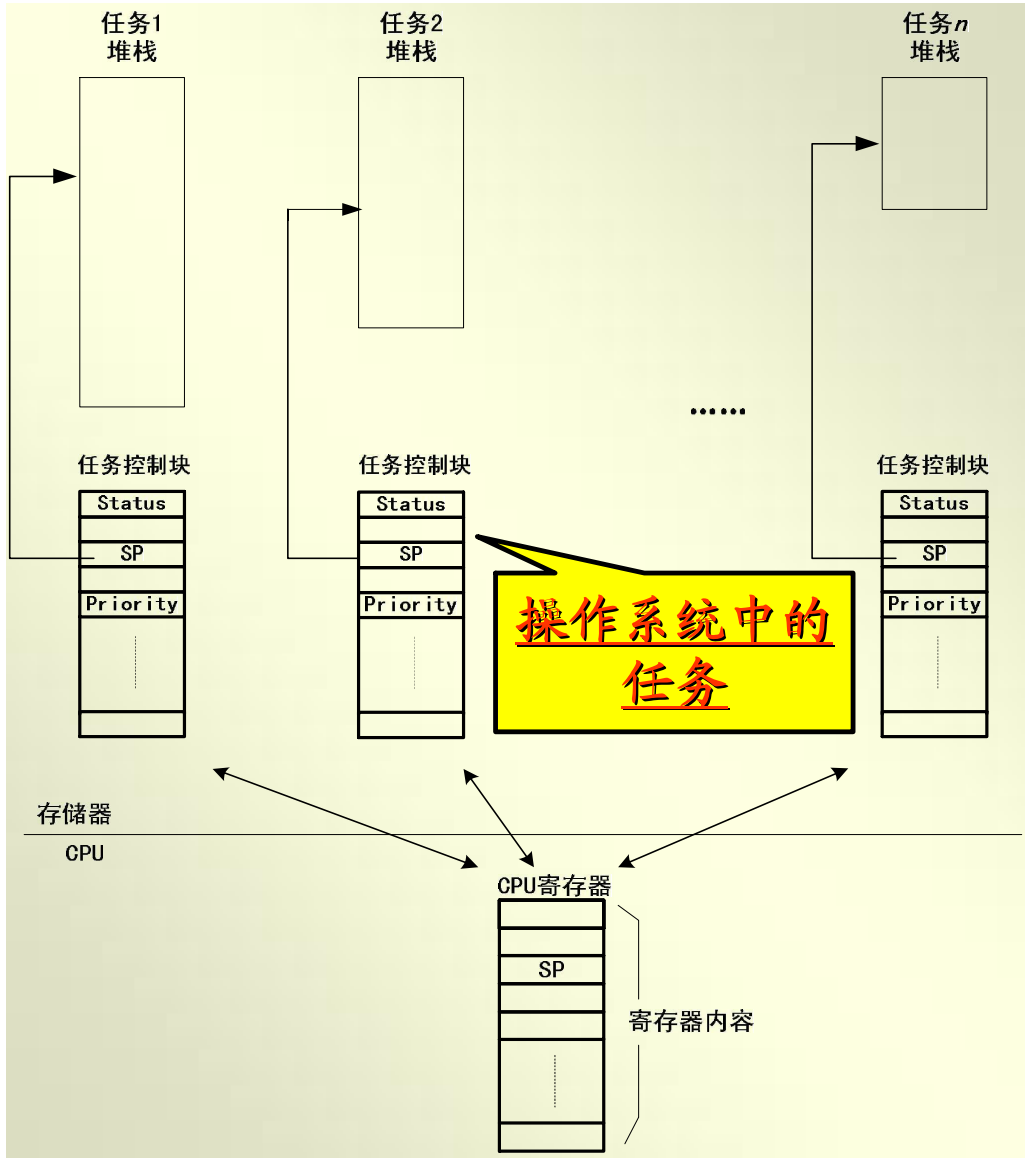
一个任务，也称一个线程，是一个简单的程序，该程序可以认为CPU完全属于该程序自己。实时应用程序的设计过程，包括如何把问题分割成多个任务，每个任务都是整个应用的某一部分，每个任务被赋予一定的优先级，有自己的一套CPU寄存器和栈空间。

进程与线程：

——进程：资源分配的基本单位，拥有独立的地址空间和资源

——线程：系统调度的基本单位，同一进程的线程共享进程的资源

1. 线程切换无须现场保护和资源转移，提高并发度减少开销
2. 同一进程的线程共享同一地址空间，易于实现同步和通信
3. 即使某线程阻塞或冗长仍继续执行，提高用户交互响应度
4. 每个线程运行可在不同的处理器上，利用多处理器的结构





# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 任务优先级

**任务优先级 (priority)** 是表示任务被调度的优先程度。每个任务都具有优先级。任务越重要，赋予的优先级应越高，越容易被调度而进入运行态。

任务优先级有静态优先级和动态优先级。

**静态优先级**指应用程序执行过程中任务的优先级不变。在静态优先级系统中，各任务以及它们的时间约束在程序编译时是已知的。

**动态优先级**指应用程序执行过程中任务的优先级是可变的。



# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 任务状态

每个任务可以处在以下三种状态之一：

#### 1、就绪态

任务已分配到除CPU外所有必要资源等待被分配CPU执行的状态。

#### 2、运行态

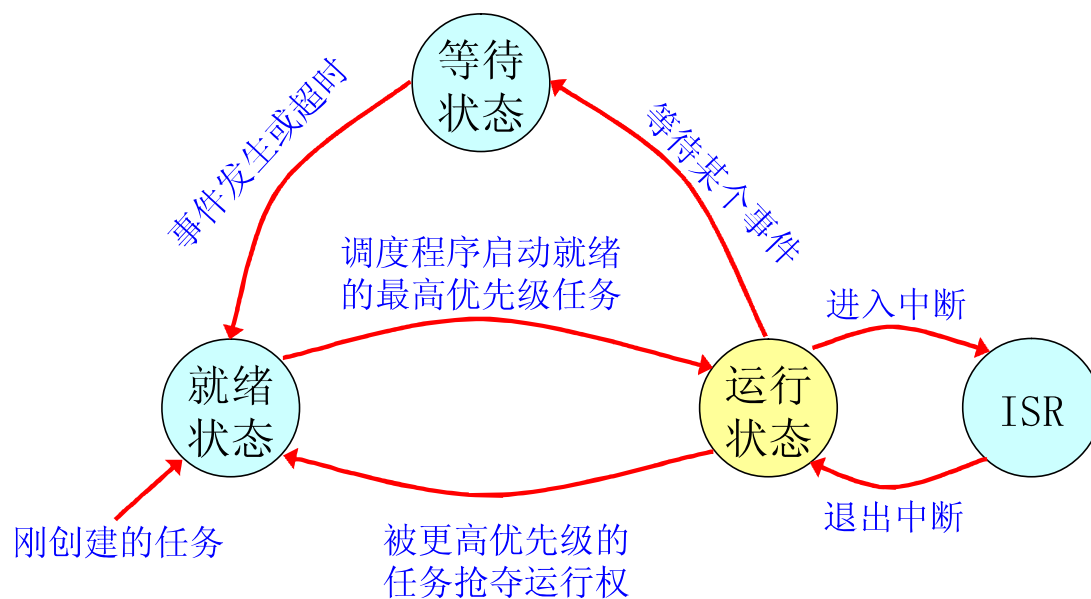
任务正在被执行的状态。

#### 3、等待态（阻塞态、睡眠态）

任务等待一定事件（如I/O请求、收到某信号等）而暂停执行的状态。



# 任务状态



- 每个任务都有确定的状态，而同时只能是一个任务占有CPU；
- 拥有运行权的任务一定是就绪任务中优先级最高的。



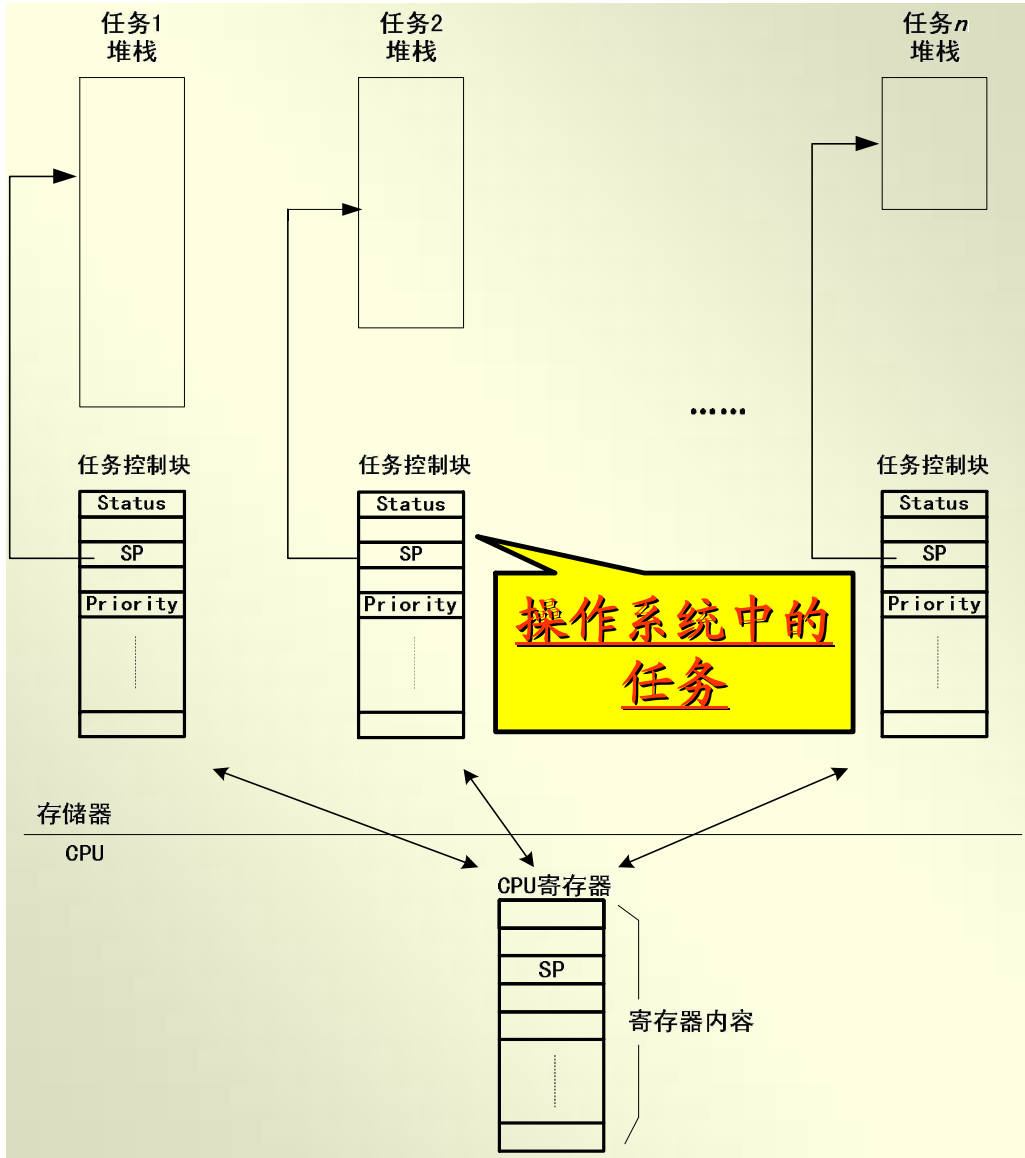
# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 任务切换

当多任务内核决定运行另外的任务时，它保存正在运行任务的当前状态，即CPU寄存器的全部内容。这些内容保存在任务的当前状态保存区，也就是任务自己的栈区之中。入栈工作完成以后，就把下一个将要运行的任务的当前状态从任务的栈中重新装入CPU寄存器，并开始下一个任务的运行。这个过程就称为任务切换（CPU寄存器 $\leftrightarrow$ 每个任务的栈区）。

这个过程增加了应用程序的额外负荷。CPU内部寄存器越多，额外负荷就越重。做任务切换所需要的时间取决于CPU有多少寄存器要入栈。







# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 任务通信

多任务系统中，任务间或中断服务与任务间常常需要交换信息，这种信息传递称为任务间的相互通信 (inter task communication)。

#### 任务通信的基本途径：

1. 共享数据结构
2. 消息机制



# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 任务通信（共享数据结构）

共享数据结构的类型有多种，可以是全局变量、指针、缓冲区等。

共享数据结构的使用必须注意排他性，即要保证每个任务或中断服务子程序独享该数据结构，可通过开/关中断，禁止任务切换和信号量机制等。



# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 任务通信（消息机制）

消息机制是指任务可以通过内核提供的系统服务向另一个任务发消息，包括消息邮箱和消息队列。

消息邮箱是通常内存空间的一个数据结构，通常是一个指针型变量。一个任务或一个中断服务子程序通过内核服务，可以把一则消息放到邮箱里；同样，一个或多个任务通过内核服务可以接收这则消息。



# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 任务通信（消息机制）

消息机制是指任务可以通过内核提供的系统服务向另一个任务发消息，包括消息邮箱和消息队列。

消息队列实际上是邮箱阵列，在消息队列中允许存放多个消息。故对消息队列的操作和对消息邮箱的操作基本相同。



# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 任务同步

**任务间的同步是指**，异步环境下的一组并发执行任务因为各自的执行结果互为对方的执行条件，因而任务间需要互发信号，使各任务按一定的速度执行。

**任务间的同步通常使用信号量**。与任务间的通信不同，信号量的使用不再作为一种互斥机制，而代表某个特定的事件是否发生。

任务间的同步 —— 各任务执行次序一定，如聊天  
任务间的异步 —— 各任务执行互不干涉，如吵架



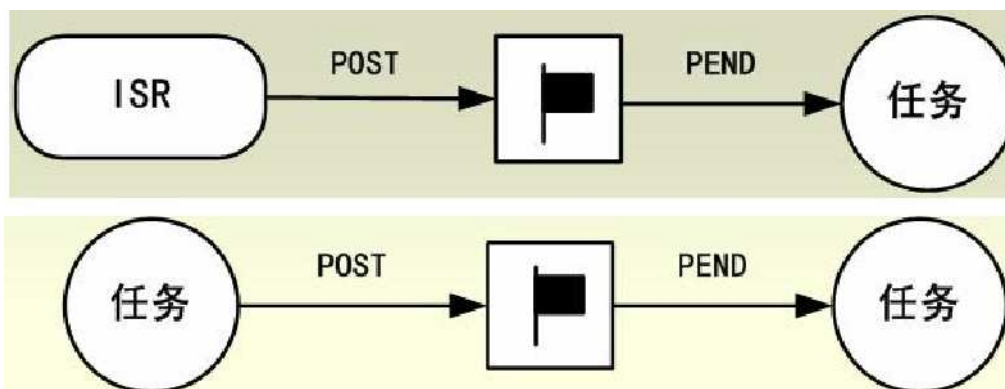
# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 任务同步

任务同步分为单向同步和多向同步。

**单向同步**多指任务与中断服务子程序之间的同步（只能是任务等待然后中断服务子程序唤醒任务），或任务与任务之间的同步。





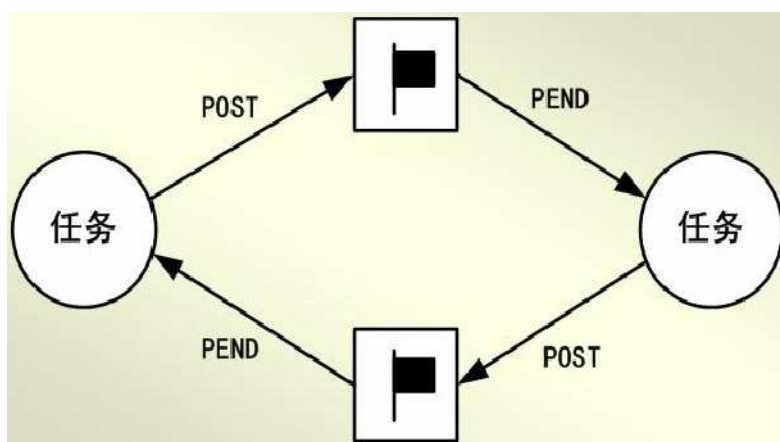
# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 任务同步

任务同步分为单向同步和多向同步。

**多向同步**多指任务与任务间的同步，通常可使用多个信号量来实现。





# 嵌入式操作系统基础

- 2 常用的操作系统术语
  - 任务相关
    - 任务<sup>2</sup>、优先级<sup>2</sup> 和状态<sup>3</sup>、切换、通信和同步
  - **内核相关**
  - 中断相关
  - 资源相关





# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

- 内核（任务管理：任务切换、任务通信...）

多任务系统中，内核负责管理各个任务，或者说为每个任务分配CPU时间，并且负责任务之间通信。

使用实时内核可以大大简化应用系统的设计，是因为实时内核允许将应用分成若干个任务，并由实时内核来管理它们。内核需要消耗一定的系统资源，比如2%~5%的CPU运行时间、RAM和ROM等。



# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 内核（任务管理：任务切换、任务通信...）

多任务系统中，内核负责管理各个任务，或者说为每个任务分配CPU时间，并且负责任务之间通信。

内核提供的基本服务是任务切换。

除了基本的任务切换服务，内核还提供必不可少的系统服务，如信号量、消息邮箱、消息队列、延时等。

如：

1. 消息邮箱的初始化
2. 将消息放入邮箱（POST）
3. 等待消息进入邮箱（PEND）
4. 从邮箱中得到消息

如：

1. 消息队列的初始化
2. 放一则消息到队列（POST）
3. 等待一则消息进入队列（PEND）
4. 从队列中得到消息



# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 内核（调度和派遣）

调度和派遣是内核的主要职责。

调度(Scheduler)指决定该轮到哪个任务运行。派遣(Dispatcher)指控制给可以执行的进程的时间片分配

多数实时内核是基于优先级调度法的。每个任务根据其重要不同被授予一定的优先级。基于优先级的调度法指CPU总是让处在就绪态的优先级最高的任务先运行。

究竟何时让最高优先级任务掌握CPU的使用权，有两种不同的情况，这要看用的是什么样的内核，非占先式的还是占先式的内核。



# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 内核（占先式内核）

当系统响应时间很重要时，要使用占先式内核。因此大多数商业上销售的实时内核都是占先式内核。

最高优先级任务一就绪，总能得到CPU控制权。

1. 当一个运行着的任务使一个比它优先级高的任务进入就绪状态，当前任务的CPU使用权就被剥夺，或者说被挂起了，高优先级的任务立刻得到了CPU的控制权。

2. 如果是中断服务子程序使一个高优先级的任务进入就绪态，中断完成后，中断了的任务被挂起，优先级高的那个任务开始运行。



# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 内核（占先式内核）

#### 占先式内核的优点：

1. 任务响应时间确定并得到优化

#### 占先式内核的缺点：

1. 使用共享数据时，需采用互斥或信号量等保护机制
2. 使用不可重入函数时，需对其进行加锁保护后再使用



# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 内核（非占先式内核）

非占先式内核要求每个任务自我放弃CPU的所有权。

非占先式调度法也称作合作型多任务，各个任务彼此合作共享一个CPU。

异步事件还是由中断服务来处理。中断服务可以使一个高优先级的任务由挂起状态变为就绪状态。但中断服务以后CPU的控制权还是回到原来被中断了的那个任务，直到该任务主动放弃CPU使用权时，那个高优先级的任务才能获得CPU的使用权。



# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 内核（非占先式内核）

#### 非占先式内核的优点：

1. 共享数据方便
2. 可以直接使用不可重入函数

#### 非占先式内核的缺点：

1. 任务响应时间不确定，商业软件中几乎不用



# 嵌入式操作系统基础

- 2 常用的操作系统术语
  - 任务相关
    - 任务<sup>2</sup>、优先级<sup>2</sup> 和状态<sup>3</sup>、切换、通信和同步
  - 内核相关
    - 服务<sup>2</sup>、调度和派遣、占先式和非占先式
  - 中断相关
  - 资源相关





# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 中断（中断服务子程序）

**中断**是一种硬件机制，用于通知CPU有**异步事件**（随机发生的事件）发生。

中断一旦被识别，CPU保存部分（或全部）上下文即部分或全部寄存器的值，跳转到专门的子程序，称为**中断服务子程序**（ISR）。

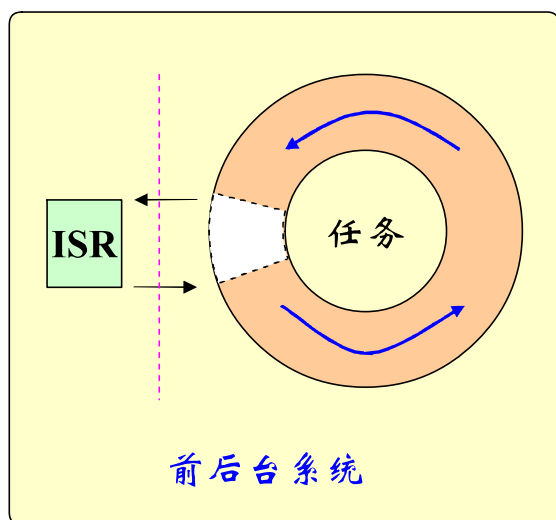
中断服务子程序做事件处理，处理完成后，程序回到：

1. 前后台系统，程序回到后台程序；
2. 占先式内核，就绪态的优先级最高的任务；
3. 非占先式内核，程序回到被中断了的任务；

**同步事件：**  
时间相关的事件

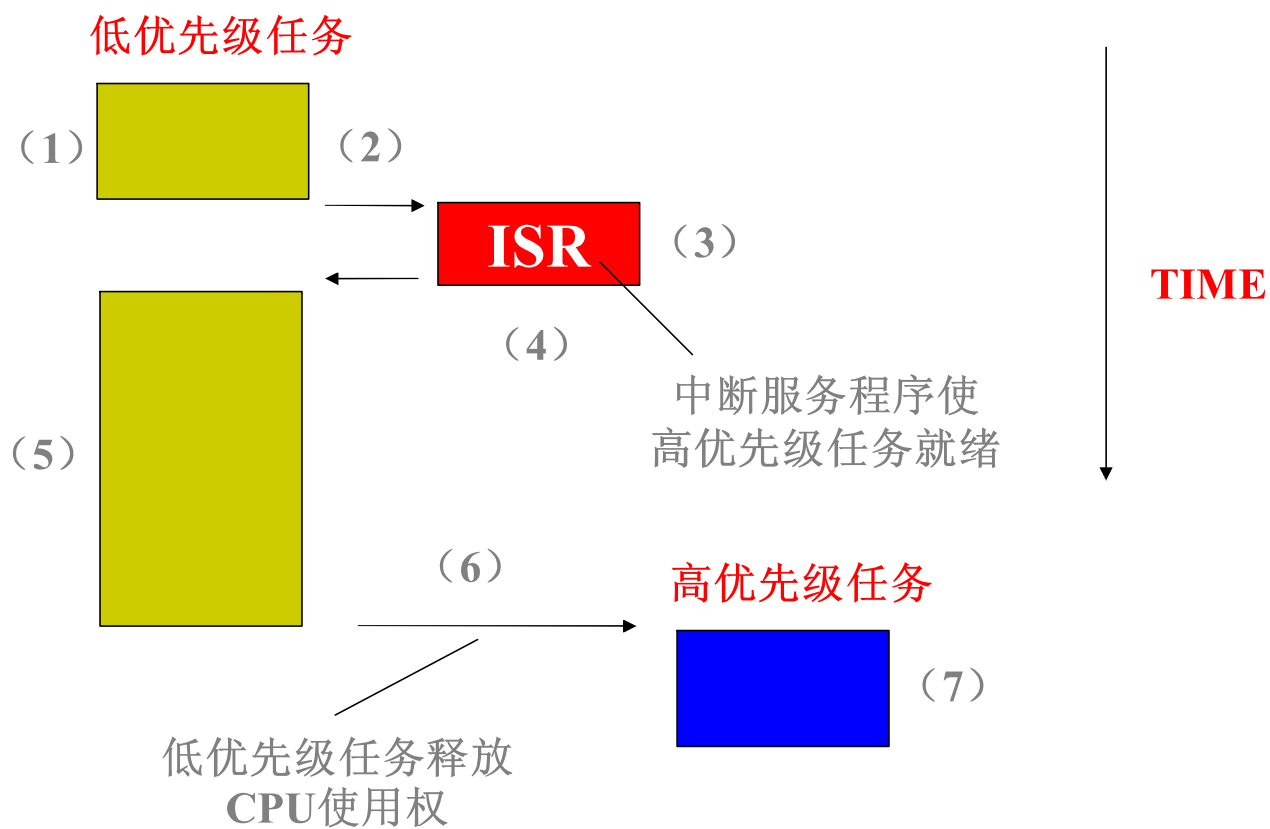


# 中断（前后台系统）



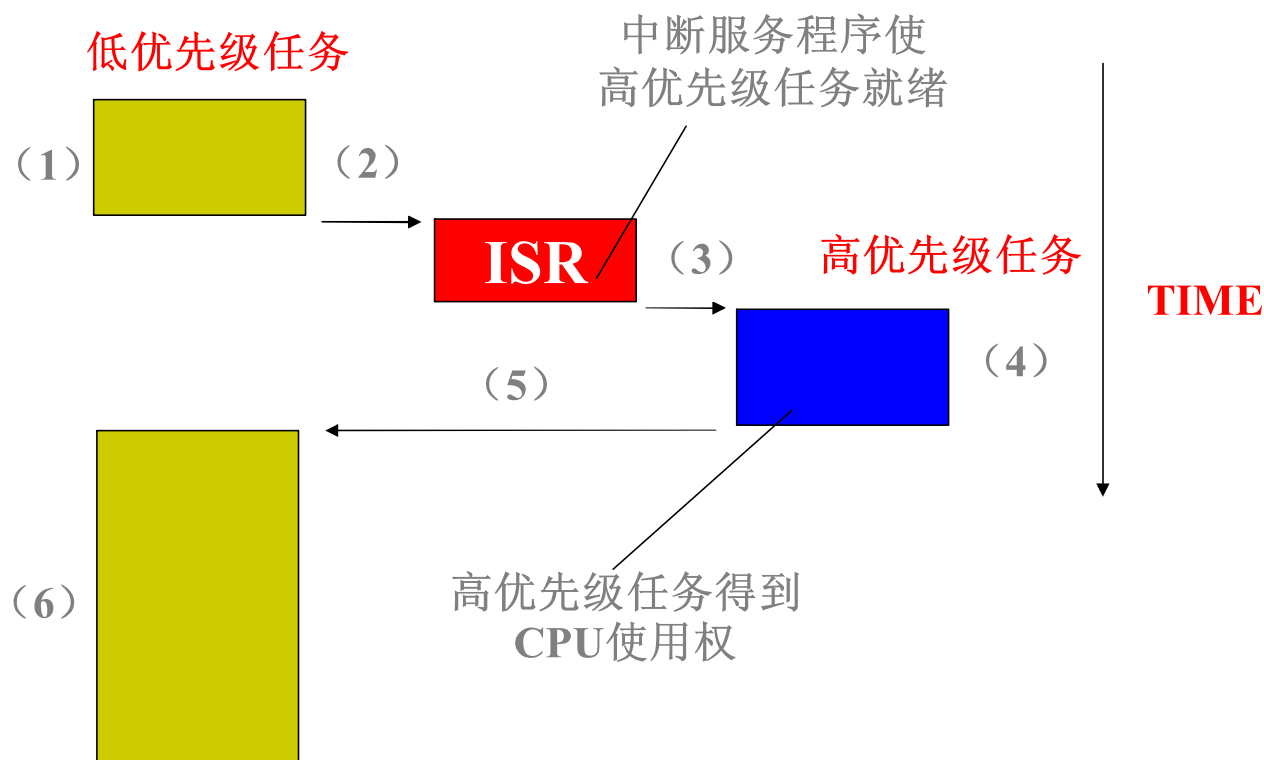


# 中断（非占先式内核）





# 中断（占先式内核）





# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 中断（时钟节拍）

**时钟节拍是特定的周期性中断**。这个中断可看作是系统心脏的脉动。中断之间的时间间隔取决于不同应用，一般在10ms到200ms之间。

时钟的节拍式中断使得内核可以将任务延时若干个整数时钟节拍，以及当任务等待事件发生时，提供等待超时的依据。

时钟节拍率越快，系统的额外开销就越大。



# 嵌入式操作系统基础

- 2 常用的操作系统术语
  - 任务相关
    - 任务<sup>2</sup>、优先级<sup>2</sup> 和状态<sup>3</sup>、切换、通信和同步
  - 内核相关
    - 服务<sup>2</sup>、调度和派遣、占先式和非占先式
  - 中断相关
    - 中断、中断服务程序、异步和同步事件、时钟
  - 资源相关



# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 可重入函数

**可重入函数**是指可以被一个以上的任务调用，而不必担心数据的破坏。

**可重入函数与中断**：可重入函数任何时候都可以被中断，一段时间以后又可以运行，而相应数据不会丢失。

**不可重入函数与占先式内核**：使用占先式内核，应用程序不应直接使用不可重入函数。如果调入不可重入函数时，低优先级任务CPU的使用权被高优先级任务剥夺，不可重入函数中的数据有可能被破坏。



# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 可重入函数

**可重入型函数**只使用局部变量，即变量保存在CPU寄存器中或堆栈中。

一个可重入型函数的例子

```
Void swap (int *x,int*y)
{
    int Temp;
    Temp=*x;
    *X=*Y;
    *y=Temp;
}
```

一个不可重入型函数的例子

```
int Temp;
Void swap (int *x,int*y)
{
    Temp=*x;
    *X=*Y;
    *y=Temp;
}
```





# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 临界区

代码的临界区也称为临界区，指处理时不可分割的代码，运行这些代码时不允许被打断。一旦这部分代码开始执行，不允许任何中断打入（这并非绝对，如果中断不调用任何包含临界区的代码也不访问任何临界区使用的共享资源，这个中断可能可以执行）。

为确保临界区代码的执行，在进入临界区之前要关中断，而临界区代码执行完成以后要立即开中断。



# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 资源、共享资源和互斥

**资源**是程序运行时可使用的软、硬件环境。

资源可以是输入输出设备，例如打印机、键盘、显示器。

资源可以是一个变量、一个结构或一个数组等。



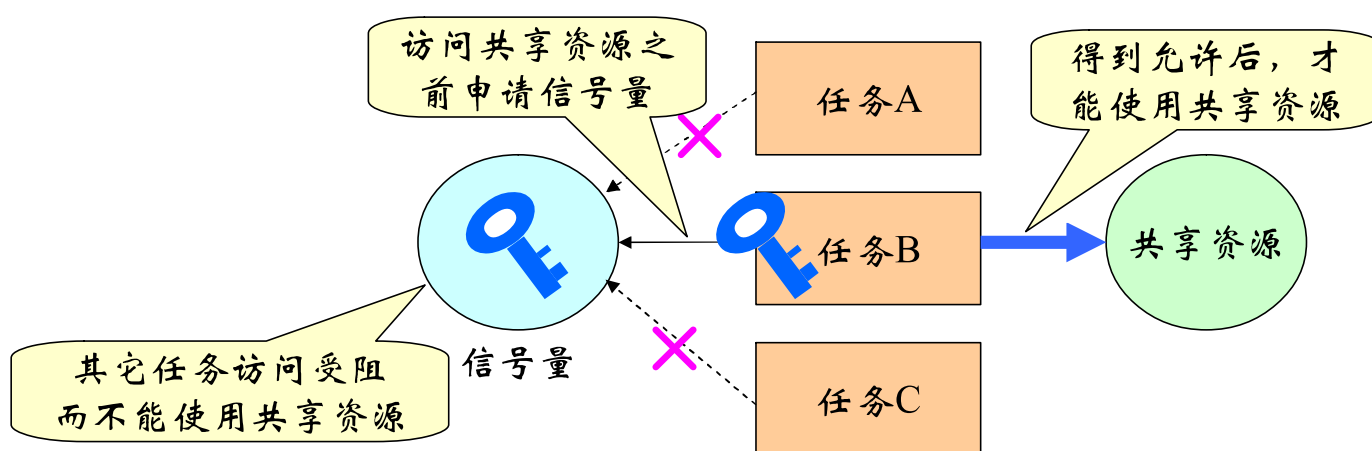
# 嵌入式操作系统基础

## ● 2 常用的操作系统术语

### ● 资源、共享资源和互斥

**共享资源**是可以被一个以上进程使用的资源。

为了防止数据被破坏，每个任务在与共享资源打交道时，必须独占该资源，这叫做**互斥**。





# 嵌入式操作系统基础

- 2 常用的操作系统术语
  - 任务相关
    - 任务<sup>2</sup>、优先级<sup>2</sup> 和状态<sup>3</sup>、切换、通信和同步
  - 内核相关
    - 服务<sup>2</sup>、调度和派遣、占先式和非占先式
  - 中断相关
    - 中断、中断服务程序、异步和同步事件、时钟
  - 资源相关
    - 资源、共享资源和互斥、临界区、可重入函数



# 嵌入式操作系统基础

- **3 嵌入式操作系统概述**
  - 嵌入式系统

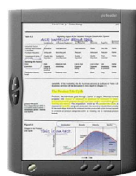


# 嵌入式操作系统基础

## ● 3 嵌入式操作系统概述

### ● 什么是嵌入式系统

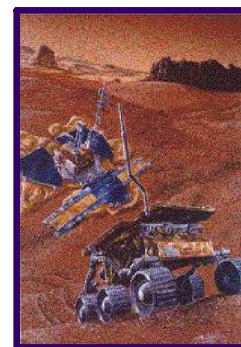
从广义的角度，嵌入式系统是一切非PC和服务器的计算机系统。



goReader  
Internet  
eBook



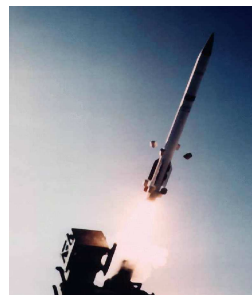
Nixvue Digital Album  
Digital Photo Album



Tektronix  
TDS7000 Digital  
Oscilloscopes



Samsung AnyWeb  
Internet Screen  
Phone





# 嵌入式操作系统基础

- 2 嵌入式操作系统概述

- 什么是嵌入式系统

从技术的角度，嵌入式系统是以应用为中心、以计算机技术为基础，软硬件可裁剪的，应用系统对功能、可靠性、成本、体积、功耗和应用环境有特殊要求的专用计算机系统。



# 嵌入式操作系统基础

## ● 2 嵌入式操作系统概述

### ● 嵌入式系统的特点

#### 嵌入式系统 ≠ 通用桌面系统

- 形态：“嵌入”于不同设备 Vs 基本雷同的形态独立存在
- 价值：“嵌入”的设备 Vs 通用指标(计算能力、存储能力)
- 功耗：几mW—几W Vs 几百W
- 功能：专用单一 Vs 通用复杂
- 资源：够用就行 Vs 大而全
- 成本：低 Vs 高





# 嵌入式操作系统基础

## ● 2 嵌入式操作系统概述

### ● 嵌入式系统的组成

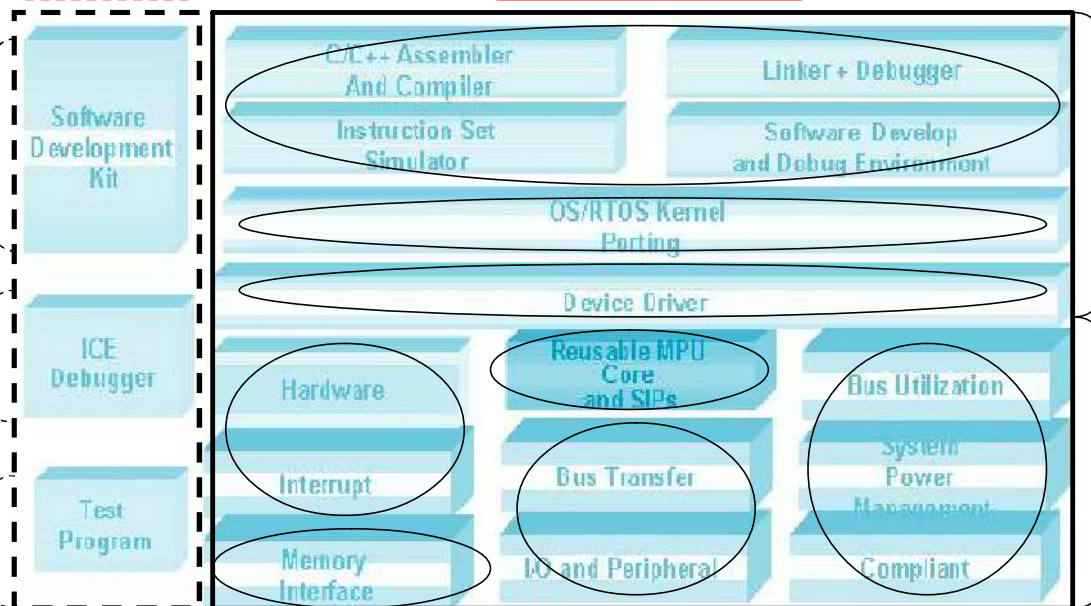
辅助工具

嵌入式系统

软件开发包

在线仿真器

测试例程



嵌入式应用软件

软件

嵌入式操作系统

驱动程序

嵌入式处理器

硬件

嵌入式外设和接口

嵌入式存储系统



# 嵌入式操作系统基础

- 2 嵌入式操作系统概述

- 什么是嵌入式操作系统

从系统的角度，嵌入式操作系统位于嵌入式硬件和嵌入式应用软件(应用程序)之间，是工作在嵌入式系统中的操作系统，通常由系统内核、设备驱动层、图形界面、文件系统等部分组成。



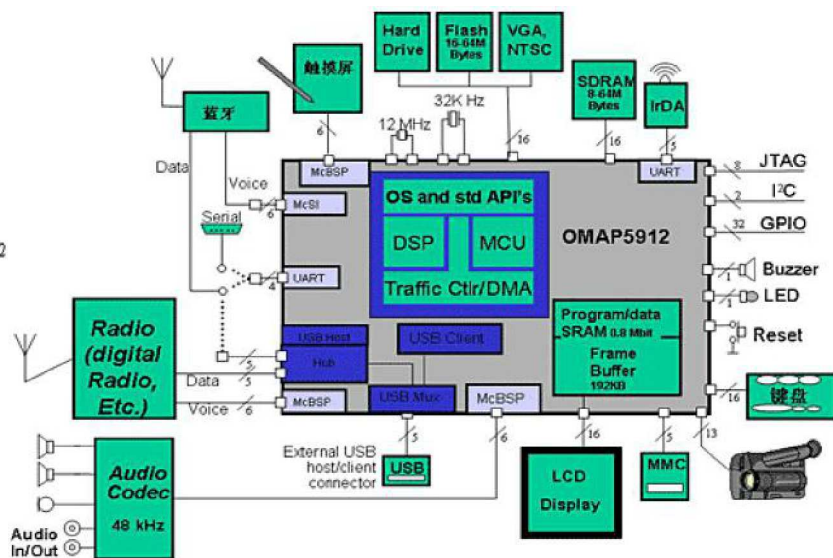
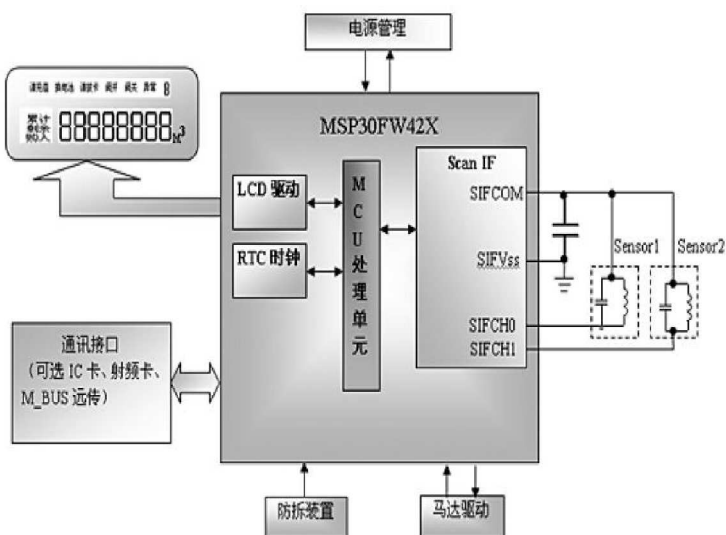
# 嵌入式操作系统基础

- 2 嵌入式操作系统概述
  - 为什么需要嵌入式操作系统
    - 实现复杂功能
      - 多任务处理
      - 网络通信
      - 数据库管理
      - 图形用户界面



# 嵌入式操作系统基础

- 2 嵌入式操作系统概述
    - 为什么需要嵌入式操作系统
- 嵌入式系统 ≠ 单片机系统





# 嵌入式操作系统基础

- 2 嵌入式操作系统概述
  - 为什么需要嵌入式操作系统
    - 提高开发效率
      - 硬件透明
      - 可移植性
      - 可靠性
      - 二次开发



# 嵌入式操作系统基础

## ● 2 嵌入式操作系统概述

- 为什么需要嵌入式操作系统

嵌入式系统  $\neq$  单片机系统

嵌入式操作系统

软硬件设计工作量比  
7: 3

软硬件设计工作量比  
1: 1



# 嵌入式操作系统基础

## ● 2 嵌入式操作系统概述

### ● 嵌入式操作系统的特点

#### ● 实时性

正确性不仅取决于计算逻辑结果，而且与计算时间相关

#### ▪ 硬实时系统

必须满足响应时限的系统（飞机控制系统）

#### ▪ 软实时系统

尽量满足响应时限的系统（视频点播系统）



# 嵌入式操作系统基础

## ● 2 嵌入式操作系统概述

### ● 嵌入式操作系统的特点

#### ● 可靠性

- 可能导致难以预测后果的原因
  - 微小机械故障
  - 某些关键任务超过时限
  - 其他任何不可靠因素
- 解决措施
  - 静态分析
  - 保留资源
  - 冗余配置





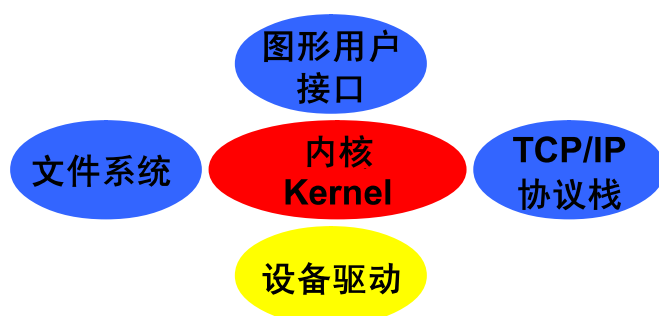
# 嵌入式操作系统基础

## ● 2 嵌入式操作系统概述

### ● 嵌入式操作系统的特点

#### ● 可裁剪性

吝啬的积木式系统





# 嵌入式操作系统基础

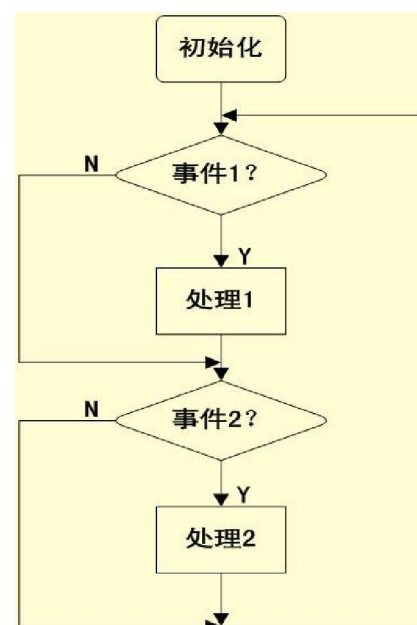
## ● 2 嵌入式操作系统概述

### ● 嵌入式操作系统的发展

#### ● 轮询系统

**循环轮询**是最简单的结构。

程序依次检查系统的每一个输入条件，一旦条件成立就进行相应的处理。





# 嵌入式操作系统基础

## ● 2 嵌入式操作系统概述

### ● 嵌入式操作系统的发展

#### ● 轮询系统

```
Initialize();
```

```
While (true)
```

```
{
```

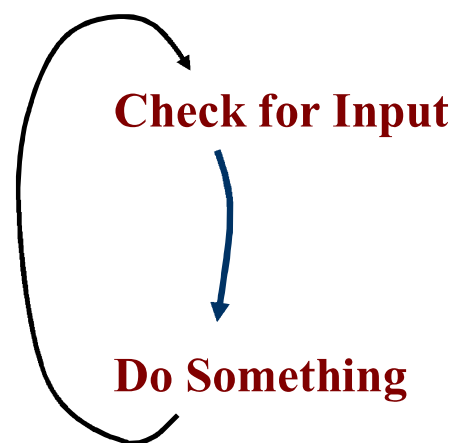
```
    if (condition_1) action_1();
```

```
    if (condition_2) action_2();
```

```
    .....
```

```
    if (condition_n) acition_n();
```

```
}
```





# 嵌入式操作系统基础

## ● 2 嵌入式操作系统概述

### ● 嵌入式操作系统的发展

#### ● 轮询系统

##### 优点:

1. 对于简单系统便于编程和理解。
2. 没有中断的机制，程序不会出现随机的问题。

##### 缺点:

1. 大的程序不便于调试。
2. 应用有限，对于大量的I/O服务的应用不易实现



# 嵌入式操作系统基础

## ● 2 嵌入式操作系统概述

### ● 嵌入式操作系统的发展

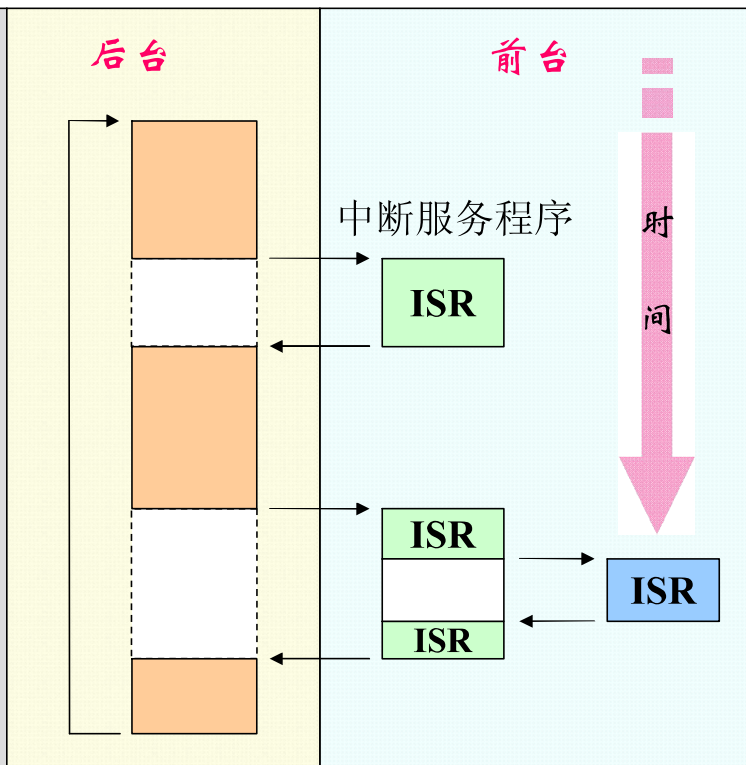
#### ● 前后台系统

前后台系统，超循环系统或中断驱动系统，能对  
外部事件作出直接响应，一般系统是一个无限的循环  
(后台，任务级)，系统对外部事件的响应或对异步  
事件的处理是通过中断服务子程序(前台，中断级)  
来完成。



# 前后台系统

循环中调用相应的函数完成相应的操作，这部分可以看成**后台行为**，后台也可以叫做任务级。这种系统在处理的及时性上比实际可以做到的要差。



中断服务程序处理异步事件，这部分可以看成**前台行为**，前台也叫中断级。时间相关性很强的关键操作一定是靠中断服务程序来保证的。



# 嵌入式操作系统基础

## ● 2 嵌入式操作系统概述

### ● 嵌入式操作系统的发展

#### ● 前后台系统

前后台系统的设计目的，是为了将时间性很强的关键操作（critical operation）通过中断服务来保证。

很多基于微处理器的产品采用前后台系统设计，例如微波炉、电话机、玩具等。在一些基于微处理器应用中，从省电的角度出发，平时微处理器处在停机状态，所有事都靠中断服务来完成。



# 嵌入式操作系统基础

## ● 2 嵌入式操作系统概述

### ● 嵌入式操作系统的发展

#### ● 前后台系统

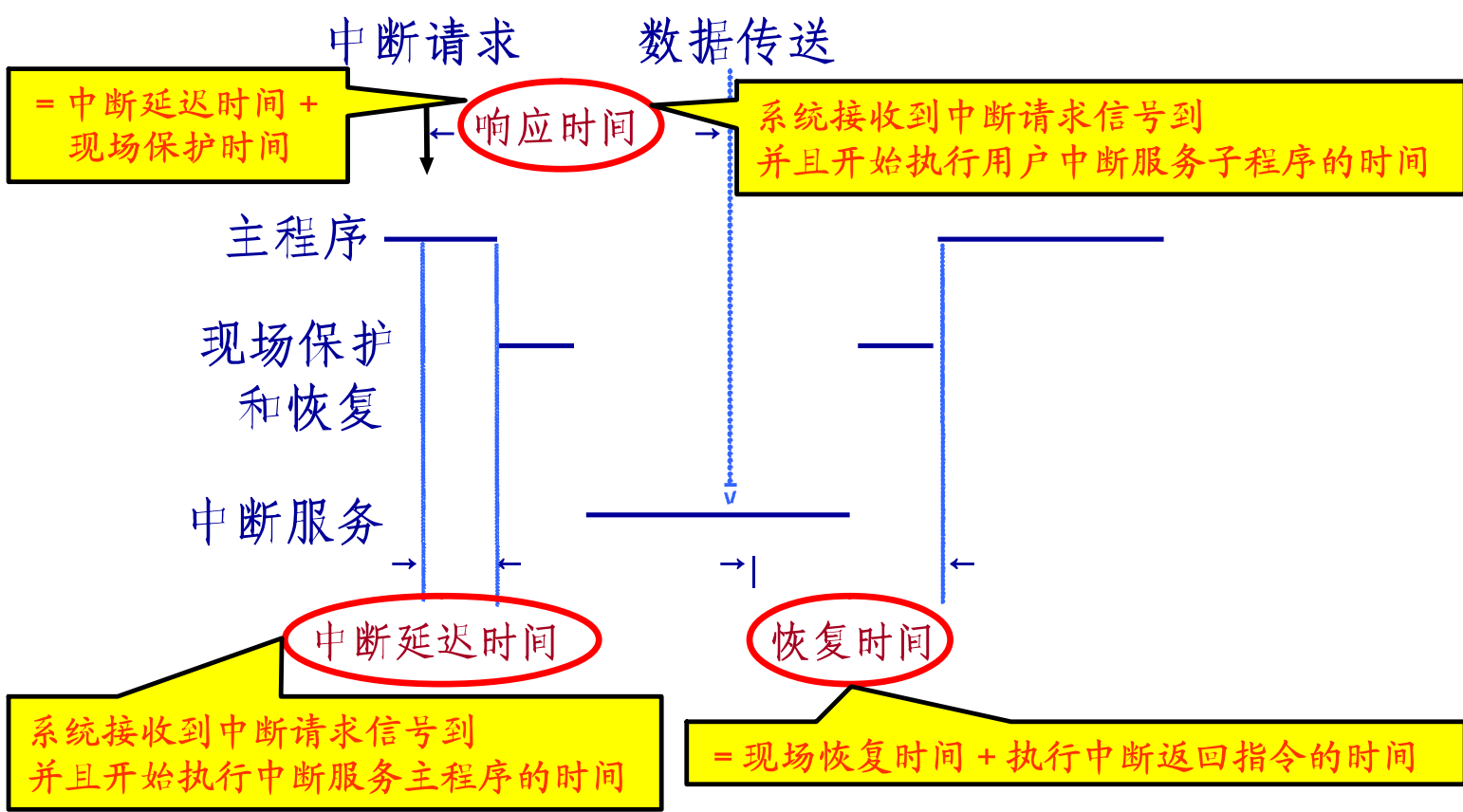
**前后台系统的设计**，考虑中断的现场保护和现场恢复、中断嵌套、中断处理过程与主程序的协调（共享资源）等问题。

**前后台系统的性能**，是由中断延迟时间(Interrupt latency time)、响应时间(response time)和恢复时间(recovery time)来描述。





# 前后台系统





# 嵌入式操作系统基础

## ● 2 嵌入式操作系统概述

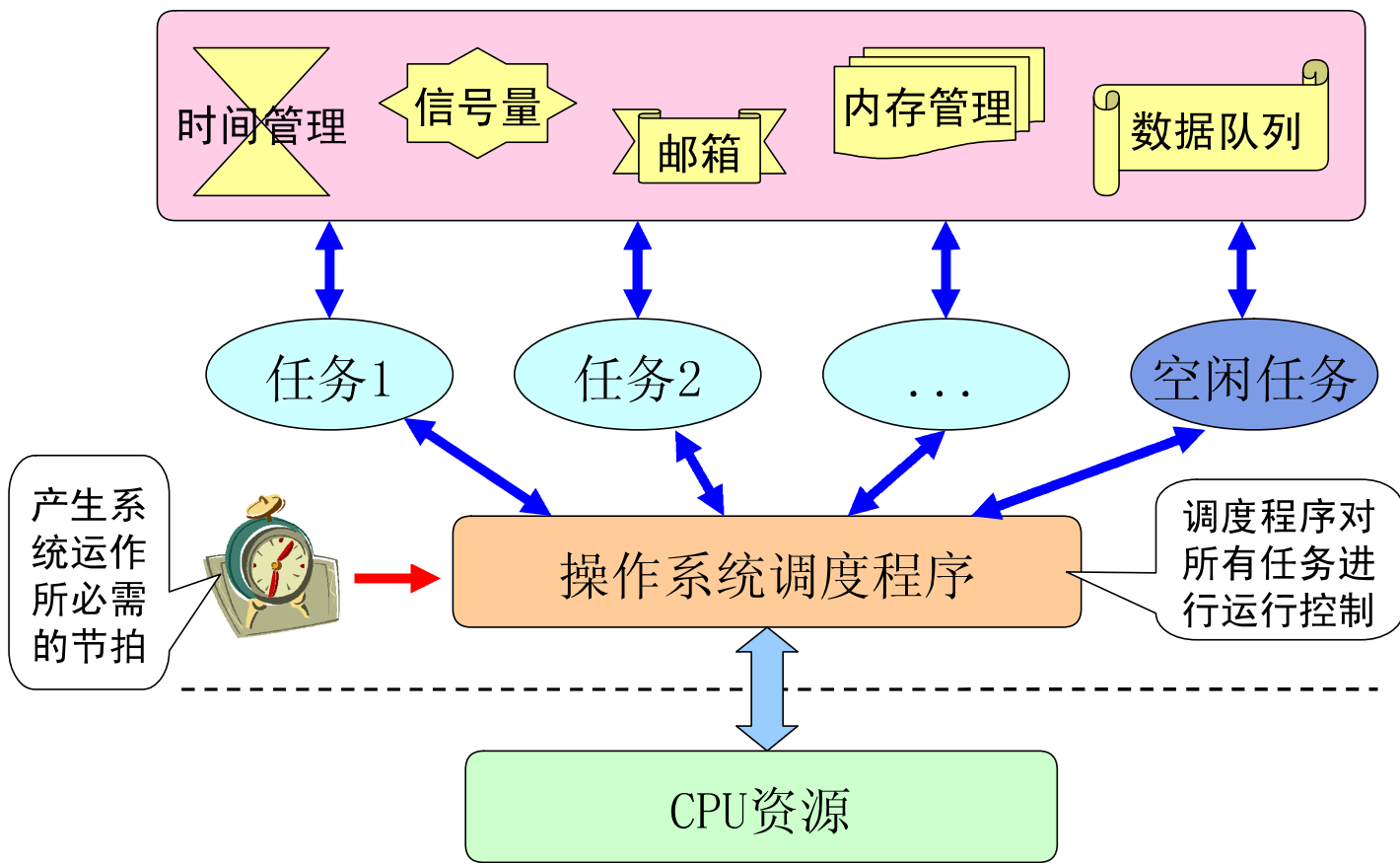
### ● 嵌入式操作系统的发展

#### ● 多任务系统

多任务系统，由多个任务、多个中断处理过程、操作系统内核组成的有机的整体，可以同时并发执行多个任务。



# 多任务系统





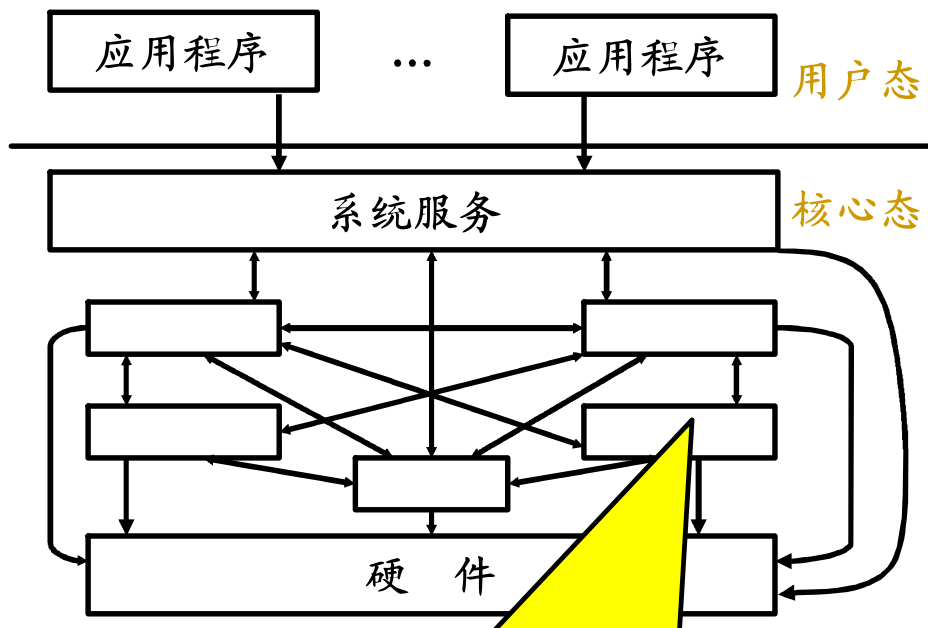
# 嵌入式操作系统基础

- 2 嵌入式操作系统概述
  - 嵌入式操作系统的结构
    - [整体结构](#)
    - [层次结构](#)
    - [微内核结构](#)

下一页



# 整体结构



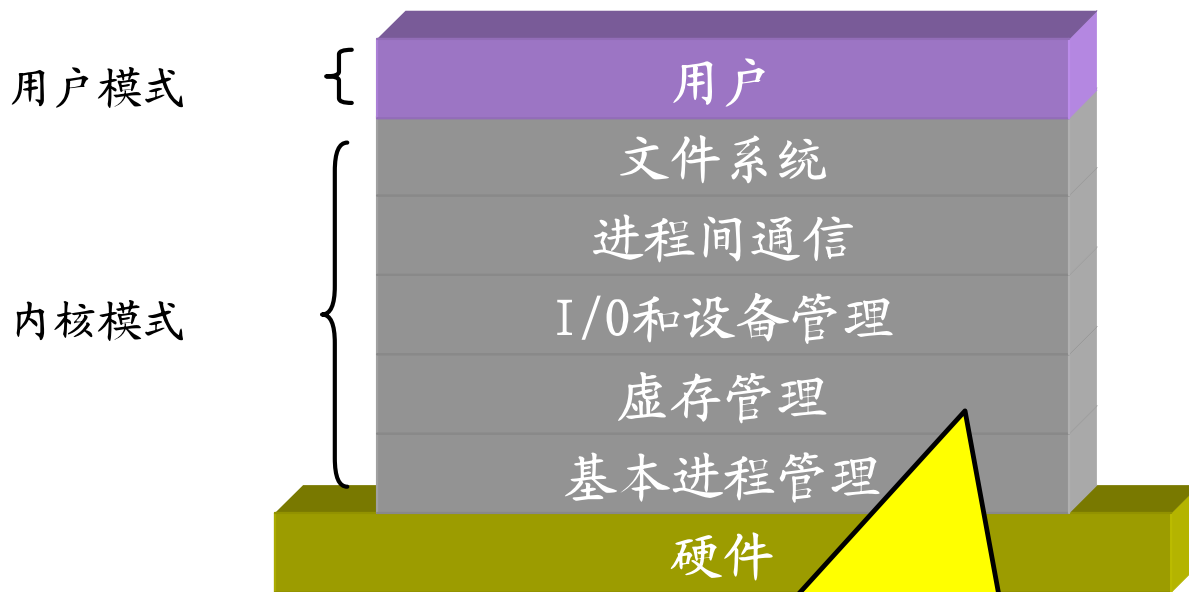
## 过程的集合

- 每个过程都有各自定义的接口
- 每个过程都可以调用其他过程

返回



# 层次结构



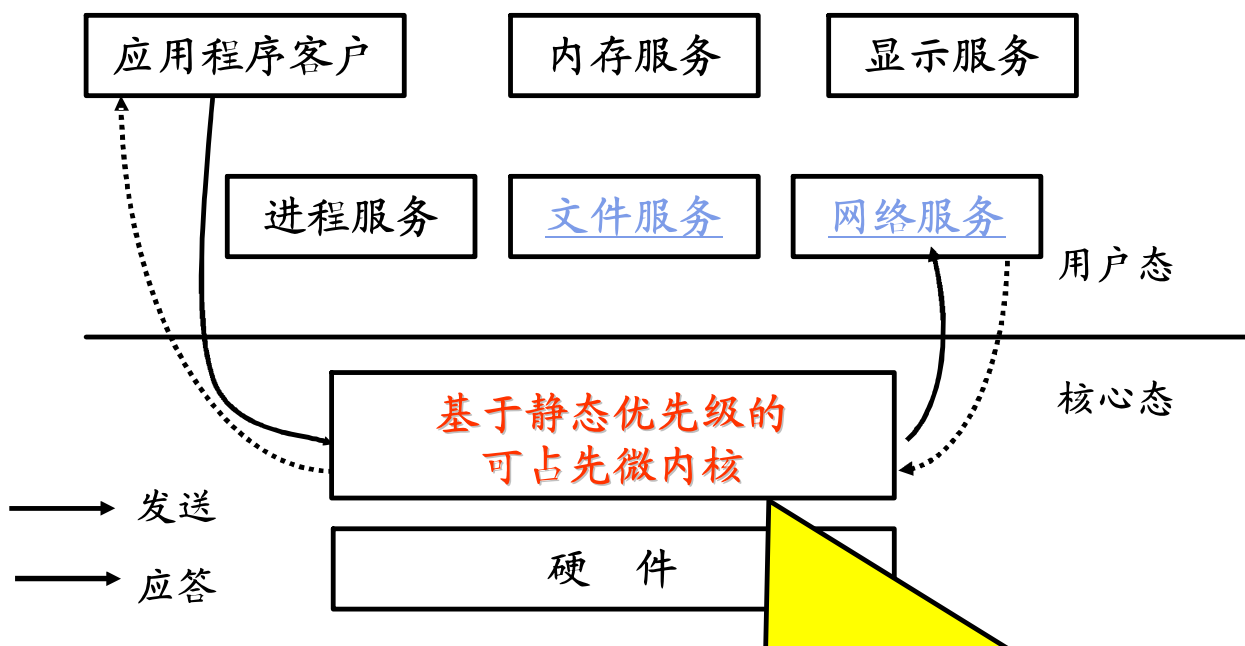
## 功能分层的叠加

- 层与层之间相互独立便于调试和修改
- 上一层只能调用相邻的下一层的过程  
相邻的下一层也只为上一层提供服务

返回



# 微内核结构



**微内核是嵌入式操作系统的基础**  
——通过消息交换处理客户机服务器间的通信  
——提供暂无统一标准的特定的应用编程接口

返回



# 微内核结构（文件服务）

- 嵌入式文件系统较为简单，主要具有文件的存储、检索、更新等功能，一般不提供保护和加密等安全机制。
- 嵌入式文件系统以系统调用和命令方式提供对文件的操作，主要有：
  - 设置和修改对文件和目录的存取权限
  - 提供建立、修改、改变、删除目录等服务
  - 提供创建、打开、读、写、关闭、撤消文件等服务





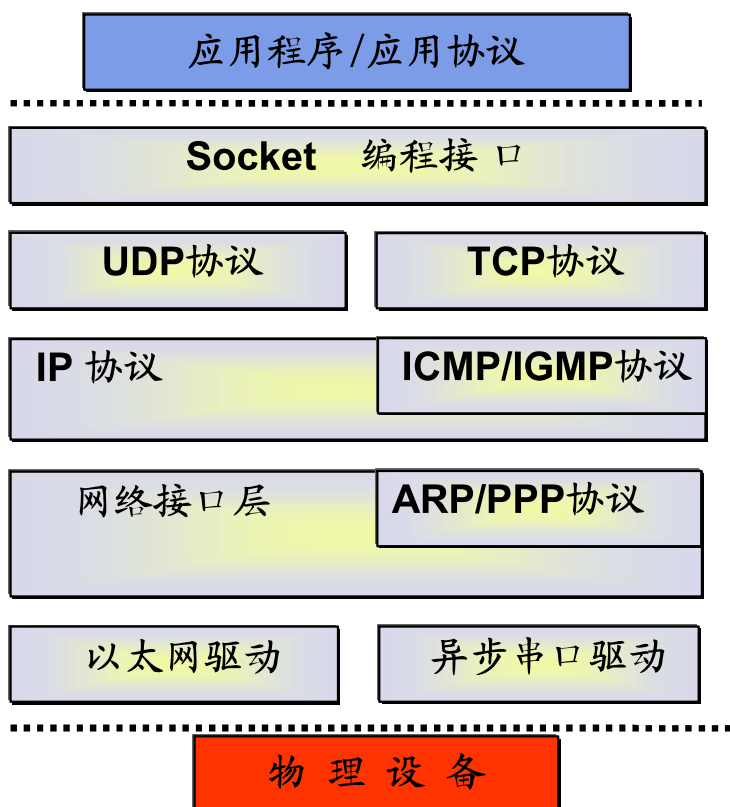
# 微内核结构（文件服务）

- 嵌入式Linux常见的文件系统：
  - ramfs: 利用VFS自身结构形成的内存文件系统，RAM中
  - cramfs: 压缩只读文件系统，RAM或ROM中
  - JFFS/JFFS2: 为Flash设计的日志文件系统 ← **作根文件系统**
  - Yaffs: 为Nand Flash设计的文件系统
  - NFS: 网络文件系统 ← **调试时作根文件系统方便调试**
  - proc: 为内核和内核模块发送信息给进程提供一种机制  
← **可以查看系统模块装载的信息**

返回



# 微内核结构（网络服务）



返回



# 嵌入式操作系统基础

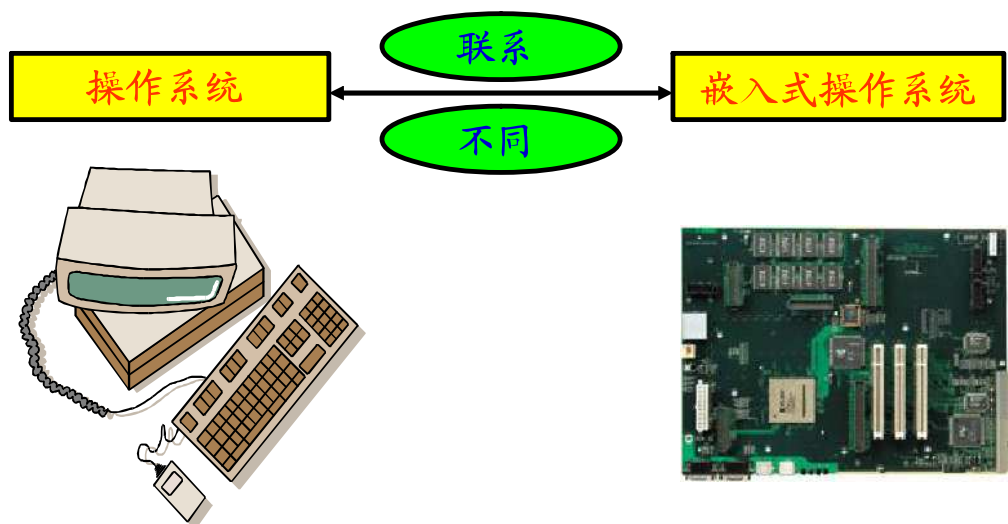
- **3 嵌入式操作系统概述**
  - 定义
  - 特点
  - 发展
  - 结构



# 嵌入式操作系统基础

## ● 3 嵌入式操作系统概述

- 留给大家的思考...





# 嵌入式操作系统基础

- 4 常见的嵌入式操作系统
  - 典型性能指标
    - 内核大小：几K—几百K
    - 调度时间片：1ms
    - 实时任务响应时间：20us—40us
    - 普通任务响应时间：20us—200ms



# 嵌入式操作系统基础

- 4 常见的嵌入式操作系统
  - 按收费模式分
    - 商用型：VxWorks, WinCE
    - 免费型：uC/OS, 嵌入式Linux
  - 按响应时间分
    - 硬实时：VxWorks
    - 软实时：uC/OS, 嵌入式Linux, WinCE



# 嵌入式操作系统基础

## ● 4 常见的嵌入式操作系统

### ● uC/OS

uC/OS, Micro Controller OS, 是个源码公开、可移植、可固化、可裁剪、占用资源少、可占先式的实时多任务操作系统。

它是由美国人Jean Labrosse于1992年使用ANSI C语言编写, 其中包含一小段汇编代码, 并于2000年通过联邦航空局商用航行器认证, 目前高校教学使用无需申请许可证。



# 嵌入式操作系统基础

## ● 4 常见的嵌入式操作系统

### ● uC/OS

uC/OS，具有执行效率高、实时性好、扩展性强和占用空间小等特点，最小内核可编译至2KB。

它适合于小型控制系统，可以方便的移植并支持多数8位、16位、32位以至64位处理器（微处理器、微控制器和数字信号处理器），应用覆盖医疗设备、音响设备、自动取款机、发动机控制、高速公路电话系统和飞行器等领域。

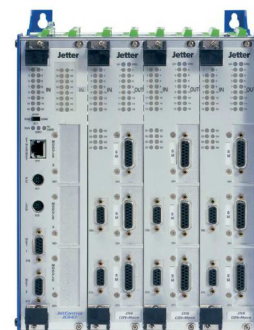
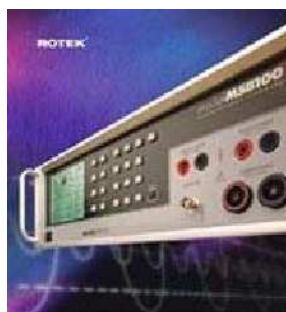




# uC/OS的应用

- 全世界有数百种uC/OS产品在应用:

- Avionics
- Medical
- Cell phones
- Routers and switches
- High-end audio equipment
- Washing machines and dryers
- UPS (Uninterruptible Power Supply)
- Industrial controllers
- GPS Navigation Systems
- Microwave Radios
- Instrumentation
- Point-of-sale terminals
- ...





# 嵌入式操作系统基础

- 4 常见的嵌入式操作系统
  - uC/OS



Chinese

ISBN 7-81077-290-2  
北京航空航天大学出版社

网站:

<http://www.ucos.com>

<http://www.micrium.com>



# 嵌入式操作系统基础

## ● 4 常见的嵌入式操作系统

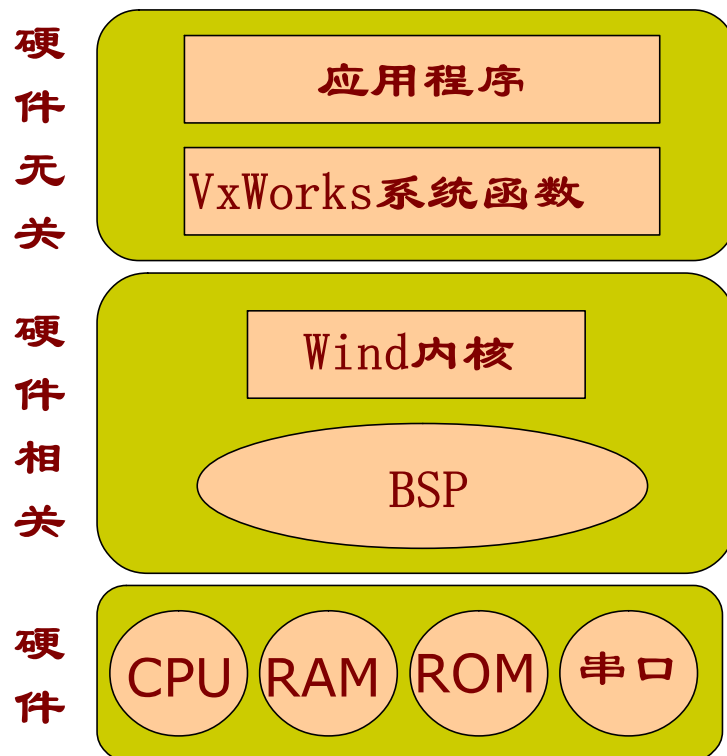
### ● VxWorks

VxWorks是美国Wind River公司于1983年开发的一种32位嵌入式实时操作系统。

VxWorks以其良好的可靠性和卓越的实时性而被广泛地应用在通信、军事、航空航天等领域。如美国F-16战斗机、B-2隐形轰炸机和爱国者导弹上，甚至连1997年4月在火星表面登陆的火星探测器上也使用VxWorks，在美国市场占有率第一名。



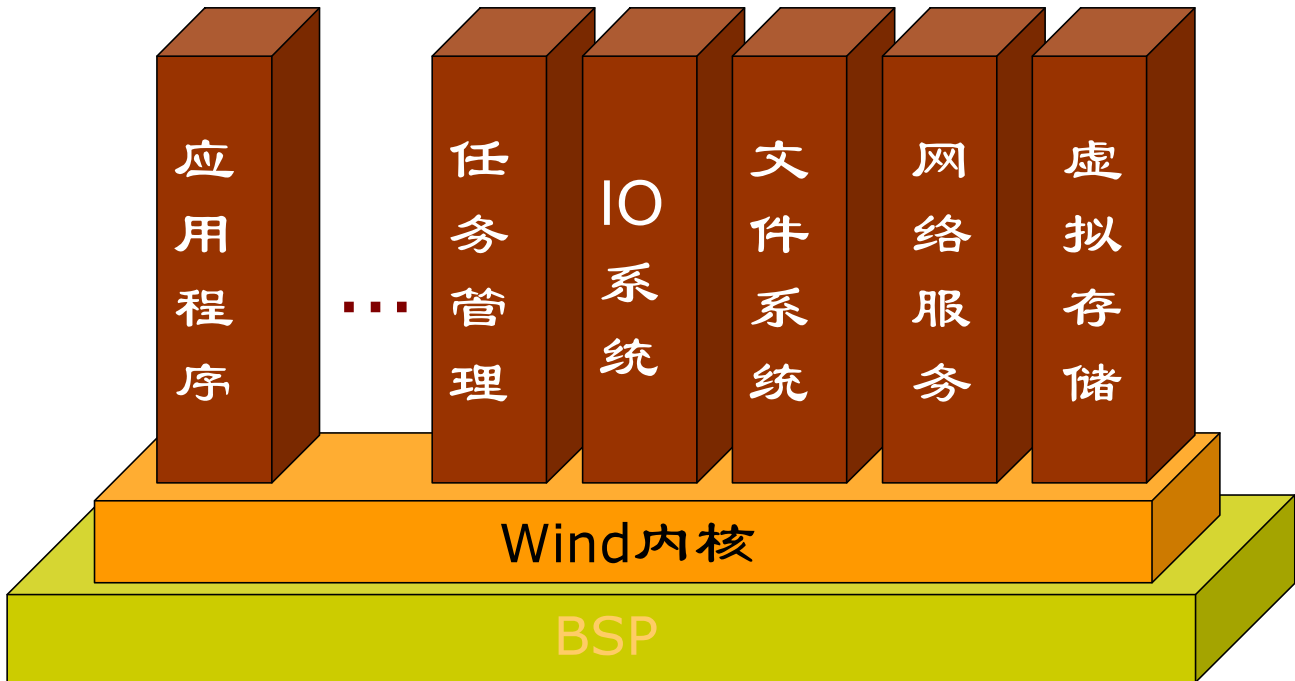
# VxWorks的架构



# VxWorks的架构



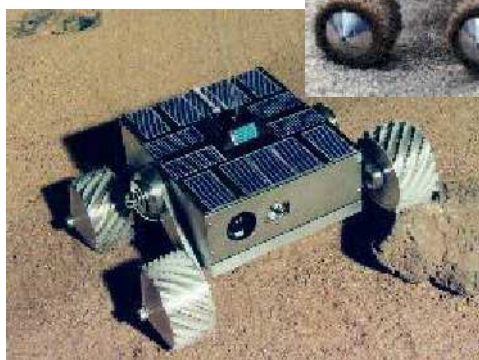
Wind内核 —— 微内核示意图





# VxWorks的应用

- 基于VxWorks的火星探路者:





# 嵌入式操作系统基础

## ● 4 常见的嵌入式操作系统

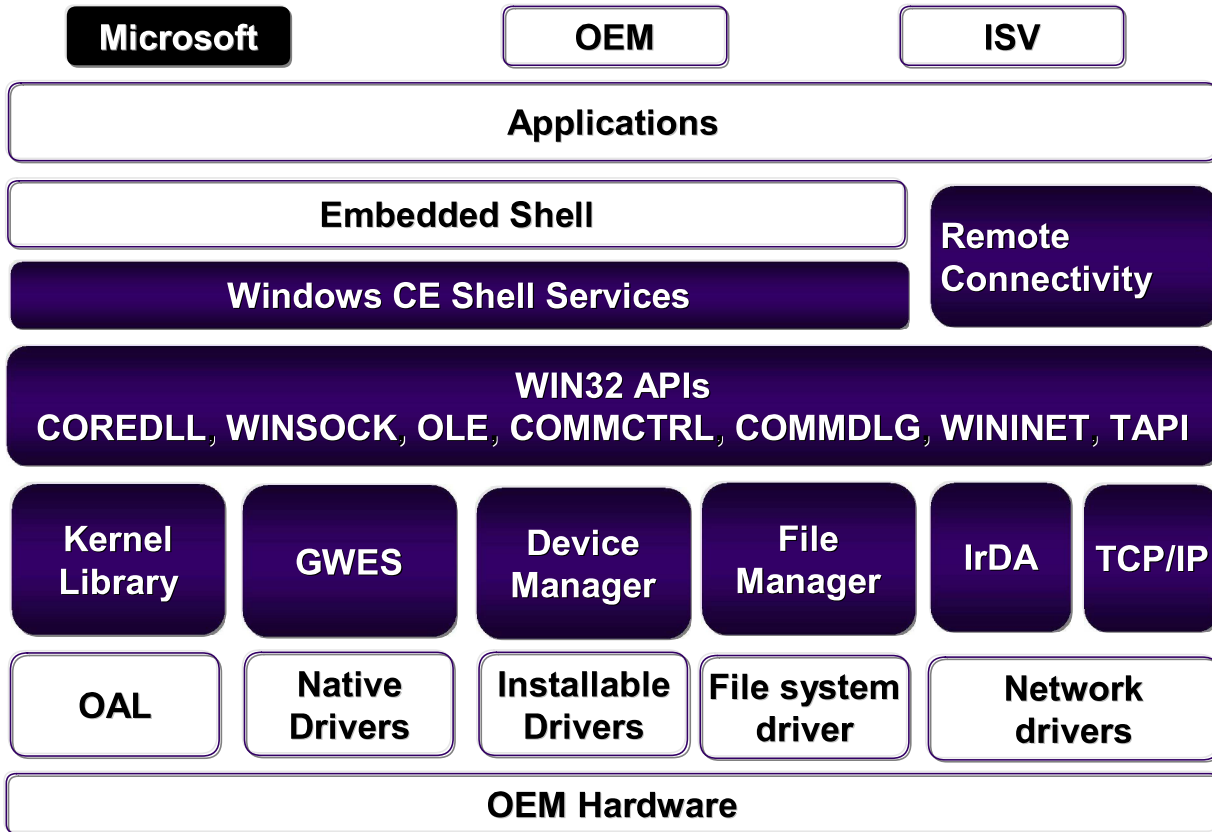
### ● WinCE

WinCE, Windows CE, 是微软公司开发的一个开放、可升级的嵌入式操作系统。由于它的图形用户界面相当出色, 通用应用于基于掌上型电脑类的电子设备中。

WinCE, 精简版的Windows 95, 它具有模块化、结构化和基于Win32应用程序接口以及与处理器无关特点。WinCE不仅继承了传统的Windows图形界面, 并且在WinCE上平台可以使用Windows的编程工具(如Visual C++等)、使大多数应用软件只需简单的修改和移植就可以在WinCE平台上继续使用。

# WinCE的架构

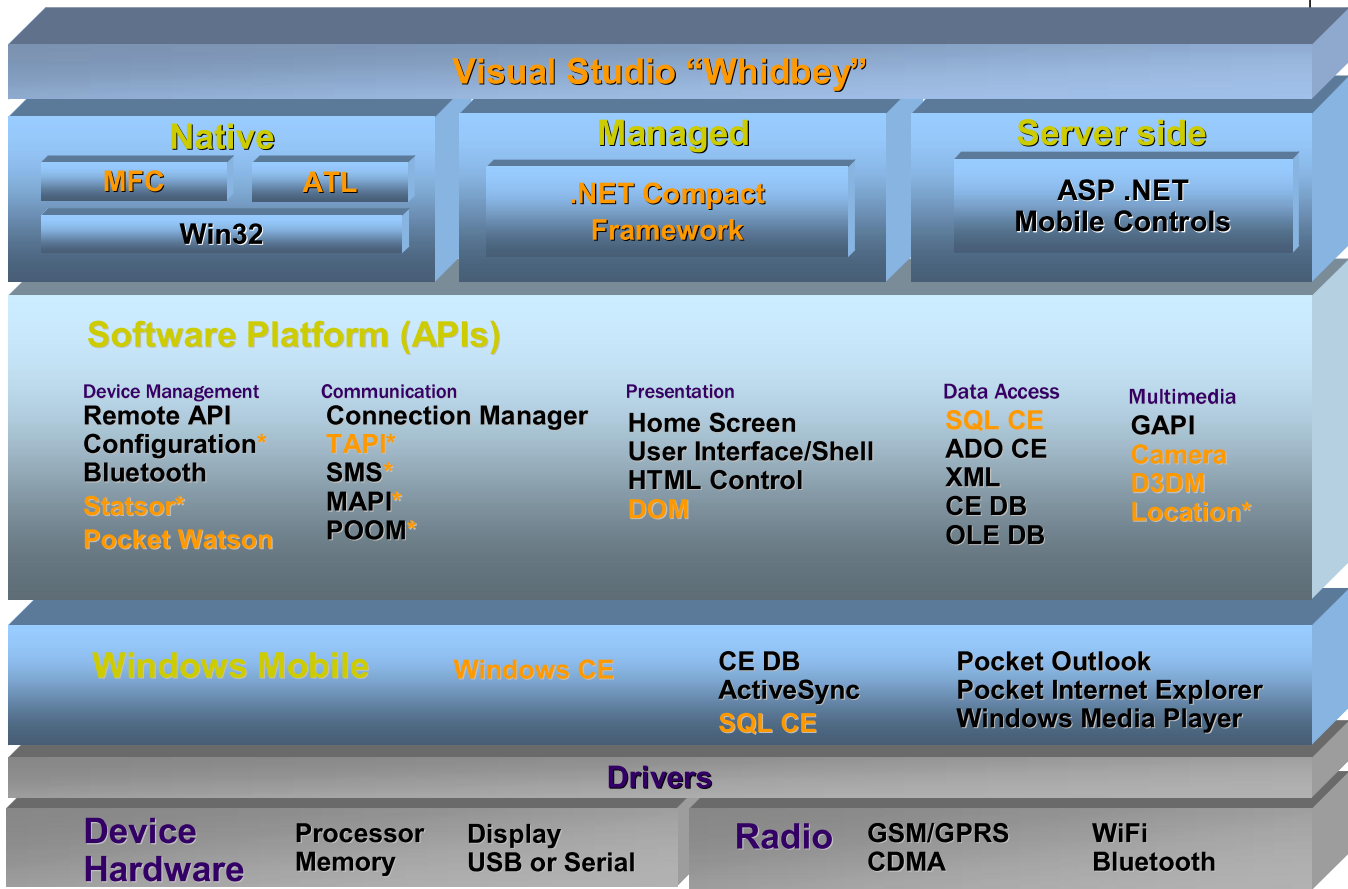
Windows CE





# WinCE的架构

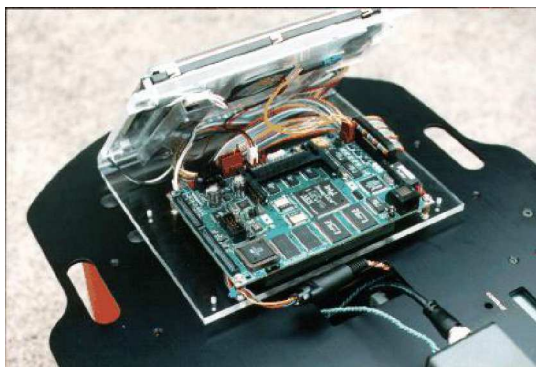
Windows Mobile Platform





# WinCE的应用

- 基于WinCE的移动机器人平台:





# WinCE的应用

- 基于WinCE的机器人装配平台:





# 嵌入式操作系统基础

- 4 常见的嵌入式操作系统

- 嵌入式Linux

嵌入式Linux, Embedded Linux, 是对Linux经过小型化裁剪后, 能够固化在容量为几百K或几M字节的存储器芯片或单片机中, 应用于特定嵌入式的场合的专用Linux操作系统。

嵌入式Linux的开发和研究是目前操作系统领域的一个热点。嵌入式Linux主要代表有uClinux。



# 嵌入式操作系统基础

- 4 常见的嵌入式操作系统

- 嵌入式Linux

uClinux，从Linux2.0/2.4派生而来，沿袭了主流Linux的大部分特性，完全开放代码。

它保留Linux大部分优点：稳定、良好的移植性、优秀的网络功能、完备的对各种文件系统的支持以及标准丰富的API等。



# 嵌入式操作系统基础

- 4 常见的嵌入式操作系统

- 嵌入式Linux

uClinux，通常被应用于具有很少内存或Flash的嵌入式系统，编译后目标文件可控制在几百KB量级，内核小于512KB，采用时间片轮番调度算法。

它专门针对没有虚拟内存或内存管理单元(MMU)处理器，如ARM7TDMI，并且为嵌入式系统做了许多小型化的工作。



# uC/OS和uClinux的对比

	uC/OS	uClinux
任务调度	抢占式	非抢占式
内存管理	独立堆栈空间	进程共享空间
文件系统	本身不支持	完善的文件系统
移植性	硬件要求低 方法简单	硬件要求高 方法复杂
功能	只提供基本功能 和扩展接口	继承Linux大部分 强大功能
易学习性	短小精悍、结构优美、 可读性强	代码量大、结构复杂、 不易读懂
应用类型	适合小型控制系统	适合PDA互联网终端等 复杂系统

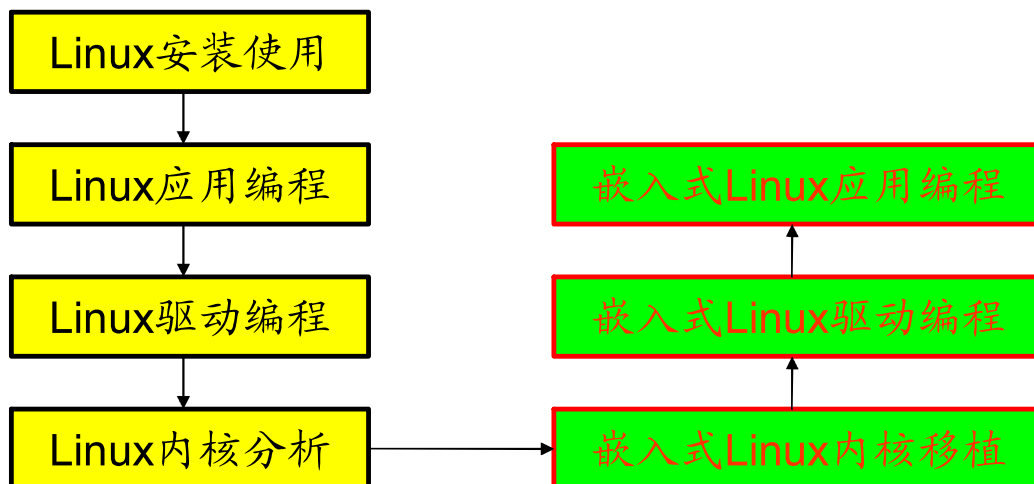


# 嵌入式操作系统基础

## ● 4 常见的嵌入式操作系统

### ● 嵌入式Linux

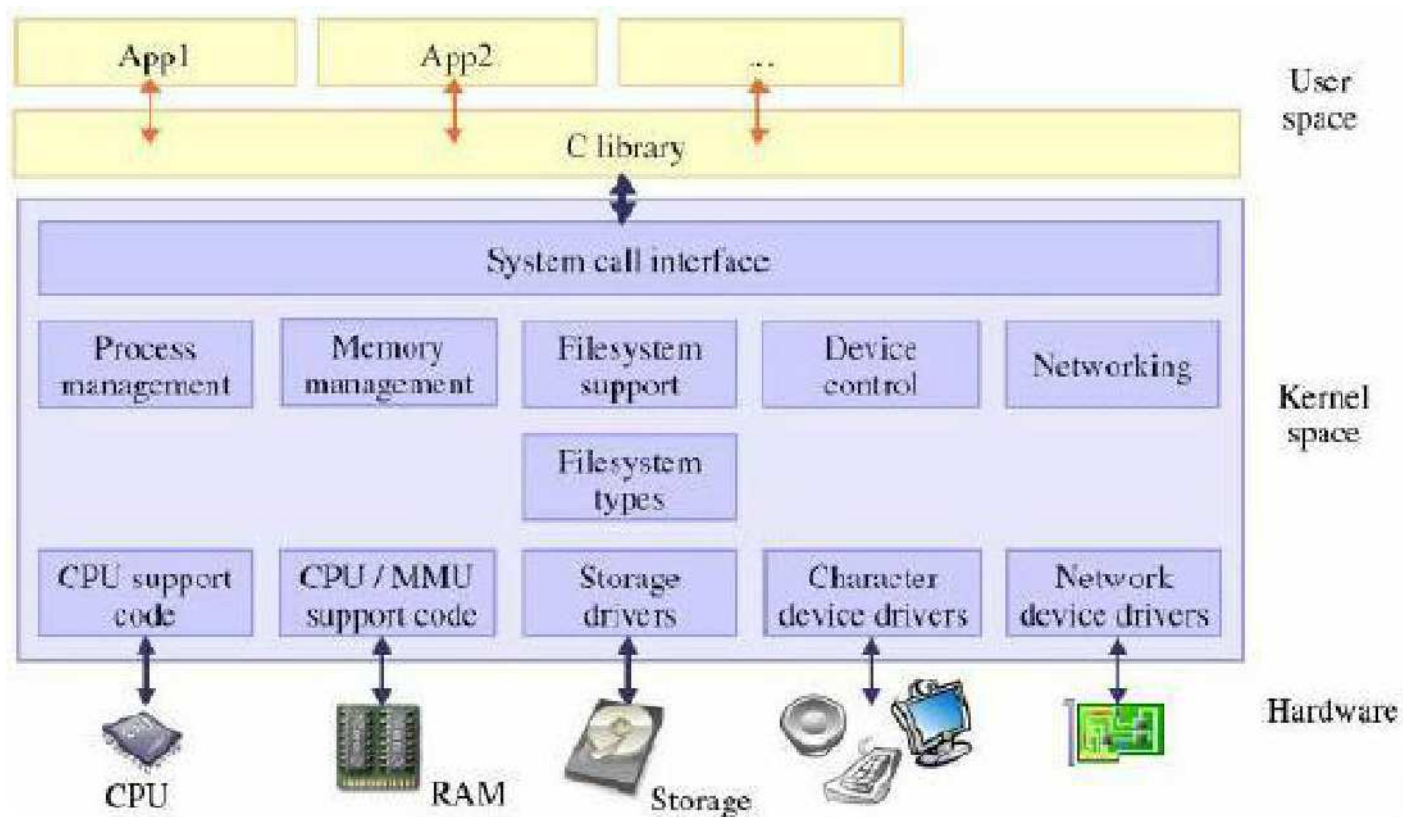
嵌入式Linux学习之路： 从Linux到嵌入式Linux





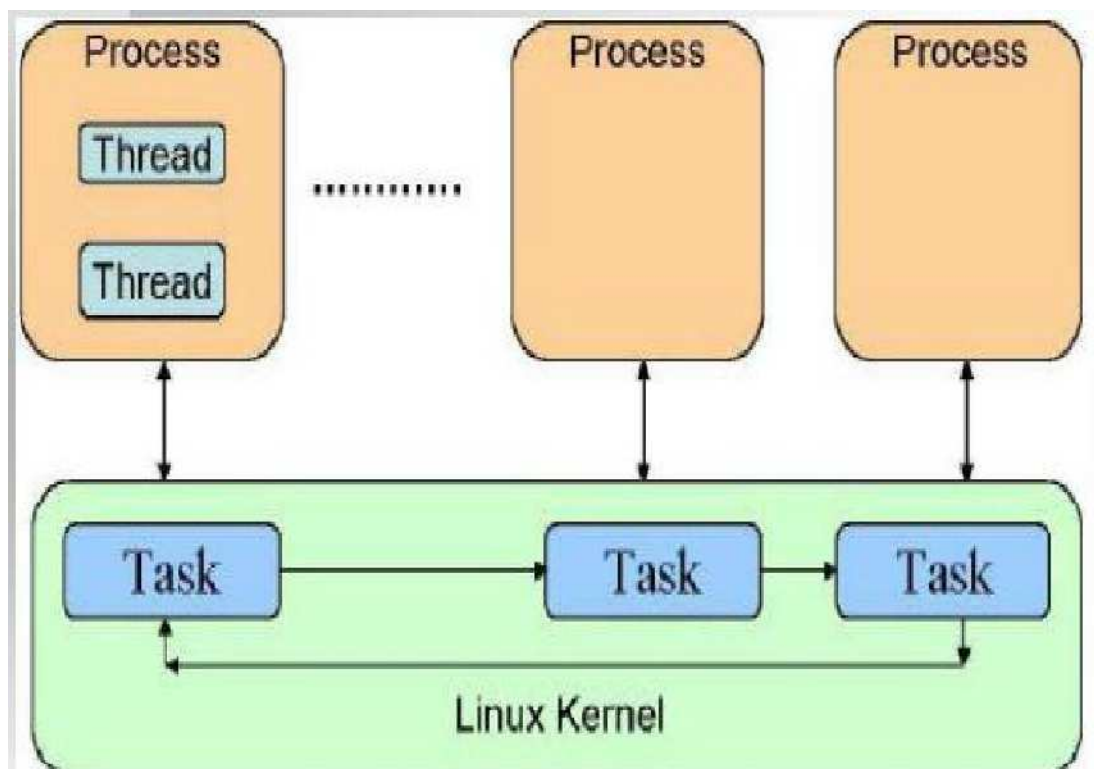


# 嵌入式Linux的结构



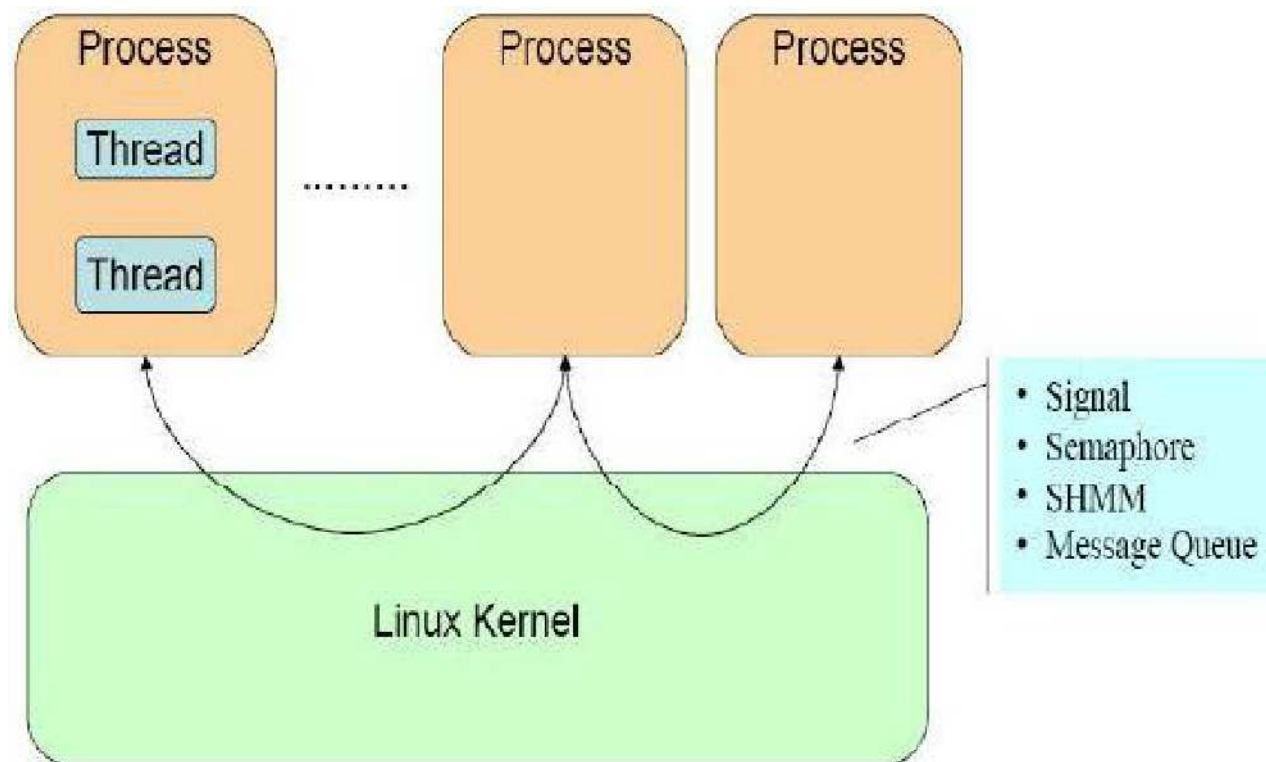


# 嵌入式Linux的进程管理



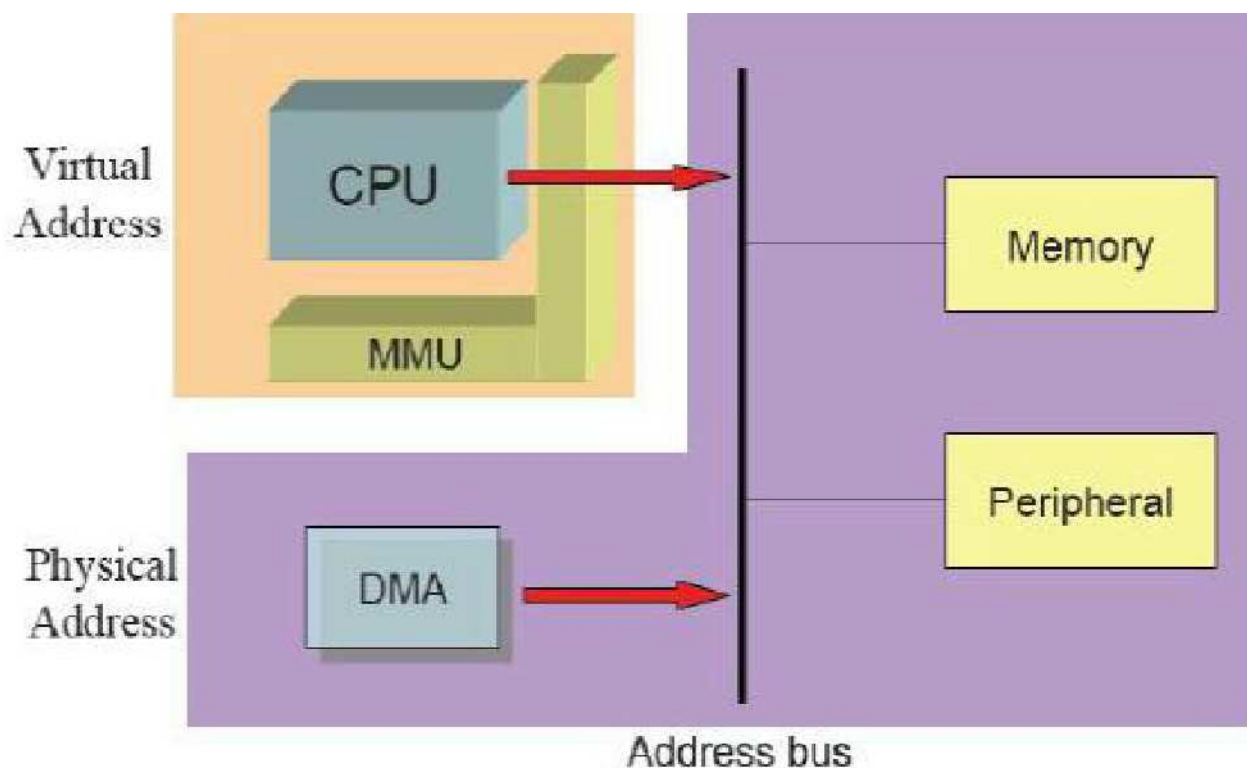


# 嵌入式Linux的进程间通信





# 嵌入式Linux的存储管理



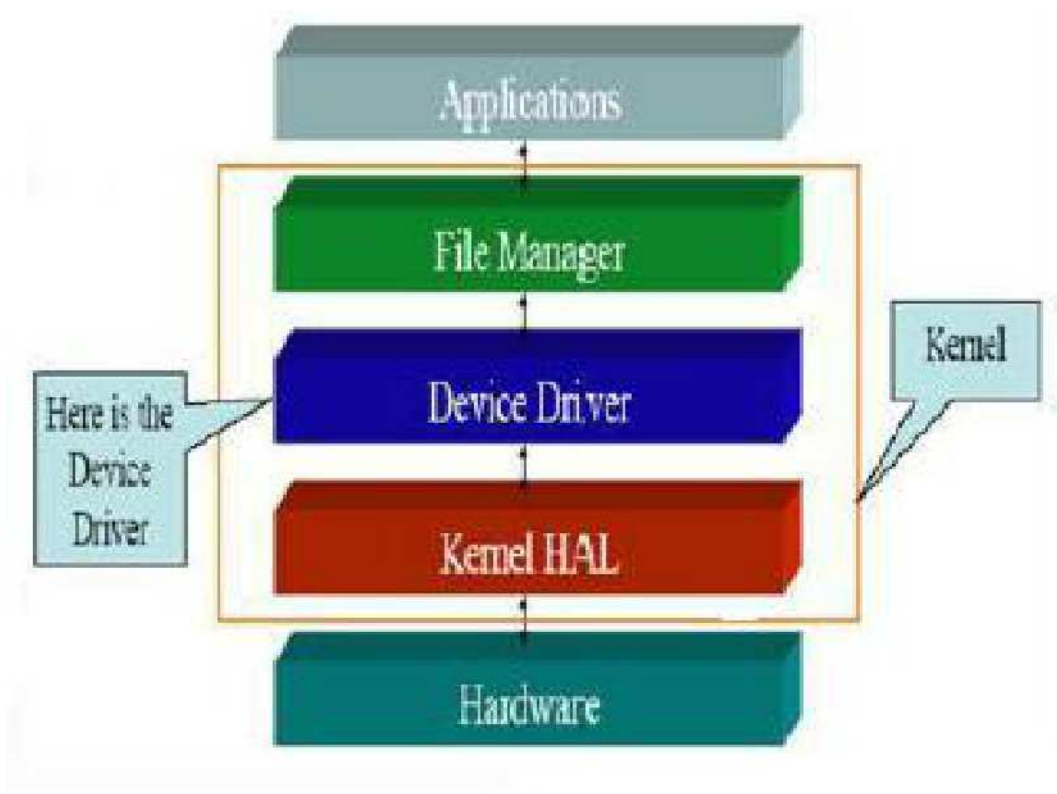


# 嵌入式Linux的文件管理

filesystem	write	persistent	Power down reliability	compression
CRAMFS	No	N/A	N/A	Yes
Jffs	Yes	Yes	Yes	No
Jffs2	Yes	Yes	Yes	Yes
Yaffs	Yes	Yes	Yes	Yes
Ext2 over NFTL	Yes	Yes	No	No
Ext3 over NFTL	Yes	Yes	Yes	No



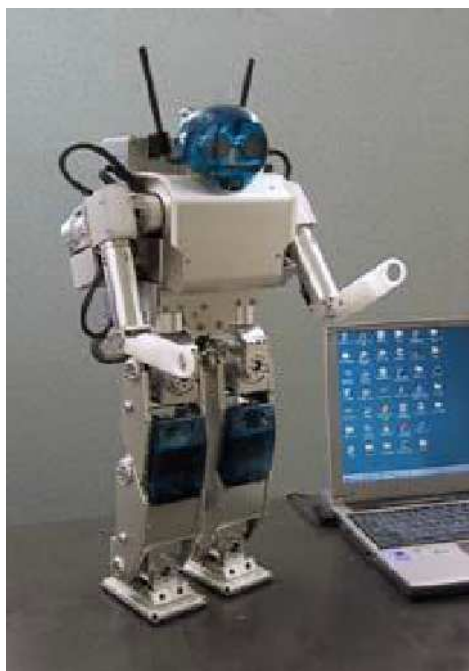
# 嵌入式Linux的设备驱动





# 嵌入式Linux的应用

- 基于RTlinux的仿人机器人:



高 48 cm

重: 6 kg

灵活性: 20 DOF

操作系统: RT-Linux

接口形式: USB 1.0 (12Mbps)

响应周期: 1ms

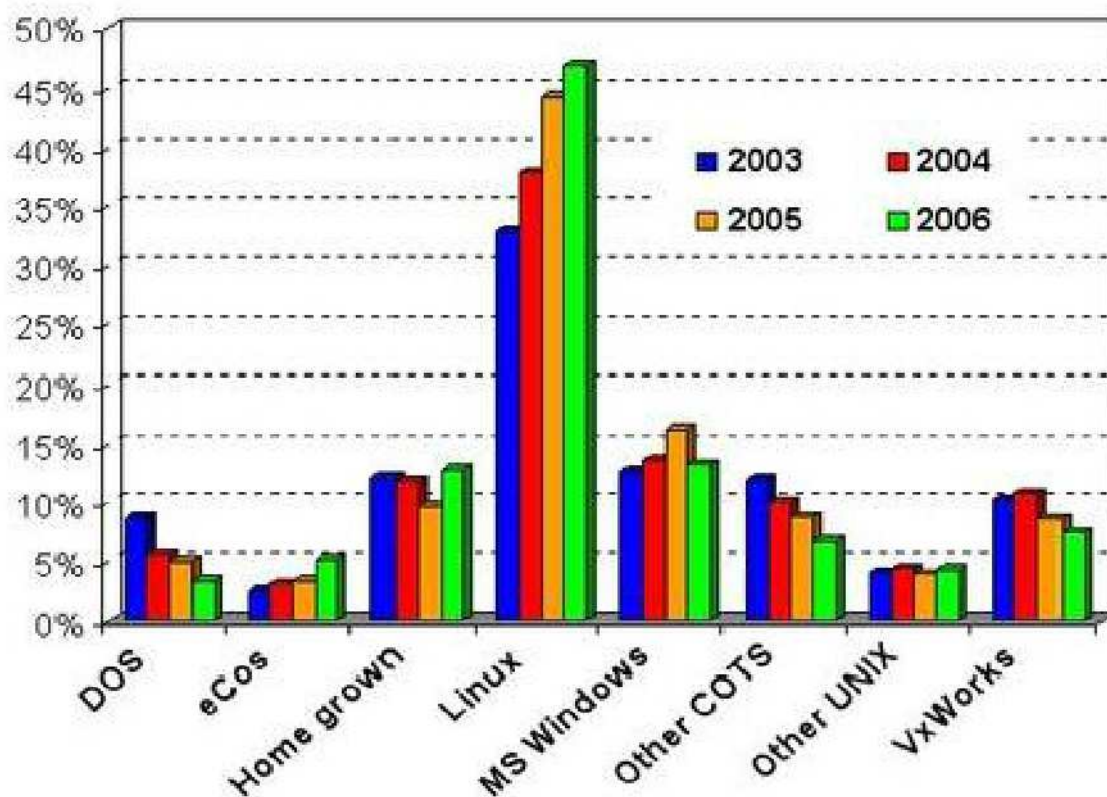
能源: DC24V x 6.2A (150W)

制造: 富士通



# 常用的嵌入式操作系统小结

Embedded OS sourcing trends







# 嵌入式操作系统基础

- 4 常见的嵌入式操作系统

- 国产嵌入式操作系统

操作系统	供应商	应用领域
Hopen	凯思昊鹏	通信、消费电子
Delta	科银京成	军工、通信、消费电子
Smart OS	浙江大学	通信、消费电子
Elast OS	科泰世纪	通信终端
Reworks	华东计算所	通信、工业控制、航空航天等
嵌入Linux	中科红旗、中标软等	通信、消费电子