



湖南大学

HUNAN UNIVERSITY

一种车用嵌入式实时操作系统内核的设计与实现

答 辩 人：张然

指导老师：李仁发教授，刘彩萍讲师

研究方向：嵌入式软件



提纲

1. 研究背景及意义
2. 基础理论
3. 主要工作
4. 总结与展望



研究背景和意义

□ 背景

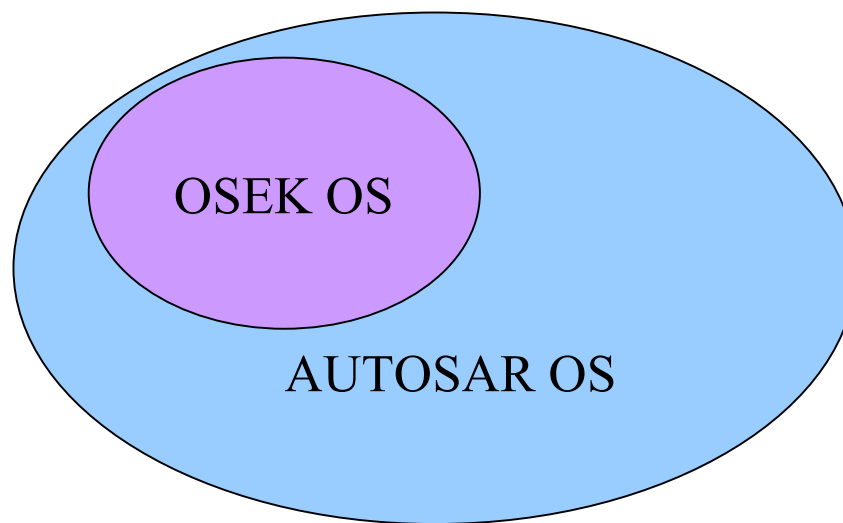
- 课题来源于国家核高基重大专项：实时嵌入式操作系统及开发环境。
- 汽车电子硬件的多元化导致应用软件的可复用性降低，AUTOSAR定义了统一的操作系统规范（AUTOSAR OS）使得软件的生产标准化，大大提高了复用性。

□ 意义

- 研究AUTOSAR规范，设计基于AUTOSAR OS规范的车用嵌入式实时操作系统，对于我国汽车电子的发展具有一定的实际意义。

基础理论—AUTOSAR OS规范

- AUTOSAR OS定义了基于单处理器的静态配置的操作系统，其完全兼容OSEK OS，并添加了许多模块。
- 对操作系统的定义
 - 可裁减类
 - 任务调度
 - 基本任务
 - 扩展任务
 - 中断类型
 - ISR1
 - ISR2
 - 任务同步
 - 事件同步
 - 资源同步
 - 报警器
 - 调度表



AUTOSAR OS与OSEK OS



主要工作

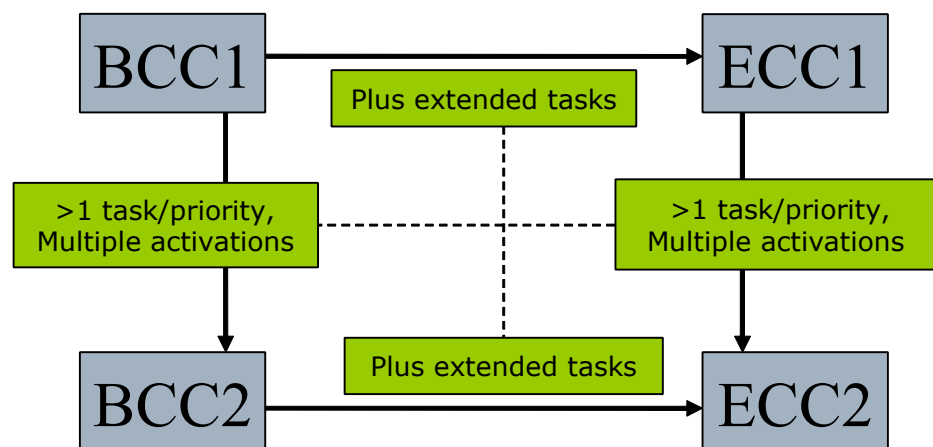
1. 操作系统内核需求分析
2. 操作系统内核设计与实现
3. 操作系统内核功能验证
4. 操作系统内核性能分析



操作系统需求分析

需求分析 > 设计实现 > 功能验证 > 性能分析

- OSEK OS定义的四符合类
- AUTOSAR OS定义四个可扩展类
- 基于ECC2和SC1

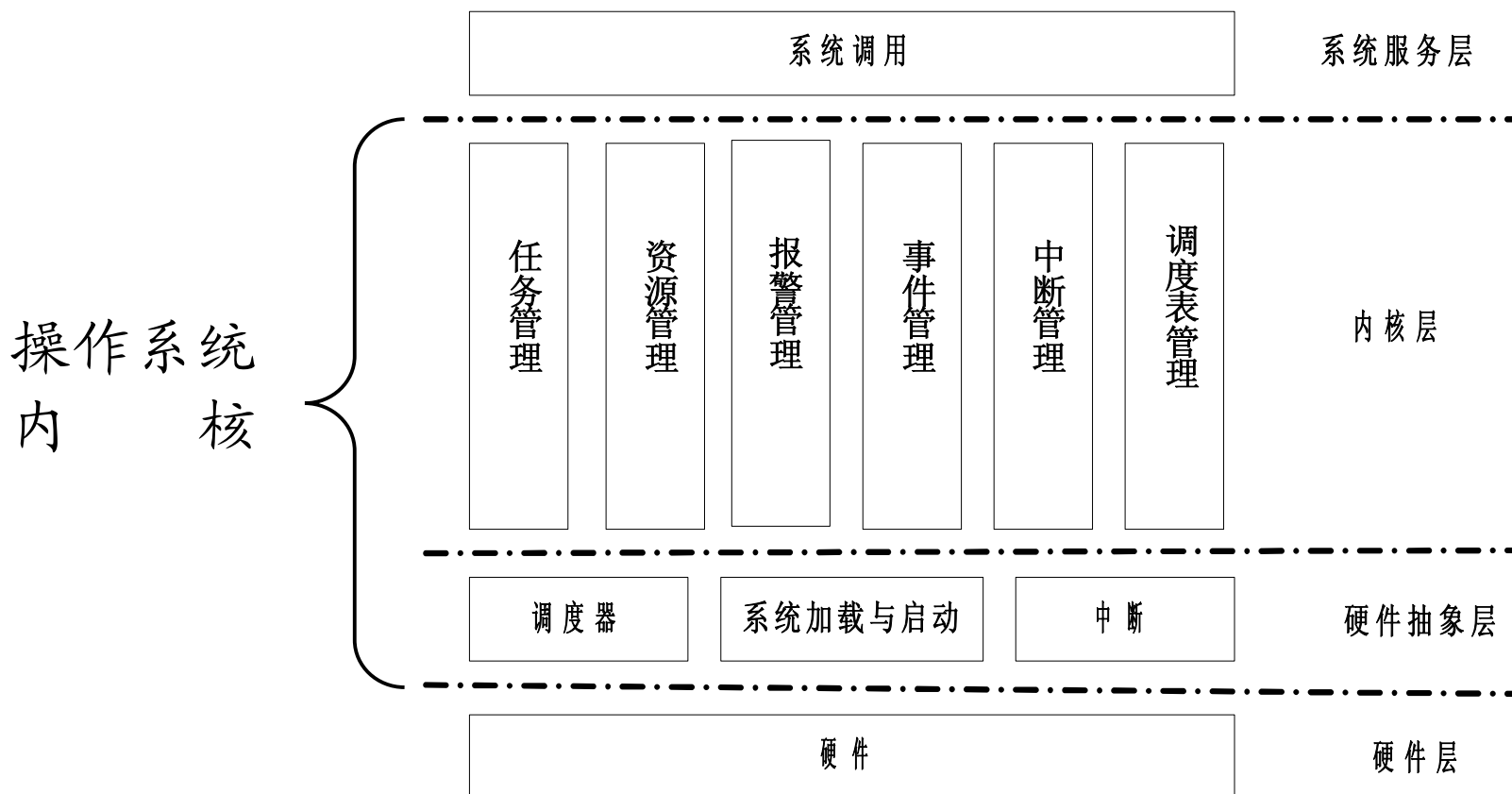


	符合类	计数器接口	调度表	栈监测	保护钩子	时间保护	全局时间同步	内存保护	OS-应用	系统保护	信任函数
SC1	•	•	•	•							
SC2	•	•	•	•	•	•	•				
SC3	•	•	•	•	•			•	•	•	•
SC4	•	•	•	•	•	•	•	•	•	•	•



操作系统设计与实现——架构设计

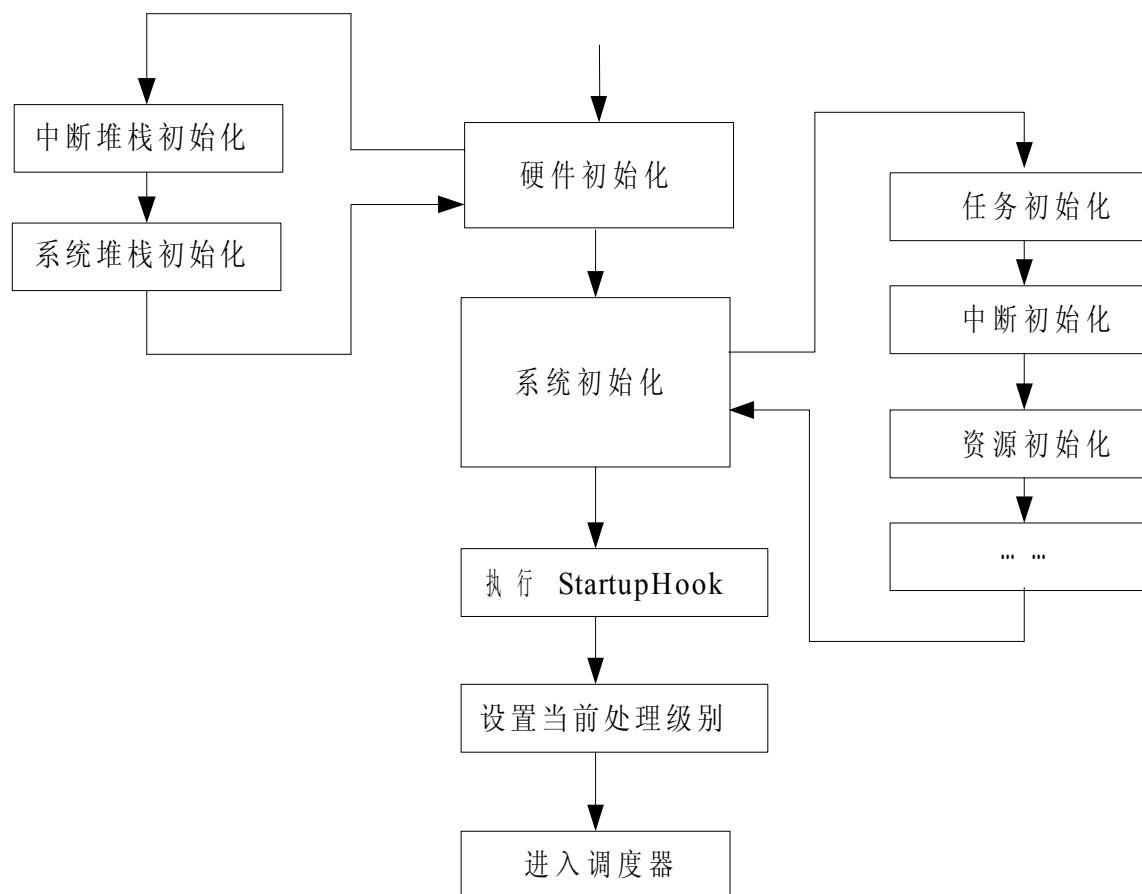
需求分析 > 设计实现 > 功能验证 > 性能分析





操作系统设计与实现——加载与启动

需求分析 > 设计实现 > 功能验证 > 性能分析





操作系统设计与实现——任务管理

需求分析 > 设计实现 > 功能验证 > 性能分析

- 全抢占调度策略
- 基于优先级的任务调度

```
typedef struct
{
    UINT8      info; /* 任务的属性信息 */
    PRIORITY   init_pri; /* 任务的初始优先级 */
    UINT8      maxact; /* 任务的最大激活次数 */
    UINT32     stack_sz; /* 任务的堆栈大小 */
    ADDRESS    stack; /* 任务的堆栈基地址 */
    ADDRESS    task; /* 任务入口函数地址 */
}TINIB;
```

ROM

```
typedef struct
{
    TaskStateType state; /* 任务状态 */
    PRIORITY      current_pri; /* 任务的当前优先级 */
    #if defined (O S _B C C 2) || defined (O S _E C C 2)
    TaskType      next_task; /* 下一个要执行的任务 */
    #endif
    ADDRESS       pc; /* 任务的当前指令地址 */
    ADDRESS       sp; /* 任务当前的栈地址 */
    UINT32        actcnt; /* 任务当前的激活次数 */
    #if defined (O S _E C C 1) || defined (O S _E C C 2)
    EventMaskType curevt; /* 任务当前所获得的事件 */
    EventMaskType waitevt; /* 任务所等待的事件 */
    #endif
    ResourceType lastres; /* 任务所占有的最近一次的资源 */
    TaskType      prev_task; /* 任务的上一个执行的任务 */
}TCB;
```

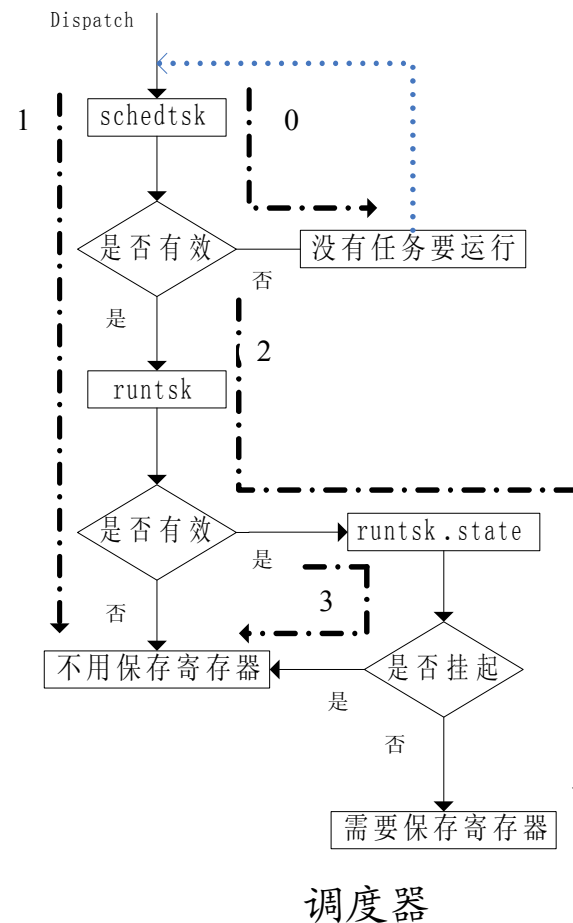
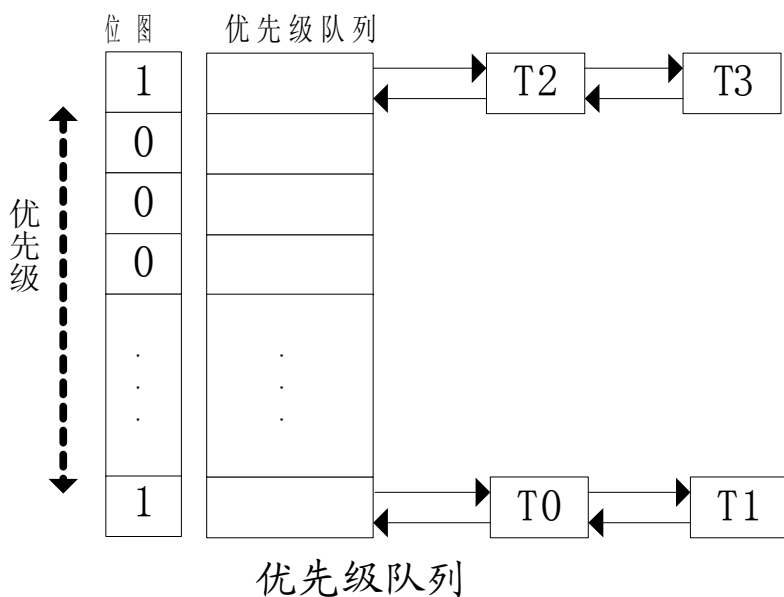
RAM



操作系统设计与实现——任务管理

需求分析 > 设计实现 > 功能验证 > 性能分析

- 全抢占调度策略
- 基于优先级的任务调度

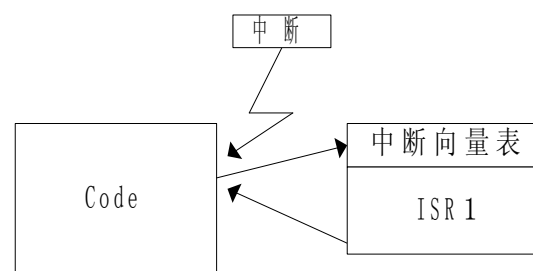




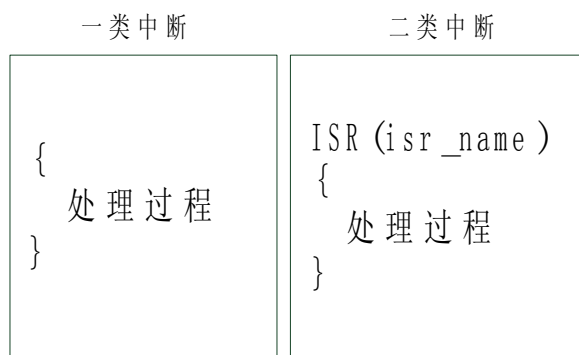
操作系统设计与实现——中断管理

需求分析 > 设计实现 > 功能验证 > 性能分析

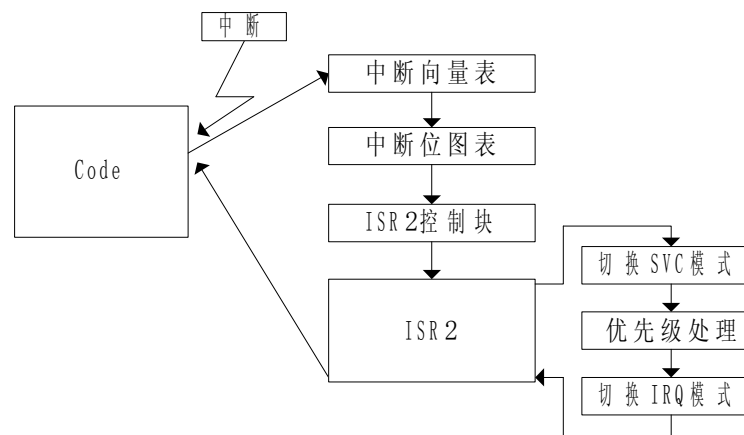
- ISR1和ISR2
- 基于AT91的中断调度
- ISR1优先级最高
- ISR2三级索引可嵌套



ISR1调度



中断类型



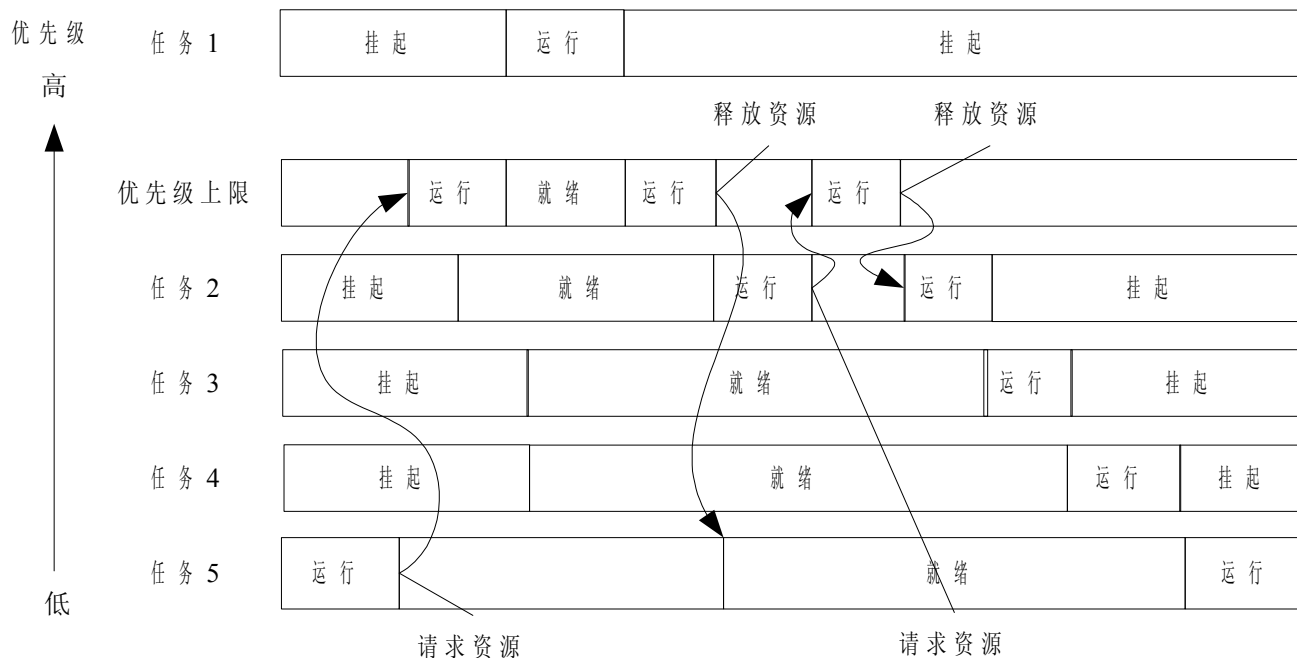
ISR2调度



操作系统设计与实现——资源管理

需求分析 > 设计实现 > 功能验证 > 性能分析

□ 优先级天花板协议用来解决优先级反转和死锁



优先级天花板协议



操作系统设计与实现——资源管理

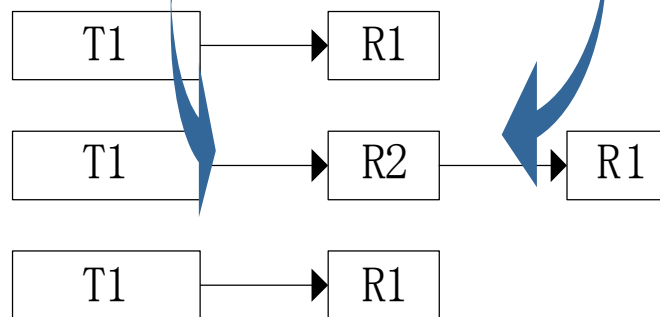
需求分析 > 设计实现 > 功能验证 > 性能分析

- 资源控制块
- 资源申请过程
 - 任务总能申请到资源
 - 在释放资源前不能等待事件或结束

```
typedef struct
{
    PRIORITY ceilpri; //天花板优先级;
    PRIORITY prevpri; //先前的优先级;
    ResourceType prevres; //上一个资源;
}RCB;
```

资源控制块

1. Enter_critical
2. $T = \text{the current task}$
3. if $\text{call_level} == \text{CL_TASK} \ \&\&$
 $T.\text{current_pri} < R.\text{ceilpri}$
4. $R.\text{prevpri} = T.\text{current_pri}$
5. $R.\text{prevres} = T.\text{lastres}$
6. $T.\text{current_pri} = R.\text{ceilpri}$
7. $T.\text{lastres} = R$
8. end if
9. Exit_critical



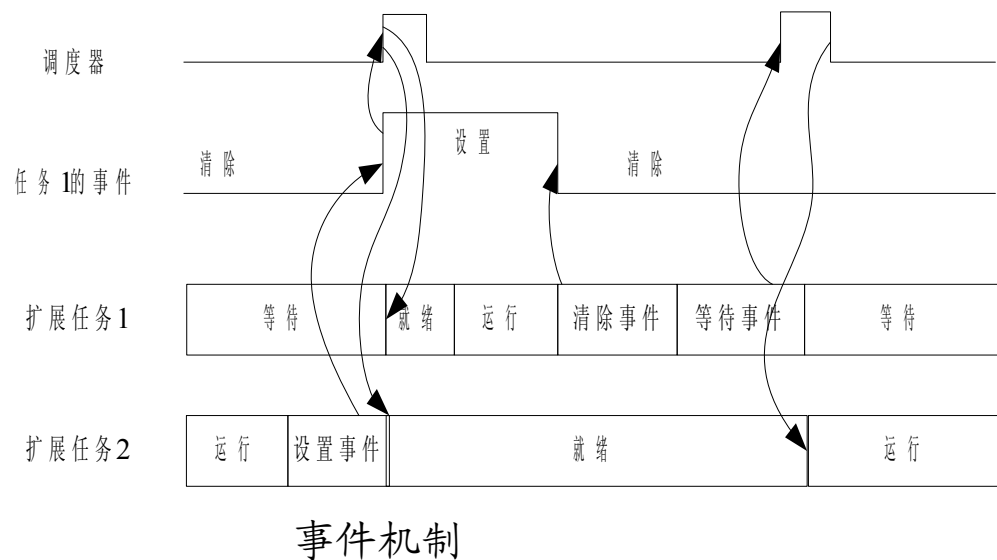
资源申请和释放



操作系统设计与实现——事件管理

需求分析 > 设计实现 > 功能验证 > 性能分析

- 事件仅适用于扩展任务
- 任务包含事件
 - 当前事件
 - 等待事件



```

1.  Enter_critical
2.  T.curevt = mask
3.  if T.curevt & T.waitevt
4.      T.state = READY
5.      T.waitevt &= ~(T.curevt & T.waitevt)
6.      Search task
7.      Enter dispatch
8.  end if
9.  Exit_critical
    
```

```

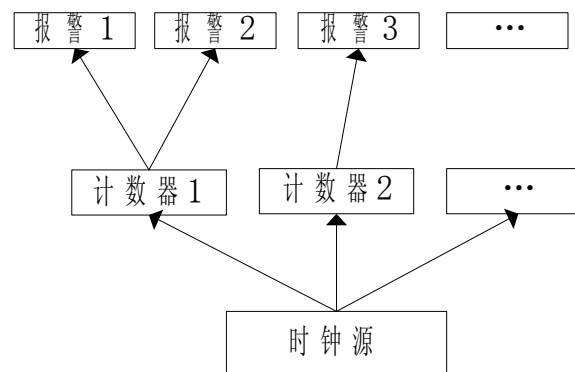
1.  Enter_critical
2.  if !(T.curevt & mask)
3.      T.state = WAITTING
4.      T.waitevt = mask
5.      Search task
6.      Enter dispatch
7.  end if
8.  Exit_critical
    
```



操作系统设计与实现——报警管理

需求分析 设计实现 功能验证 性能分析

- 报警器与计数器关联
- 一个计数器可关联多个报警器，构成队列
- 计数器由时钟源驱动



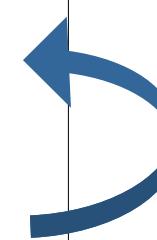
```
typedef struct
{
    TickType maxallowedvalue ;
    TickType ticksperbase ;
    TickType value ;
    AlarmType alarm;
    ScheduleTableType st;
}CounterCB ;
```

计数器控制块

```
typedef struct
{
    CounterType counter;
    UINT8 autostart;
    TickType value;
    TickType cycle;
    FP callback;
} AlarmInit;
```

报警控制块

```
typedef struct
{
    TickType cycle;
    TickType value;
    AlarmType next;
}AlarmCB;
```

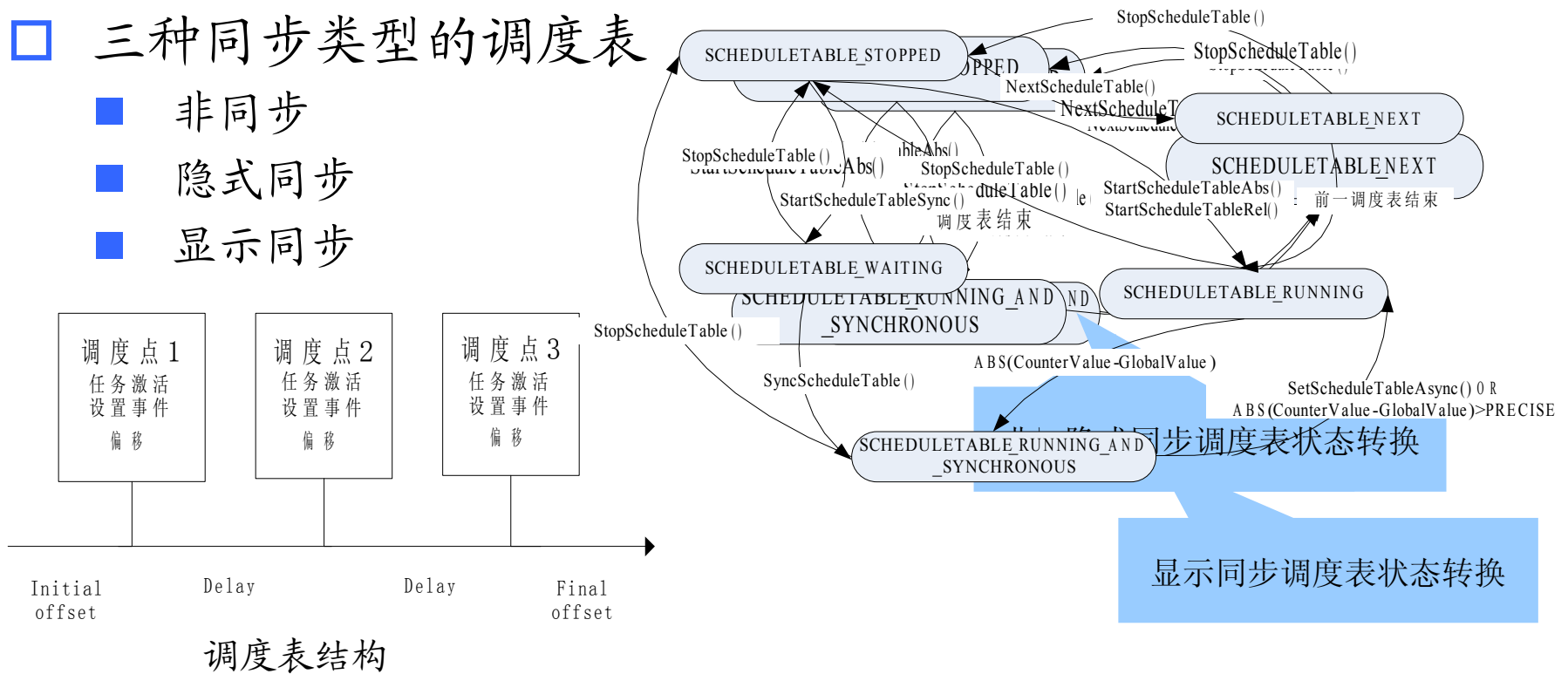




操作系统设计与实现——调度表管理

需求分析 > 设计实现 > 功能验证 > 性能分析

- 调度表由多个调度点组成，在时间上具有持续性
- 三种同步类型的调度表
 - 非同步
 - 隐式同步
 - 显示同步

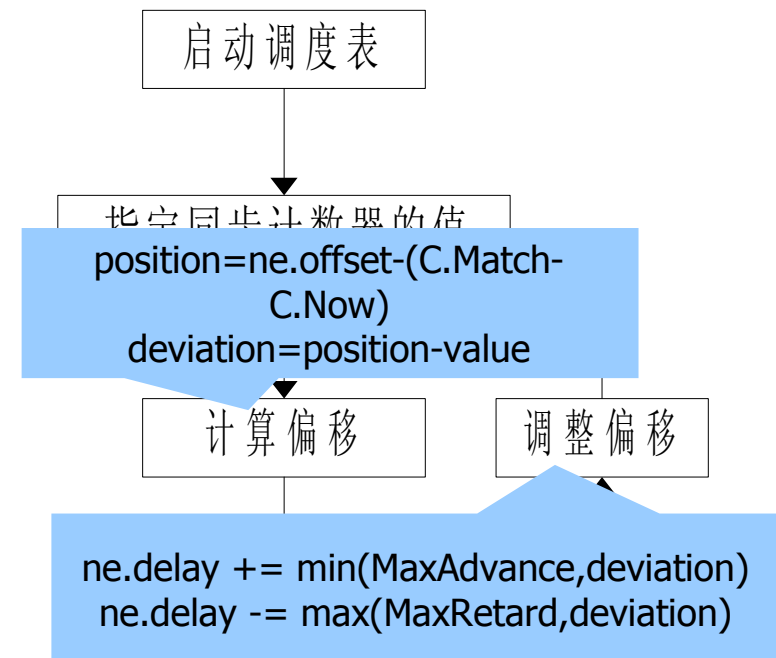
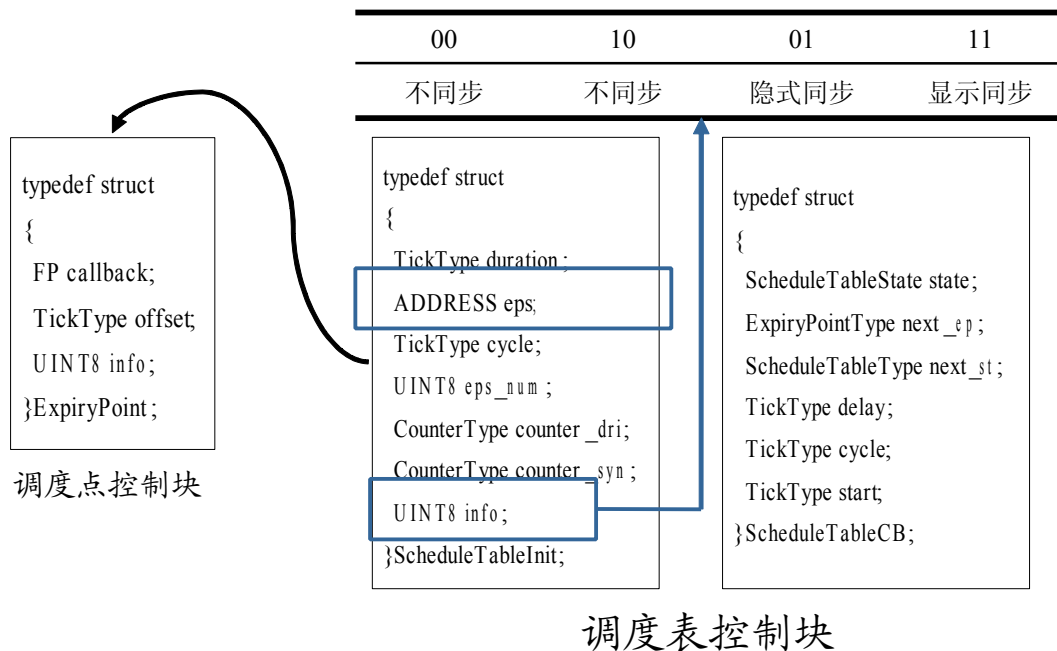




操作系统设计与实现——调度表管理

需求分析 > 设计实现 > 功能验证 > 性能分析

- 调度表和调度点控制块
- 三种同步类型的定义
- 显示同步关键算法





操作系统功能验证

需求分析 > 设计实现 > **功能验证** > 性能分析

- SkyEye模拟器
- 在运行时输出每一步的日志

```
Task1 is called
Task3 is called
Task1 is called again
Task2 is called
```

任务功能验证

优先级
Task3>Task1>Task2

```
Timer2 is called
Timer2 is called
Timer2 is called
Timer2 is called
```

```
Timer1 is called
Timer1 is called
Timer1 is called
Timer1 is called
```

```
Timer1 is called
Timer2 is called
Timer1 is called
```

中断嵌套验证

中断嵌套

Timer1为一类中断
Timer2为二类中断

```
Timer1 is called
Timer2 is called
Timer1 is called
Timer2 is called
Timer1 is called
Timer2 is called
Timer1 is called
Timer2 is called
Timer1 is called
Timer2 is called
Timer1 is called
Timer2 is called
Timer1 is called
Timer2 is called
Timer1 is called
Timer2 is called
Timer1 is called
Timer2 is called
Timer1 is called
Timer2 is called
```

中断验证

```
Task1 is waiting event
Task2 is setting event
Task1 is called after event set
Task2 is called
```

事件功能验证

Task1等待Task2
设置事件



操作系统功能验证

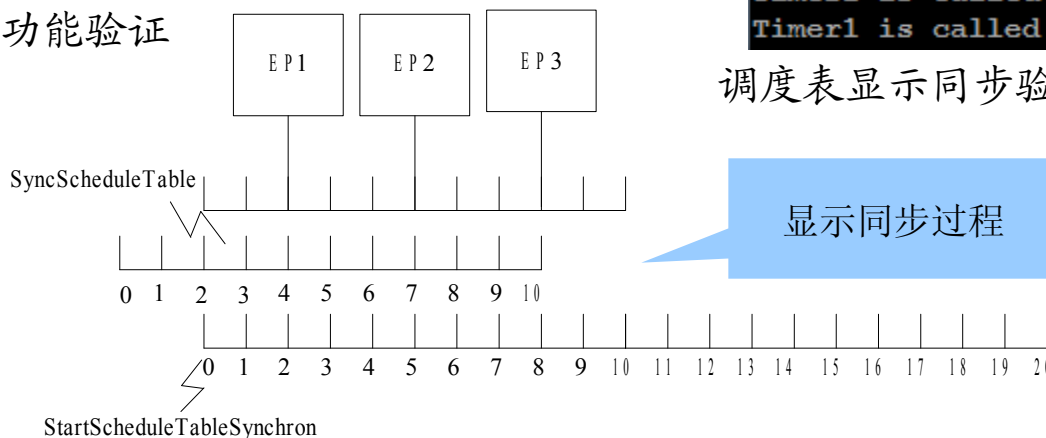
需求分析 > 设计实现 > 功能验证 > 性能分析

- SkyEye模拟器
- 在运行时输出每一步的日志

```
Task1 is called
Task1 gets resource
Task1 releases resource
Task2 is called
Task1 is called again
Timer1 is called
Timer1 is called
Timer1 is called
```

资源管理功能验证

优先级Task2>Task1,
Task1获得资源后激活
Task2



报警器在时间4激活Task1

```
Timer1 is called
Timer1 is called
First EP
Timer1 is called
Timer1 is called
Second EP
Timer1 is called
Timer1 is called
Third EP
Timer1 is called
Timer1 is called
```

调度表显示同步验证

显示同步过程

```
Timer1 is called
Timer1 is called
Timer1 is called
Timer1 is called
alarm1 is called
Task1 is called
Timer1 is called
Timer1 is called
Timer1 is called
Timer1 is called
Timer1 is called
Timer1 is called
Timer1 is called
Timer1 is called
Timer1 is called
Timer1 is called
Timer1 is called
alarm1 is called
Task1 is called
Timer1 is called
```

报警功能验证



操作系统性能分析

需求分析 > 设计实现 > 功能验证 > 性能分析

- CodeWarrior IDE
- 存储性能
- 实时性能

操作系统内核大小对比

操作系统	EMERALDS-OSEK	SmartOSEK	本论文	Trampoline	RTA-OSEK
内核大小	<5.5KB	<4.2KB	<2.7KB	<2.6KB	<1.6KB

操作系统对象所占RAM空间对比

	EMERALDS-OSEK	Trampoline	本论文
任务	36	21	28
警报	18	17	9
资源	2	10	3
计数器	8	14	10



操作系统性能分析

需求分析 > 设计实现 > 功能验证 > 性能分析

- CodeWarrior IDE
- 存储性能
- 实时性能

操作系统性能指标比较 (us)

	OSEKTurbo	RTA-OSEK	SmartOSEK	EMERALDS-OSEK	本论文
任务激活时间	4.4	6.2	-	-	8.1
任务切换时间	5.2	≈5	7.2	21-28	9.7
任务结束时间	4.4	4.65	-	-	7.8
中断延迟时间	2.9	1.1	2.4	-	2
中断响应时间	-	-	12.8	-	6.6
中断结束时间	4.0	-	6.65-13.65	-	5.7



总结与展望

□ 总结

- 设计并实现了一个兼容AUTOSAR OS规范的车用嵌入式实时操作系统内核
- 对操作系统内核的功能进行了验证，并对其存储需求和实时性能进行了对比分析

□ 展望

- 继续对操作系统的设计，使其完全符合AUTOSAR OS规范
- 需要继续对代码做优化，改进实时性能



谢谢！