

CpuMemSets在Linux操作系统中的实现 (4) 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_144493.htm 3.2 进程调度及内存分配 内核调

用0update_cpus_allowed(struct task_struct *p)根据任务的current_cms的处理器列表更改它的cpus_allowed位向量，从而影响该任务的处理器调度。 【kernel/cpumemset.c】 585

```
void 586 0update_cpus_allowed(struct task_struct *p) 587 { 588  
#ifdef CONFIG_SMP 589 int i. 590 cpumemset_t *cms =  
p->current_cms. 591 cpumask_t cpus_allowed =
```

```
CPU_MASK_NONE. 592 593 for (i = 0. i nr_cpus. i ) 594  
__set_bit(cms->cmm->cpus[cms->cpus[i]],
```

```
amp.cpu_online_map) 内核根据虚拟存储区
```

的vm_mems_allowed位向量为任务分配内存，如果是在中断上下文中，虚拟存储区的内存分配则依赖于kernel_cms的mems_allowed。

宏CHECK_MEMS_ALLOWED(mems_allowed, zone)负责检查zone所在的节点是否落在mems_allowed设定的内存块集合内。 【mm/memory.c】 1383

```
int handle_mm_fault(struct  
mm_struct *mm, struct vm_area_struct * vma, 1384 unsigned long  
address, int write_access) 1385 { : 1390 /* 1391 * We set the
```

```
mems_allowed field of the current task as 1392 * the one pointed by  
the faulting vma. The current 1393 * process will then use the correct  
mems_allowed mask 1394 * if a new page has to be allocated. 1395 */
```

```
1396 if(!in_interrupt()) 1397 current->mems_allowed =
```

```
vma->vm_mems_allowed. : 1417 } 【mm/page_alloc.c】 334
```

```
struct page * __alloc_pages(..) : 343 if (in_interrupt()) 344
  mems_allowed = kernel_cms->mems_allowed. 345 else 346
  mems_allowed = current->mems_allowed. 347 if (mems_allowed
  == 0) { 348 printk(KERN_DEBUG workaround zero
  mems_allowed in alloc_pages\\n). 349 mems_allowed = -1UL. 350 }
  : if(!CHECK_MEMS_ALLOWED(mems_allowed, z)) continue.
  : 450 } 【include/linux/cpumemset.h】 194 /* Used in
__alloc_pages() to see if we can allocate from a node */ 195 #define
CHECK_MEMS_ALLOWED(mems_allowed, zone) 196 ((1UL
zone_pgdat->node_id) & (mems_allowed))
```

为虚拟存储区分配页时，如果当前执行该任务的处理器包含在该虚拟存储区的cms中，则从该处理器的内存块列表中分配，否则从该虚拟存储区的cms定义的CMS_DEFAULT_CPU的内存块列表中分配。

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