

《嵌入式操作系统》实验设计与指导书

实验一：嵌入式操作系统调度机制设计与实现

■ 实验目的：

掌握嵌入式操作系统事件驱动调度机制的设计与实现方法。

■ 实验内容：

- 1) 假设有 6 个事件 E1, E2, E3, ……，E6,
- 2) 每个事件触发后需执行对应的任务：Task_1, Task_2, Task_3, ……，Task_6。其中，Task_i 的功能是：每 0.6 秒打印字符串 “Task_i is running\n”，连续打印的次数为 i 次。
- 3) 事件执行的优先级为 E1>E2>E3>……>E6。
- 4) 事件触发的时间点为：事件 E_i 在 main 函数执行后第 (7-i) 秒触发。每当一个事件触发时，打印字符串 “E_i is actived\n”。

■ 开发环境：

选用自己熟悉的开发环境，VC、Borland C、CodeBlocks 等。

■ 编程要求：

代码尽可能优化，避免重复，编程语句越少越好。

■ 提示：

调度系统构架设计可参考如下图所示。

编程需实现几个系统：

- 1) 事件集合触发操作编程。
- 2) 任务执行操作编程。
- 3) 标记位管理。
- 4) 调度器调度。

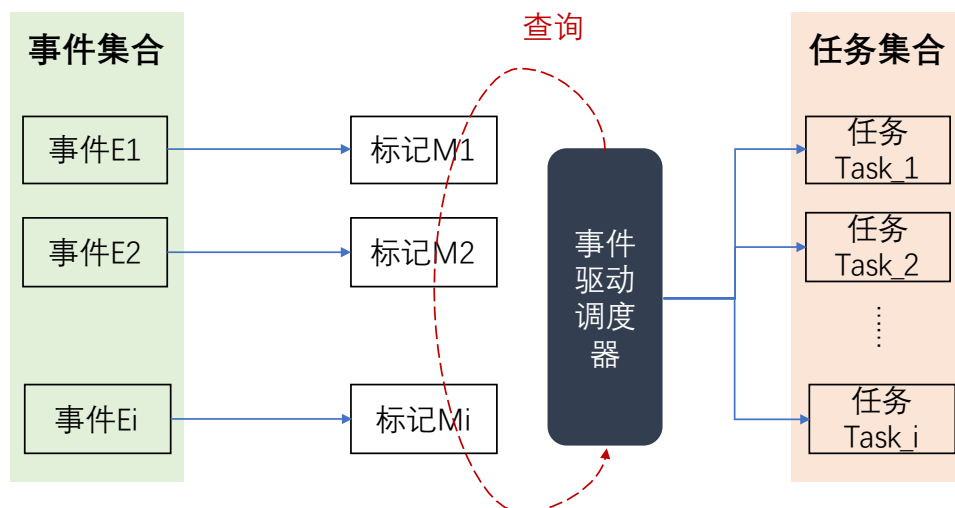


图 1. 实验一系统构架参考设计图

《嵌入式操作系统》实验设计与指导书

实验二：嵌入式操作系统动态内存管理机制设计与实现

■ 实验目的：

掌握嵌入式操作系统动态内存管理机制设计与实现方法。

■ 实验内容：

- 1) 假设堆空间大小为 900 字节（可定义数组 `char a[900]`，将数组 `a` 的内存空间作为堆栈空间，在数组 `a` 内部进行动态分配），采用固定大小块动态内存分配方法，将堆空间划分为二个分区。分区一大小为 300 字节，共含有 10 个块，每个块大小 30 字节。分区二大小为 600 字节，共含有 10 个块，每个块大小为 60 字节。
- 2) 空闲内存管理方法：采用链表管理方法。
- 3) 内存分配函数名：`OS_malloc`，内存释放函数名 `OS_free`。
- 4) 分区 1 测试程序如下：
 - 定时器每 2s 调用 1 次 `OS_malloc`，分配的内存大小分别为（字节）：18, 46, 16, 35, 13, 38, 22, 32, 25, 43, 20, 39, 11, 41。第 i 次分配内存块时，将分配的内存空间数值全部初始化为 i 。
 - 每个块分配后等待 3s 调用 `OS_free` 进行释放，每当一个块释放后，将释放的内存空间数值清除为 0。
 - 每分配一个块或释放一个块后，采用 `printf` 以“16 进制打印格式 (%02X)，按照每行打印 30 字节、打印 30 行方式”打印出整个 900 字节空间中数据信息至文件 `log.txt` 中，并根据 `log.txt` 中信息判断分配过程是否正确。

■ 开发环境：

选用自己熟悉的开发环境，VC、Borland C、CodeBlocks 等。

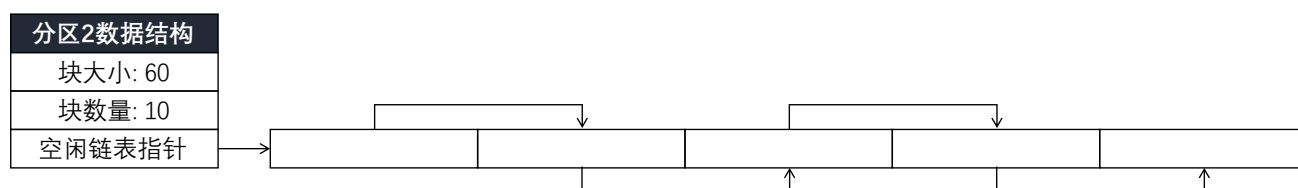
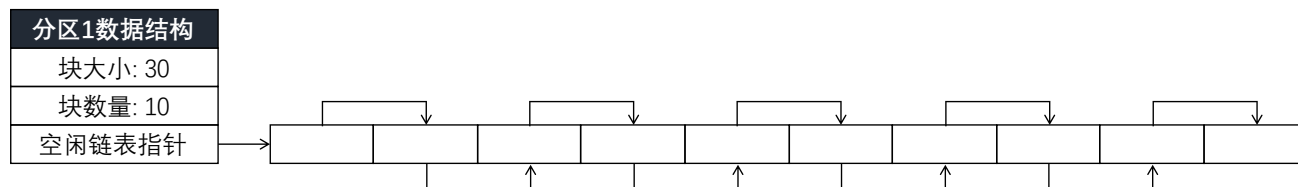
■ 编程要求：

代码尽可能优化，避免重复，编程语句越少越好。

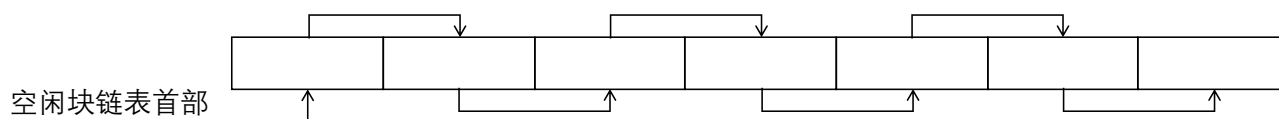
■ 提示：

分配操作原理：

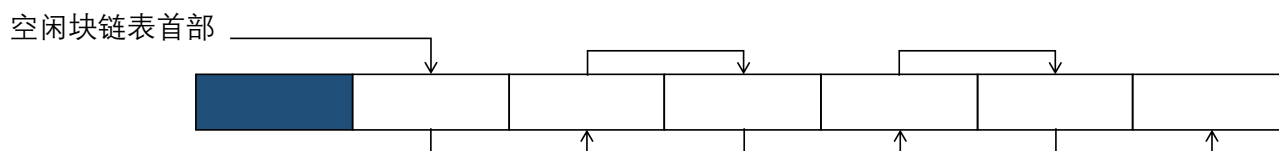
1) 分区方法：



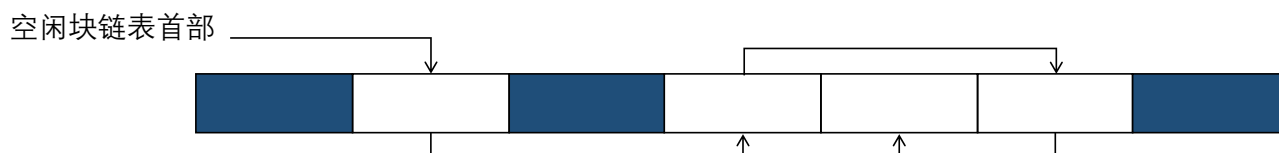
2) 初始化时，所有块均为空闲，故将所有块全部进行链接。



3) 由于所有块大小均相同，每次分配块时，直接分配空闲链表首部指向的块（如下图中蓝色部分），并将此块从链表中删除。第一次分配后，空闲链表指针链接情况如下图所示：



4) 当释放一个块时，将其插入空闲链表的首部或尾部。分配至一定程度后，链接结构将如下图所示（其中蓝色部分为已经分配的块）：



■ 数据结构参考：

/* 分区中每个块头部数据结构定义 */

```
typedef struct memblk {  
    struct memblk *next; /* next指针，用于链接所有的空闲块. */  
    partition_t *pt;      /* pt指针，指向块所属的分区. 当一个块释放时，将其释放至该分区中. */  
} memblk_t;
```

/* 分区数据结构定义. */

```
struct partition {  
    uint8_t blk_size;    /* 分区中块大小. */  
    uint8_t blk_num;     /* 分区中块数量. */  
    memblk_t *ptFreeQ;   /* 空闲块管理链表，此链表上链接所有的空闲块. */  
};
```

《嵌入式操作系统》实验设计与指导书

实验三：嵌入式操作系统消息传递机制设计与实现

■ 实验目的：

掌握嵌入式操作系统消息传递机制设计与实现方法。

■ 实验内容：

- 1) 创建 2 个线程以及一个消息队列，两个线程通过消息队列进行通信。
- 2) 一个线程每 2 秒向消息队列中投放一个消息，消息的内容依次为字符串 “msg_i \n”，其中 i 代表消息编号，连续发送 10 次消息。另一线程每隔 3 秒查看队列 1 次，若发现队列中存在消息，将其取出并 printf 打印。
- 3) 每产生一个消息时，利用实验二中的动态内存分配方法（也即 OS_malloc）分配消息所需占用的内存空间。当一个消息被取出并打印时，采用 OS_free 释放消息内存空间。

■ 开发环境：

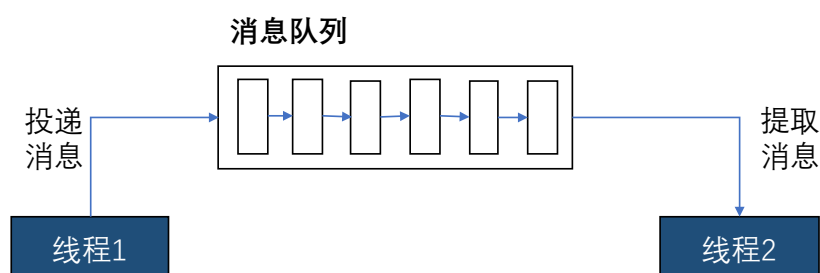
选用自己熟悉的开发环境，VC、Borland C、CodeBlocks 等。

■ 编程要求：

代码尽可能优化，避免重复，编程语句越少越好。

■ 提示：

- 1) windows 环境下可采用 pthread 创建 2 个线程。
- 2) 采用 OS_malloc 分配消息后，需用链表对所有消息进行链接以形成一个消息队列。
- 3) 可参考的系统结构设计如下图所示：



《嵌入式操作系统》实验设计与指导书

实验四：嵌入式操作系统跨平台移植方法

■ 实验目的：

掌握嵌入式操作系统跨平台移植方法。

■ 实验内容：

移植 uCOS 操作系统至 X86 台式机电脑上。

移植方法参考教材最后一章详细说明，以及网上参考资料。

移植成功后，创建 3 个线程，观察多线程相互切换抢占结果。