

CpuMemSets在Linux操作系统中的实现（2）PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/144/2021\\_2022\\_CpuMemSets\\_c103\\_144500.htm](https://www.100test.com/kao_ti2020/144/2021_2022_CpuMemSets_c103_144500.htm) 三、系统实现 在介绍CpuMemSets

在Linux-2.4.20中的具体实现之前，我们首先说明CpuMemSets涉及的几个基本概念，包括：处理器：指承载任务调度的物理处理器，但是不包括DMA设备、向量处理器等专用处理器；内存块：在SMP、UP系统中，所有的内存块与所有处理器的距离相等，因此不存在差别；但是在NUMA系统中，可以按照与处理器的距离对内存块划分等价类。此外

，CpuMemSets不考虑具有速度差异的特殊存储器，如输入输出设备缓存、帧缓存等。任务：一个任务，在任一时刻，或者等待事件、资源，或者被中断，或者在处理器上运行。虚拟存储区：内核为每个任务维护的多个虚拟地址区域，可为多个任务共享。位于虚拟存储区内的页，或者没有分配，或者已分配但被换出，或者已分配且在内存中。可以指定允许分配给某个虚拟存储区的内存块以及分配顺序。CpuMemSets为Linux提供了将系统服务和应用绑定在指定的处理器上进行调度、在指定的结点上分配内存的机制。CpuMemSets在已有的Linux调度和内存分配代码基础上增加了cpumemmap（cmm）和cpumemset（cms）两层结构，底层的cpumemmap层提供一个简单的映射对，实现系统的处理器号与应用的处理号、系统的内存块号与应用的内块号的映射。这种映射不一定是单射，一个系统号可以对应多个应用号。上层的cpumemset层负责说明允许把任务调度到哪些应用处理器号所对应的处理器上运行、可以从哪些应用内存块号所对应的

内存块中为相应的内核或虚拟存储区分配内存页，也就是说，指定可供内核、任务、虚拟存储区使用的资源集合。在这种两层结构中，资源的系统号供内核执行调度和内存分配时使用；而资源的应用号供用户进程指定本应用的资源集合时使用。系统号在启动期间全系统范围内有效，而应用号仅仅相对于共享同一个cmm的所有用户进程有效。而且，由于负载均衡和热插拔引发的资源物理编号的变化对应用号是不可见的。Linux的进程调度和内存分配在保持现有代码正常运转的基础上，添加了对CpuMemSets的支持，使用系统处理器号和系统内存块号以及其他数据结构如cpus\_allowed和mems\_allowed等实现资源的分区。此外，CpuMemSets的API提供了对cpuset、dplace、runon、psets、MPI、OpenMP、nodesets的支持，并且提供/proc接口以显示cmm和cms的结构、设置以及与任务、虚拟存储区、内核的连接关系、系统资源号和应用资源号等信息。下面我们分别对cpumemmap和cpumemset、进程调度和内存分配、以及API这三个部分进行详细分析：

### 3.1 cmm&cms

#### 3.1.1 数据结构

cpumemmap和cpumemset的数据结构如下所示，具体定义在include/linux/cpumemset.h中。Cpumemmap中的scpus和smems域分别指向一组系统处理器号和一组系统内存块号，实现应用的资源号（数组下标）与系统的资源号（数组元素值）的映射。Cpumemset中的acpus域指向一组应用处理器号，而amems域指向一组

100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)