文章编号:1008-0570(2008)01-2-0090-02

一种新的嵌入式实时动态内存管理结构

A New Embedded Real-Time Dynamic Memory Management Structure

(西南科技大学)刘　邓　陈　波　刘婷婷
LIU DENG　CHEN BO　LIU TINGTING

摘要:内存资源是嵌入式操作系统中需要管理的重要资源之一。这种 O(1)时间复杂度的嵌入式实时动态内存管理结构采用页表结构组织系统内存信息,使得系统对内存的分配与回收成为实时操作。

关键词:内存管理;嵌入式系统
中图分类号:TP316.2　文献标识码:A

Abstract:The memory is one of the most important resources that embedded operating system should take into account. This embedded real-time dynamic memory management structure with average O(1) time complexity, which make use of a page table structure to organize the system memory information, have introduced a Real-Time operation for the operating system to allocate and free a memory resources.

Key Words:Memory Management, Embedded System

技术创新

1　引言

随着电子产业的不断发展,移动媒体和手持设备得到广泛的关注,这使得消费市场对此类设备的需求与日俱增。伴随嵌入式媒体需求的出现,设备所需的运算能力越来越强,内存需求也越来越大。此类设备往往配备大、中等规模的物理内存资源。在这些平台上,首次适应算法的遍历时间变得越来越长而破坏了系统的实时性约束;采用 bitmap 的内存管理结构受限于它静态的管理结构大小,使得 bitmap 结构的扩展性不能满足多样性的需求;采用伙伴算法时,系统的碎片随物理内存的增加而呈线形增加,这也导致碎片的组合操作成为一种昂贵的操作。所以需要提出一种适合中等规模内存大小的内存管理结构。

本文讨论的内存管理结构,通过引入一种页表结构进行物理内存管理信息的组织。这样在付出少许额外资源的情况下,使得系统对内存的操作即可满足实时性,又很大程度上限制碎片的产生。

2　内存管理的结构

在本文所讨论的内存管理算法中,引入了一种页表结构进行物理内存管理信息的组织。该页表结构模拟处理器中内存管理单元的结构,将空闲物理空间按照页面进行管理,从而提供一种较新的嵌入式内存分配方式。每个页面由其物理地址作为索引,唯一访问页表结构中的对应页表项。下文以页面大小 4K(页面大小可在操作系统系统产生时进行调节)为例,详细讨论该内存管理结构的细节。

2.1 页表项

页表项采用位域结构进行定义,具体内容如下代码所示:

```
typedef union mmpte {
struct {
unsigned pte_bktidx:4;
unsigned pte_ptetype:1;
unsigned pte_active:1;
unsigned pte_pagetype:1;
unsigned pte_leadercnt:CNT_SHIFT;
unsigned pte_nrslice:(25 - CNT_SHIFT - PTE_SHAREIDX);
unsigned pte_shareidx:PTE_SHAREIDX;
} bits;
uint32_t pte;
} mmpte_t;
```

在这个结构中具体定义了:内存页所属的桶(memory bucket)大小、页表项的类型、页表项当前是否活动、页表项所对应页面的类型、一次 malloc()操作可以取得的最大物理内存量、页表项对应页面中剩余的空闲内存桶数量、共享页面的共享计数。

内存桶的大小为 $2^n(3\leq n\leq 11)$ 字节,最小内存桶单元为 8 字节,最大内存桶单元为 2048 字节。每个内存桶单元为 2^n 字节的页面所拥有的内存桶数量为(页面大小/2^n)个。因此在一个 4K 页面中,最小内存桶单元的数量为 256 个;最大内存桶单元的数量为 2 个。对于大于 2048 字节的内存分配请求,内存管理系统直接分配一个空闲物理页面给请求者。在那些没有 MMU 的处理器上内存只可以顺序分配,此时内存管理系统可以通过 pte_leadercnt 域对内存进行管理,一次单独的 malloc()操作所能分配的最大长度为 $*(2^{pte\_leadercnt}-1)$。需要说明的是,malloc()操作的请求次数并没有限制。由于本内存管理结构采用页面作为管理单元,因此可以引入页面共享机制,最大页面共享计数为 $(2^{pte\_shareidx}-1)$。pte_active 表示页表项所对应的页面当前是否已被分配。pte_pagetype 表示当前页面是否是共享页面,如果是共享页面那么共享计数由 pte_shareidx 说明。如果出现大于 2048 字节的 malloc()请求,pte_ptetype 表示 malloc()分配得到的第一个页面是连续页面组的首页面,此时分配的页面数由 pte_leadercnt 说明。在嵌入式操作系统生成时,可以借由调整 CNT_SHIFT、PTE_SHAREIDX 等变量对页表项的参数进行调节,比如在不需

刘　邓:在读硕士研究生
本项目受四川省青年软件创新工程立项资助(2005AA0797)

要页面共享的系统中，可以将 PTE_SHAREIDX 调整为 1。

在嵌入式操作系统启动之时，系统计算剩余物理内存容量。以剩余容量为参数，计算页表结构所占用的存储空间。该空间的计算公式为:((物理内存- 内核使用内存) *(32/8))/(页面大小)字节。如果以物理内存为 8M，内核使用 1M(包括内核代码、数据、堆栈)，页面大小 4K 为例，得到的结果为 7K。此时系统需要初始化 7K 的额外空间来保存上述页表结构。同时系统将剩余的空闲物理内存按照页面大小为单位链接成一个双链表结构。每次系统取一个页面时只需返回链表中的第一个页面。

2.2 内存的分配

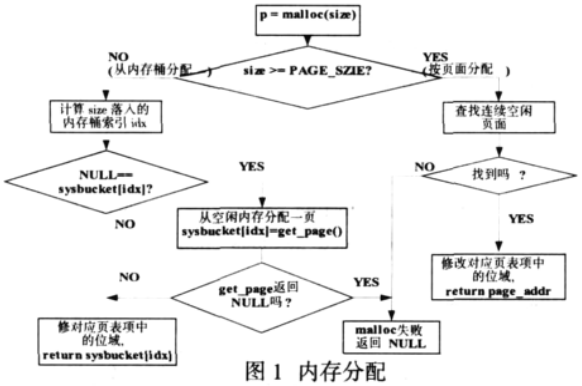

图 1 内存分配

上图 1 给出了具体的内存分配的流程图。sysbucket 是一个内核中的全局数组，原形为 mmbucket_t sysbucket[OS_CFG_BKT-SZSFT]; 该数组以内存桶的大小作为索引，指向第一个可用的空闲物理内存单元。内存分配的主要操作对象为 sysbucket[idx]。如果 sysbucket[idx]有可用内存，系统在页表项中记录相应信息，之后 malloc()返回所取得的内存地址；否则系统尝试在空闲内存堆中分配一个单独的物理内存页面然后再进行内存桶的分配。由于一个物理页面的分配只需要返回空闲内存链表中的第一个页面，因此对于 size 小于等于 PAGE_SIZE 的分配操作在最坏情况下有两步操作:O(1)的取得一个空闲物理页面，然后 O(1)的从新分配的物理页面中分配内存桶。整个过程的时间复杂度为 O(1)。如果 malloc()单次请求的尺寸大于 PAGE_SIZE，系统需要搜索整个空闲物理内存找到连续大小的一段空间，然后返回，该过程的复杂度为 O(n)。在实际的操作系统编制中，大块内存的分配所带来的性能损耗可以用保留内存区域的方式予以解决，使那些有实时性需求的大尺寸内存分配操作的时间复杂度为 O(1)。

2.3 内存的回收

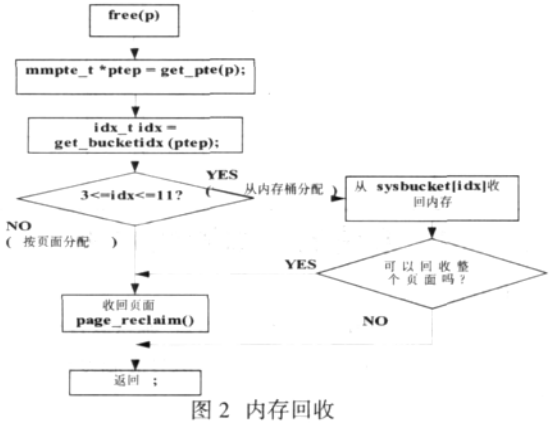

图 2 内存回收

上图 2 所示为内存回收操作的流程图。通过页面的物理地址，可以唯一索引页表结构中的一个页表项，get_pte()函数返回页面所对应的页表项。通过页表项 pte，可以取得该页的 pte_bktidx 信息。如果该页用作内存桶分配，那么 pte_bktidx 的范围为 3 到 11 之间，否则 pte_bktidx 大于 11 表示该页是作为分配尺寸大于 PAGE_SIZE 时予以整页分配的。如果 idx 在 3 到 11 之间，则 p 所指向的内存区域被插入到 sysbucket[idx]所指向的链表头。然后通过页表项中的 pte_nrslice 域计算能否收回整个页面。当 idx 不在 3 到 11 之间时，系统通过查询被回收页面页表项的 pte_leadercnt 域取得回收的内存尺寸。

在 page_reclaim()中，通过查询与回收页面左右相邻页面的页表项中的 pte_active 域可以知道此页面是否当前已经被分配，如果 pte_active 为 0，那么空闲页面可以被合并形成一个更大的可用内存区，否则被回收页面插入系统空闲内存的链表结构中。

2.3.1 对内存回收算法的进一步讨论

在内存的回收算法中，所有的信息都来自于页面所对应的页表项，由于页面与页表项的一一映射关系，该页表项可以在 O(1)时间内取得并进行实时计算。page_reclaim()中所需的左、右页面的页表项信息也可以在 O(1)时间内获得，然后进行实时计算。虽然 page_reclaim()也是一个 O(1)时间复杂度的算法，但是它却是一个相对费时的操作，在该项目的实际实现中不能达到理想的硬实时性约束，为此我们设计了一个低优先级的内核线程在系统空闲时异步的完成对内存页的批量回收（称之为 lazy reclaim）。如果系统对内存页面回收的硬实时性约束要求不高，那么上述的 page_reclaim()的同步回收已经可以很好的完成任务。

2.4 页面共享

物理页面的共享可以通过操作页表项中的 pte_pagetype 和 pte_shareidx 域来实现。pte_pagetype 表明对应页面是否在被共享，如果是，那么最多可共享计数为 $2^{pte\_shareidx}-1$ 次。

3 小结

本内存管理算法以页表作为管理结构，以页面作为管理基础，实现了对内存分配与回收的实时管理，同时提供了物理页面共享的一种方法。对于拥有大、中等规模物理内存的实时系统而言是一个十分理想的嵌入式操作系统组件。

创新点:本内存管理结构对于 4K 以下的内存分配为常数时间的实时分配。有研究表明，程序中 97%以上的内存请求集中在 128 字节- 2048 字节之间。而对那些 4K 以上的有实时要求的内存分配，也可以通过保留内存资源而予以保障。同时，本内存管理结构以不大的代价实现了进程间内存页面的有限共享。

参考文献:
[1]沈胜庆，嵌入式操作系统的内核研究[J]微计算机信息，2006，2- 2:72- 74
[2]杨雷、吴珏、陈汶滨，实时系统中动静结合的内存管理实现[J]微计算机信息，2005，10- 2:15- 16
[3] 曾非一、桑楠、熊光泽，嵌入式系统内存管理方案研究[J]. 2005，1:5- 7
[4]董庆丰、黄迪明，一种适用于嵌入式系统的动态内存管理技术[J]，2004，8:52- 54

作者简介:刘邓(1981-)，男，西南科技大学在读硕士研究生；主要研究方向:嵌入式系统;陈波(1965-)，男，西南科技大学教授；主要研究方向:嵌入式系统开发与应用;刘婷婷(1981-)，女，西南科技大学助理教师；主要研究方向:嵌入式系统，微系统设计。
Biography:Liu Deng (1981-), male, SiChuan, Southwest University of Science and Technology, Postgraduate Student. Research area: Embedded System.
(621010 四川绵阳 西南科技大学)刘邓 陈波 刘婷婷
通讯地址:(621010 四川 绵阳西南科技大学东苑 8 楼- A- 11 宿舍)刘邓

(收稿日期:2007.10.13)(修稿日期:2007.12.15)

技术创新