

# 面向二十一世纪的嵌入式系统设计技术

## 第四讲： 嵌入式操作系统介绍 Real Time Operation System

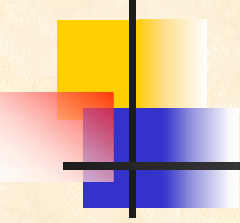
任课教员：徐欣 博士

主讲教员：习勇 博士



2002年1月

国防科大电子科学与工程学院  
嵌入式系统开放研究小组

- 
- 
- ---嵌入式操作系统的基本概念
  - ---uC/OS
  - ---嵌入式linux



## 嵌入式系统的定义

---

以**应用**为中心，计算机技术为基础，软硬件可裁剪，适应于特定应用系统，对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。

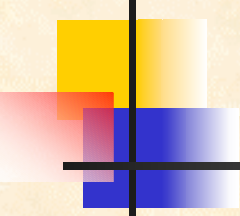




## 嵌入式系统的特点

---

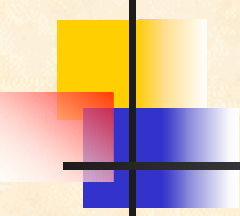
- 硬件上，体积小、重量轻、成本低、可靠性高等特点、使用专用的嵌入式CPU。
- 软件上，代码体积小、效率高，要求响应速度快，能够处理异步并发事件，实时处理能力。



## 嵌入式系统的应用

---

目前，从航天飞机到家用微波炉，嵌入式计算机系统广泛应用到工业、交通、能源、通信、科研、医疗卫生、国防以及日常生活等领域，并发挥着极其重要的作用。



## 嵌入式系统的软件

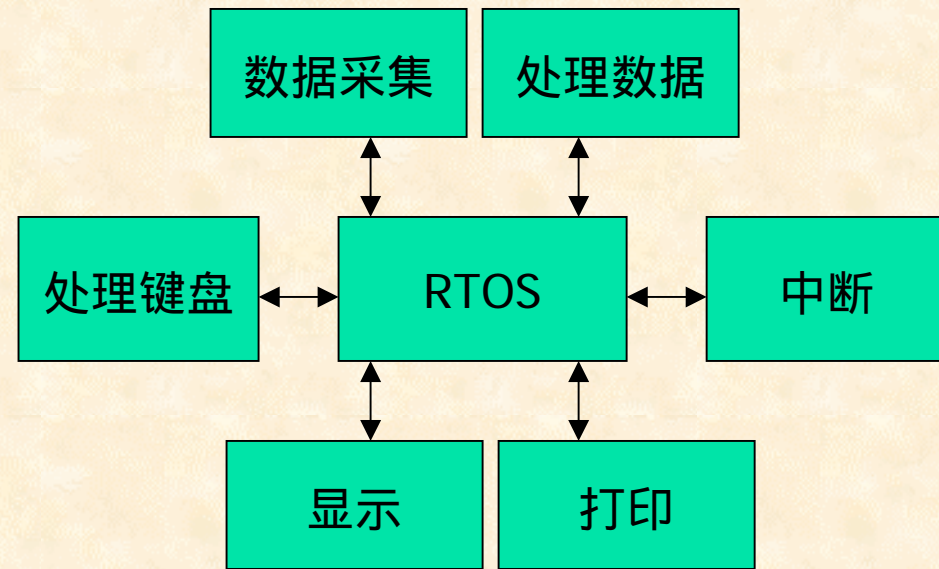
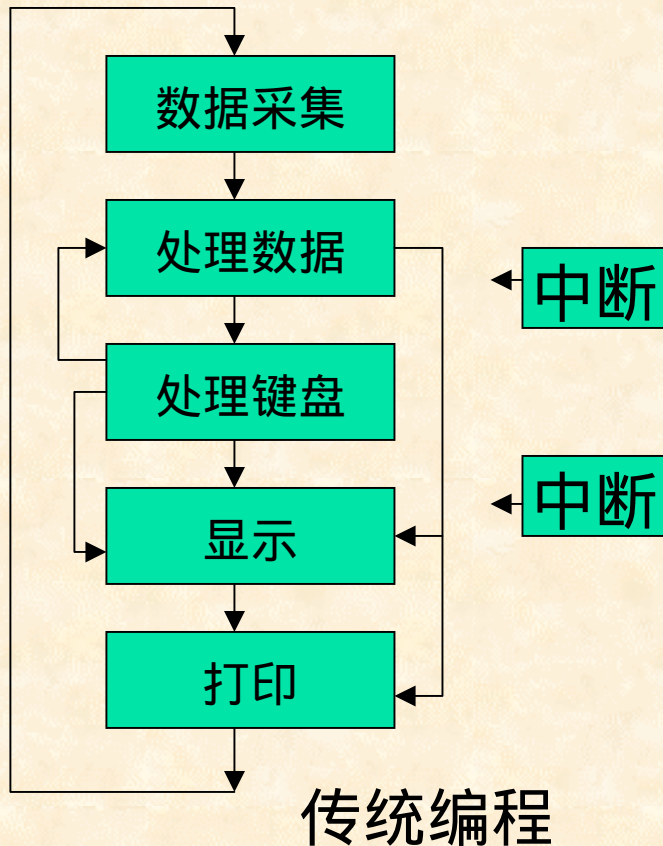
---

现代信息社会的高效性、协同性客观上要求软件的编写便于多人分工合作、编写的软件具有可重用性。软件则日趋复杂。所有这些迫切需要一个屏蔽底层硬件的、功能强大的操作系统来支持。我们认为嵌入式软件的核心在于嵌入式实时多任务操作系统(RTOS –real time operation system)。



## ----why RTOS

数据采集终端：采集，处理，键盘，  
LCD显示，微打



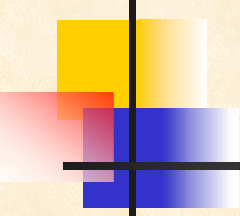


# 操作系统的概念

---

- 操作系统是软硬件资源的控制中心，它以尽量合理有效的方法组织多个用户共享计算机的各种资源。
- 目的是提供一台功能强大的虚拟机，给用户一个方便、有效、安全的工作环境。





## 实时操作系统的概念

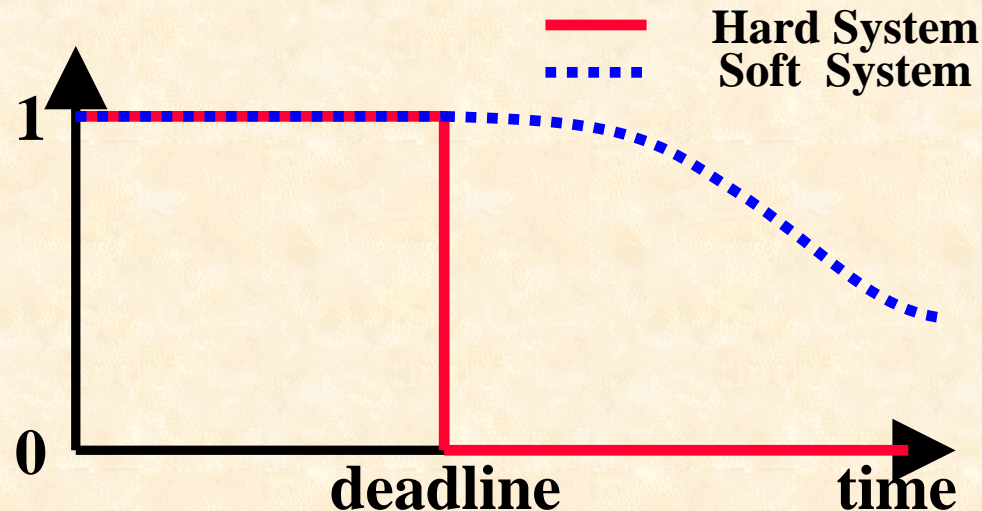
---

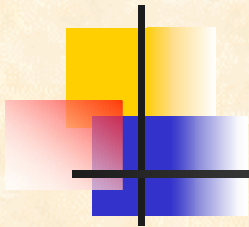
实时系统：系统的正确运行要同时满足特定的逻辑和时间关系。

实时操作系统RTOS (real time operation system)：是指能支持实时控制系统工作的操作系统

## 硬实时(hard)与软实时(soft)

- 硬实时：通过系统特定的时序得不到满足，将会引起灾难性的后果。
- 软实时：通过系统特定的时序得不到满足，系统的性能会严重下降。





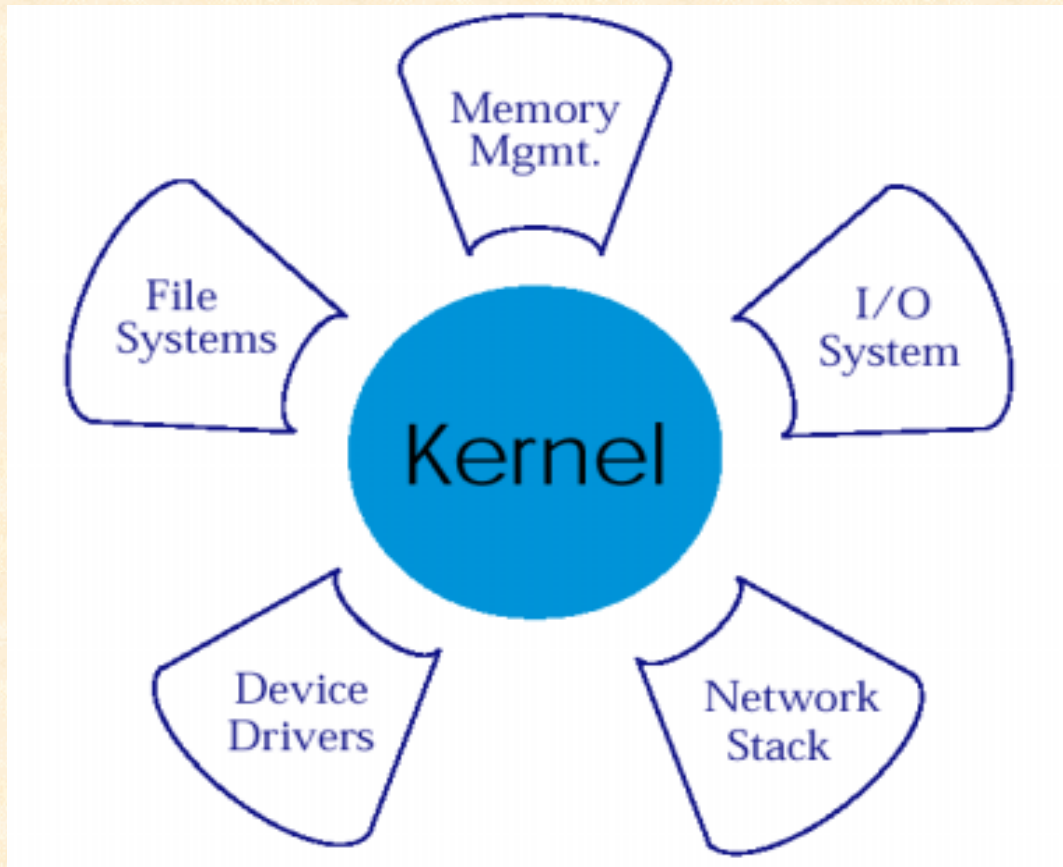
### 嵌入式操作系统

---

嵌入式操作系统：支持嵌入式系统的工作的操作系统。嵌入式系统一般具有实时特点，这里把嵌入式操作系统和实时操作系统不加区别对待。



## 嵌入式操作系统的结构





## 嵌入式操作系统的几个主要概念

---

- 任务 Task
- 调度 Scheduler
- 优先级 Priority
- 代码临界区 Critical Section
- 可预测性 predictability
- 上下文切换 Context Switch



# 任务 Task

---

- 指拥有所有CPU资源的简单程序。
- 在进行实时应用设计时通常要把工作分割成多个任务，每个任务处理一部分问题，并被赋予一定的优先级、一套自己的CPU寄存器及堆栈。
- 实时系统中的大部分任务是周期的，体现在编程上每个任务则是一个典型的无限循环。
- 任务的状态：睡眠、就绪、运行、延迟、等待





## 任务：example

---

### 手持数据采集终端的打印

```
creatTASK(myprintf, priority)
```

```
void myprintf( )
```

```
{  
    for (;;)   
    {  
        等待接收队列信息；  
        往打印机发送打印信息；  
    }  
}
```



## 内核(kernel)

---

- 多任务系统的一部分，负责管理任务。
- 占先式(preemptive)与非占先式
- 微内核(Micro kernel)与单内核(monolithic kernel)



# 调度 Scheduler

---

- 内核的主要职责之一，决定任务运行的次序。
- 基本的调度算法有先来先服务FCFS, 最短周期优先SBF，优先级法(Priority)，轮转法(Round-Robin)，多级队列法(multi-level queues)，多级反馈队列(multi-level feedback queues)等。
- 调度的基本方式有可占先式和非占先式。
- 多数实时内核是基于优先级调度的多种方法的复合。





# 优先级 ( Priority )

---

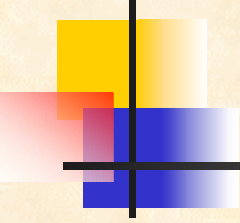
- 每个任务按其重要性被赋予一定的优先级。
- 静态优先级与动态优先级。
- 基于优先级的系统会出现优先级倒置的问题，一个好的实时内核应该提供解决倒置的方法。
- 已开发出多种算法用于实时任务的优先级分配，基本的有单调执行率调度法RMS和最早期限优先法EDF等。



## 代码临界区(Critical Section)

---

- 指一段不可分割的代码，一旦执行，不能被中断。
- 实现代码临界区的方法有：一是屏蔽中断，通常在代码执行前关闭中断，执行后打开中断，只能用于单处理机的情形；二是通过信号量机制。



# 互斥 (mutual exclusion)

---

- 资源的使用必须独占，叫做互斥。
- 解决办法：开关中断，使用硬件指令，使用信号量





# 任务间通信(inter task com)

---

- 在多任务系统中，任务之间存在相互制约的关系，或者任务之间需要交换信息，称为任务间通信。
- 方式：邮箱，队列，事件标记等。



## 时钟节拍(Clock tick)

---

- 一种周期性发生的特殊中断，该中断可视为系统心脏的跳动。
- 中断周期越短，系统响应速度越快，但开销也越大，程序的执行速度越慢。
- 典型中断时间在10-200ms之间，取20ms较为合适。



## 可预测性(predictability)

---

- 指在系统运行的任何时刻、任何情况下，实时操作系统的资源调配策略都能为争夺资源(包括CPU、内存、网络带宽等)的多个实时任务合理地分配资源，使各实时任务的实时性要求都能得到满足。
- 简单说：操作系统的行为是可知的。





## 好的实时操作系统需要具备以下功能：

---

- 任务管理（多任务和基于优先级的任务调度）
- 具备消除优先级倒置的机制
- 任务间同步和通信
- 实时时钟服务
- 中断管理服务
- 操作系统的行为是可知的和可预测的



## 评价实时操作系统几个重要指标

---

- 系统响应时间(System response time) : 是指系统发出处理要求到系统给出应答信号的时间。
- 任务切换时间(Context-switching time) : 是指任务之间切换使用的时间。
- 中断延迟(Interrupt latency) : 是指计算机接收到中断信号到操作系统做出响应, 并完成切换转入中断服务程序的时间。

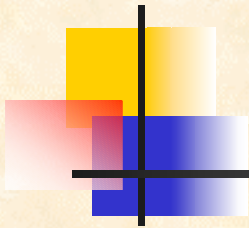


## 实时操作系统与通用操作系统的区别

---

- 设计目标：通用操作系统的目标是追求最大的吞吐率、使整体性能最佳;而实时操作系统设计的目标是采用各种算法和策略，始终保证系统行为的可预测性。
- 调度原则：通用操作系统为了达到最佳整体性能，其调度原则是公平；而实时系统多采用基于优先级的可剥夺的调度策略。
- 内存管理：通用操作系统广泛使用了虚拟内存的技术，为用户提供一个功能强大的虚拟机，但因虚存机制引起的缺页调页现象会给系统带来不确定性，因此实时系统很少或有限的使用虚存技术。

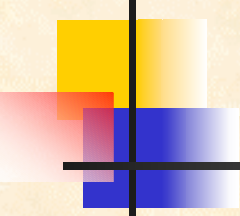




## 嵌入式操作系统的发展历程

---

初期阶段	实质是一种实时的监控程序
第二阶段	专用系统
第三阶段	通用系统



## 嵌入式操作系统的分类

---

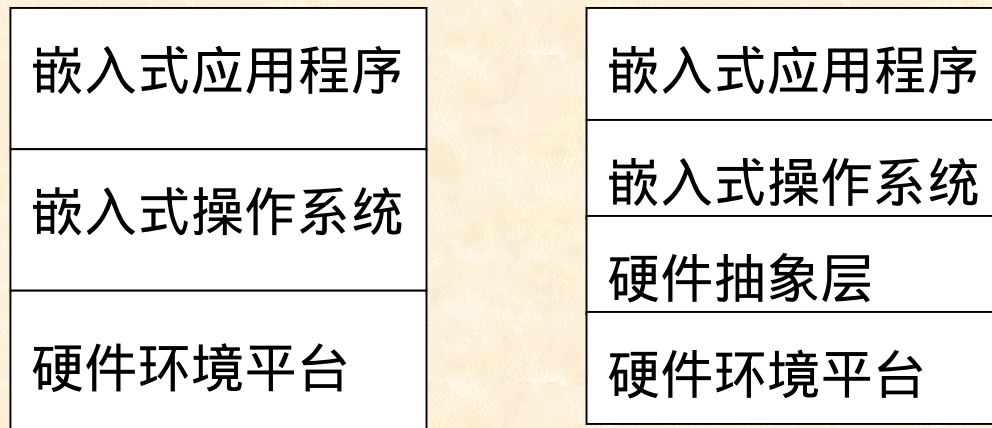
从嵌入式系统的应用来分类，可以分为面向低端设备的嵌入式操作系统和面向高端设备的嵌入式操作系统。

低端：各种工业控制系统，计算机外设，民用消费品的微波炉，洗衣机，冰箱等,ucos

高端：信息化家电，掌上电脑，机顶盒、WAP手机，路由器，wince, linux.

## 操作系统的移植

### ■ 硬件抽象层的概念



嵌入式系统结构

引入硬件抽象层后嵌入式系统结构

### ■ BSP(board support packet)





## BSP特点与功能

---

- BSP主要特点：硬件相关性与操作系统相关性
- 在实现上，BSP是一个介于操作系统和底层硬件之间的软层次，包括了系统中大部分与硬件相关的软件模块。
- 在功能上包含两部分：系统初始化及与硬件相关的设备驱动。
- 系统初始化完成的基本功能有：对CPU进行低级初始化、对主板的硬件进行初始化、加载操作系统。



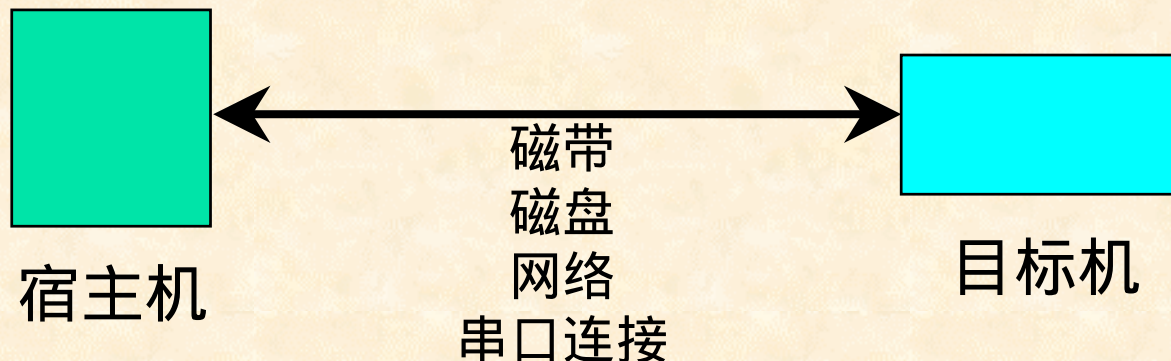
# 嵌入式系统的开发环境

---

- 嵌入式系统的一个特点在于其开发的特殊性与困难性。
- 开发机器  $\neq$  执行机器
- 开发环境  $\neq$  执行环境
- 专门的交叉编译开发环境cross compile

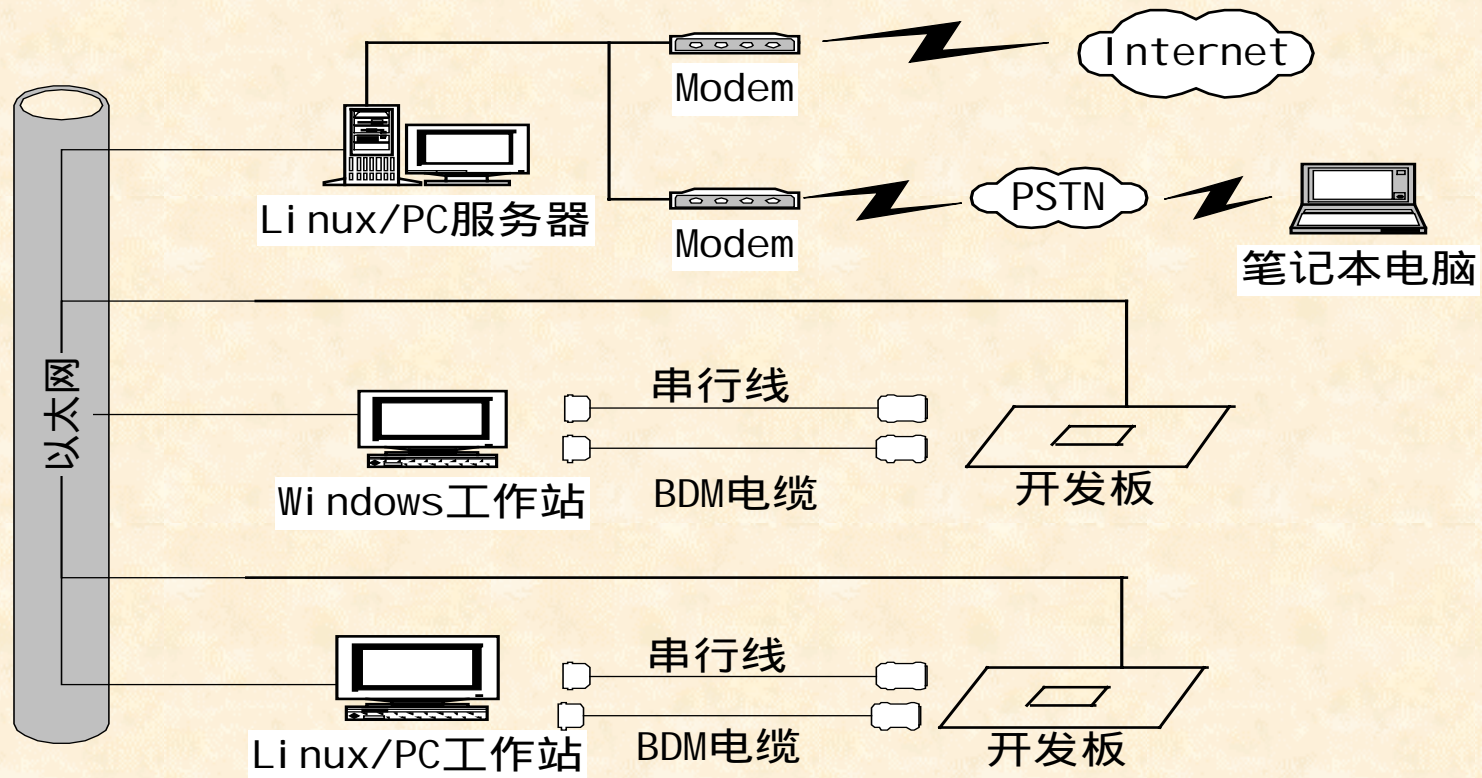
# 开发环境

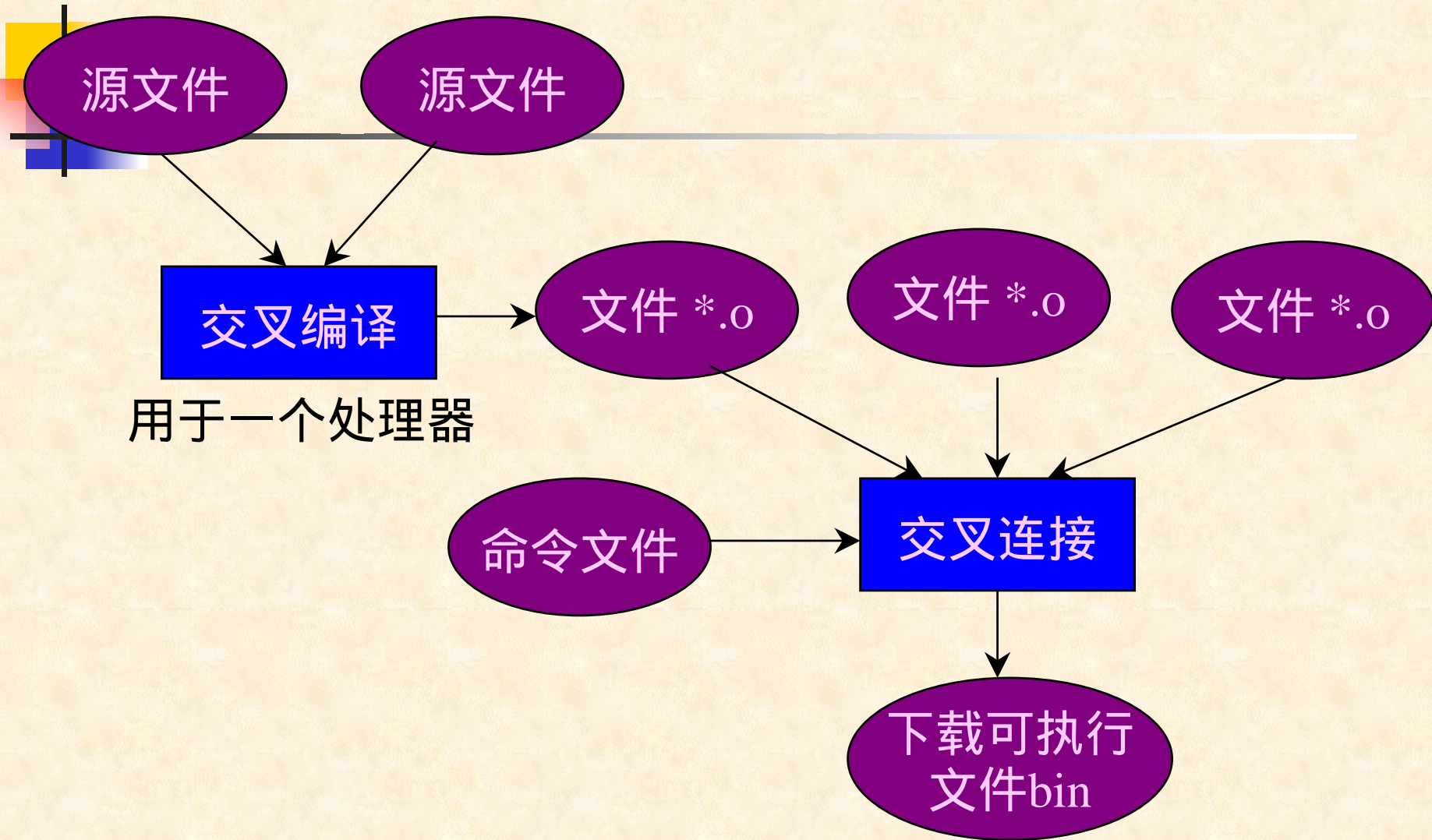
- 宿主机
  - 开发机器（编辑器、编译器、调试器、....）
- 目标机
  - 程序运行的机器
- 宿主机和目标机一样时为 **本地编译**
- **交叉编译**是指宿主机和目标机是不同的系统（必须做目标文件下载）



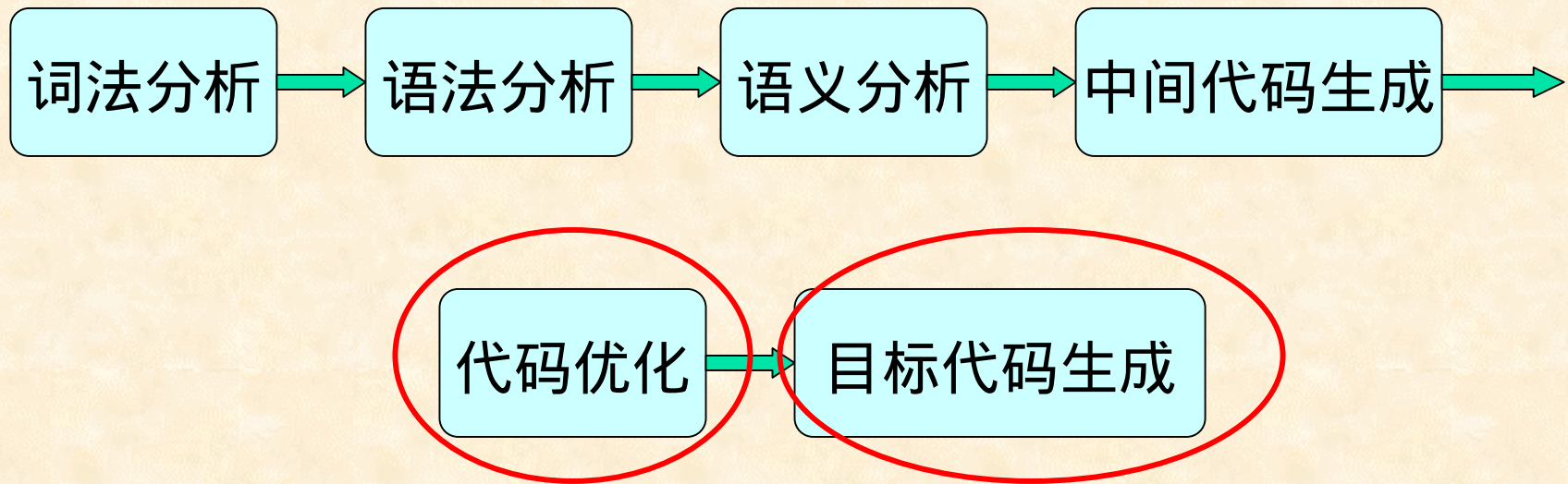


# 嵌入式操作系统介绍





# 嵌入式系统的编译技术







## 嵌入式系统编译技术—代码优化

---

- 优化选项以应用于不同环境
- 便于汇编级调试
- 合理安排程序结构
- 干扰程序正常运行



## 嵌入式系统编译技术—目标代码生成

---

- 代码段的分配与地址定位  
XIP (execute in place)技术
- 寄存器分配
- 启动程序 (bootloader) , crt0.o



## 程序在编译后，分为如下几部分

---

- .text** : 程序代码空间；----flash
- .data** : 初始化的数据，如 `int a=10;` ---ram
- .bss** : 未初始化的数据,如 `int a;`
- .sbss** : 短型未初始化的数据,如 `short a;`
- .sdata** : 短型初始化的数据,如 `short a=10;`
- .sdata2** : 常量短型的数据,如 `const short a=10;`





## 寄存器分配

---

- 加快执行速度,编写BSP, C中直接嵌入汇编语言,在汇编中调用C函数,解决参数的传递的问题。
- 编译器默认使用某些寄存器为各种数据类型基地址指针,系统开发人员在设计启动程序时应该加载这些指针的默认值。