


个人资料



darksun9972

访问：106459次

积分：1279

等级：BLOG等级4

排名：千里之外

原创：244篇

转载：46篇

译文：1篇

评论：7条

文章搜索

文章分类

编程之旅 (57)

linux和它的小伙伴 (50)

Emacs之怒 (87)

休息一会儿 (4)

读书笔记 (19)

时间管理 (11)

英文什么的最讨厌了 (42)

无主之地 (12)

SQL必须死 (3)

文章存档

2016年12月 (3)

2016年11月 (16)

2016年10月 (5)

2015年09月 (5)

2015年08月 (4)

展开

阅读排行

linux挂载samba文件系统 (2920)

linux时间与Windows时间 (2447)

LightDM配置说明 (1749)

如何把GIT仓库的子目录 (1680)

使用Emacs阅读邮件和新 (1624)

使用emacs的org-mode记 (1521)

emacs中的web浏览器-e (1208)

赠书 | AI专栏 (AI圣经！《深度学习》中文版) 评论送书 | 机器学习、Java虚拟机、微信开发

AIX线程支持的可调参数

2015-05-26 07:34 454人阅读 评论(0) 收藏 举报

分类：linux和它的小伙伴 (49)

目录(?)

转载至http://www.360doc.com/content/12/0525/16/8069262_213657224.shtml

线程支持可调参数

有很多可调的线程支持参数。

1. ACT_TIMEOUT

用

途：

值：

显

示：

更

改：

诊

断：

调

整：

调整激活超时的秒数。

缺省值：DEF_ACTOUT。范围：正整数。

echo\$ACT_TIMEOUT

此参数内部开启，因此初始缺省值无法通过 echo 命令看到。

ACT_TIMEOUT=export ACT_TIMEOUTexport SPINLOOPTIME

更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 ACT_TIMEOUT=n 命令添加到 /etc/environment 文件中可进行永久性更改。

N/A

N/A

参考：线程环境变量

2. AIXTHREAD_COND_DEBUG (AIX 4.3.3 及更高版本)

用

途：

值：

显

示：

更

改：

诊

断：

调

整：

维护条件变量列表以供调试器使用。

缺省值：ON。范围：ON，OFF。

echo\$AIXTHREAD_COND_DEBUG

此参数内部开启，因此初始缺省值无法通过 echo 命令看到。

AIXTHREAD_COND_DEBUG={ON|OFF}export AIXTHREAD_COND_DEBUG

更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 AIXTHREAD_COND_DEBUG={ON|OFF} 命令添加到 /etc/environment 文件中可进行永久性更改。

保留该变量设置为 ON 可使调试线程应用程序更容易，但可能会强加一些开销。

如果程序包含大量活动的条件变量，并且频繁创建和破坏条件变量，那么维护条件变量列表可能会产生更大的开销。将该变量设置为 OFF 将禁用该列表。

参考线程调试选项。

3. AIXTHREAD_ENRUSG

用

途：

启用或禁用 pthread 资源收集。

- DB2里面如何进行快速分 (1129)
- Purify使用指南 (1083)
- 使用emacs的org-mode记 (1046)

- 评论排行
- emacs中的web浏览器--e (2)
- 让ssh从stdin读取密码 (2)
- Emacs中读懂man文档的 (1)
- 使用Emacs删除重复行 (1)
- new-title (1)
- 14个酷毙了的Vim应用 (0)
- Ubuntu12.04禁用触摸板 (0)
- 为执行的命令设定超时 (0)
- 图形环境下的shell编程 (0)
- Ubuntu12.04安装磊科NV (0)

推荐文章

* CSDN日报20170725——《新的开始，从研究生到入职亚马逊》

* 深入剖析基于并发AQS的重入锁(ReentrantLock)及其Condition实现原理

* Android版本的"Wannacry"文件加密病毒样本分析(附带锁机)

今天与生活的对话(以图为例)



迷你仓



采光瓦



ZH131443: 18灯灯带, 布里有灯带的制功能时能多多介绍!

emacs中的web浏览器--eww darksun9972: @mr_chenyl:我也是在windows下使用, 版本是24.5, emacs的目录结构和libxm...

emacs中的web浏览器--eww mr_chenyl: 可以順便說一下, libxml2怎麼使用嗎? 我根據連接下載了然後對於壓縮包裡面的東西比較迷茫, 不知道怎...

值：	缺省值：OFF。范围：ON，OFF。
显示：	echo\$AIXTHREAD_ENRUSG 此参数内部开启，因此初始缺省值无法通过 echo 命令看到。
更改：	AIXTHREAD_ENRUSG={ON OFF}export AIXTHREAD_ENRUSG 更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 AIXTHREAD_ENRUSG={ON OFF} 命令添加到 /etc/environment 文件中可进行永久性更改。
诊断：	将此参数设置为 ON 将允许对一个进程中的所有 pthread 进行资源收集，但是会强加一些开销。
调整：	

参考[线程环境变量](#)。

4. AIXTHREAD_GUARDPAGES (AIX 4.3 及更高版本)

用途：	控制保护页数，加到 pthread 栈尾。
值：	缺省值：0。范围：正整数
显示：	echo\$AIXTHREAD_GUARDPAGES 此参数内部开启，因此初始缺省值无法通过 echo 命令看到。
更改：	AIXTHREAD_GUARDPAGES=nexport AIXTHREAD_GUARDPAGES 更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 AIXTHREAD_GUARDPAGES=n 命令添加到 /etc/environment 文件中可进行永久性更改。
诊断：	N/A
调整：	N/A

参考[线程环境变量](#)。

5. AIXTHREAD_MINKTHREADS (AIX 4.3 及更高版本)

用途：	控制应使用的最少内核线程数。
值：	缺省值：8。范围：正整数。
显示：	echo\$AIXTHREAD_MINKTHREADS 此参数内部开启，因此初始缺省值无法通过 echo 命令看到。
更改：	AIXTHREAD_MINKTHREADS=nexportAIXTHREAD_MINKTHREADS 更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 AIXTHREAD_MINKTHREADS =n 命令添加到 /etc/environment 文件中可进行永久性更改。
诊断：	N/A
调整：	库调度程序不会回收低于此变量中设置的值的内核线程。内核线程在几乎任何时候都可能被回收。一般来说，内核线程是作为 pthread 中止结果的目标。

参考：[进程范围争用作用域的变量](#)

6. AIXTHREAD_MNRRATIO (AIX 4.3 及更高版本)

用途：	控制库的比例因子。这个比率用于创建和终止 pthread。
值：	缺省值：8:1 范围：两个正数值 (p:k)，其中 k 是应该用来处理 p 变量中定义的可执行 pthread 数的内核线程数。
显示：	echo\$AIXTHREAD_MNRRATIO

示：	此参数内部开启，因此初始缺省值无法通过 echo 命令看到。
更改：	AIXTHREAD_MNRATIO=p:k export AIXTHREAD_MNRATIO 更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 AIXTHREAD_MNRATIO=p:k 命令添加到 /etc/environment 文件中可进行永久性更改。
诊断：	N/A
调整：	对于拥有极大量线程的应用程序可能很有用。但是，请总是测试 1:1 的比率，因为这可能提供更好的性能。
参考：	进程范围争用作用域的变量

7. AIXTHREAD_MUTEX_DEBUG (AIX 4.3.3 及更高版本)

用途：	维护活动互斥对象列表以供调试器使用。
值：	缺省值：OFF。可能的值：ON，OFF。
显示：	echo\$AIXTHREAD_MUTEX_DEBUG 示：此参数内部开启，因此初始缺省值无法通过 echo 命令看到。
更改：	AIXTHREAD_MUTEX_DEBUG={ON OFF} export AIXTHREAD_MUTEX_DEBUG 此更改立即生效，并且在用户从该 shell 中注销之前一直有效。将 AIXTHREAD_MUTEX_DEBUG={ON OFF} 命令添加到 /etc/environment 文件中可进行永久性更改。
诊断：	将该变量设置为 ON 可使调试线程应用程序更加容易，但可能会强加一些开销。
调整：	如果程序包含大量活动的互斥对象，并且频繁创建和破坏互斥对象，那么维护互斥对象列表可能会产生更大的开销。将该变量设置为 OFF 将禁用该列表。
参考：	线程调试选项

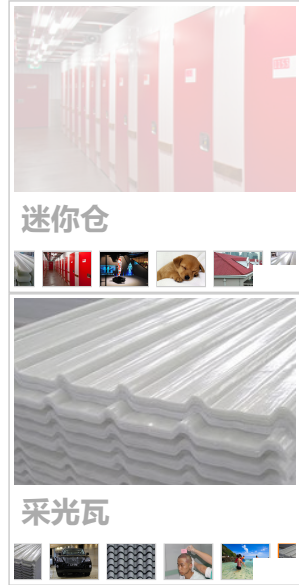
8. AIXTHREAD_MUTEX_FAST (AIX 5.2 及更高版本)

用途：	启用优化的互斥对象锁定机制。
值：	缺省值：OFF。可能的值：ON，OFF。
显示：	echo\$AIXTHREAD_MUTEX_FAST 示：此参数内部开启，因此初始缺省值无法通过 echo 命令看到。
更改：	AIXTHREAD_MUTEX_FAST={ON OFF} export AIXTHREAD_MUTEX_FAST 此更改立即生效，并且在用户从该 shell 中注销之前一直有效。将 AIXTHREAD_MUTEX_FAST={ON OFF} 命令添加到 /etc/environment 文件中可进行永久性更改。
诊断：	将该变量设置为 ON 将强制线程应用程序使用优化的互斥对象锁定机制，从而使性能提高。
调整：	如果程序由于严重的互斥对象争用而出现性能下降，那么将此变量设置为 ON 将强制 pthread 库使用仅对进程私有互斥对象起作用的优化互斥对象锁定机制。这些进程私有互斥对象必须使用 pthread_mutex_init 例程进行初始化，并且必须使用 pthread_mutex_destroy 例程破坏。
参考：	线程调试选项

9. AIXTHREAD_READ_GUARDPAGES (带有 5300-03 的 AIX 5.3 及更高版本)

用途：	控制对添加到 pthread 栈尾的保护页的读访问。
值：	缺省值：OFF。范围：ON，OFF。
显示：	echo\$AIXTHREAD_READ_GUARDPAGES 示：此参数内部开启，因此初始缺省值无法通过 echo 命令看到。





更改：	<code>AIXTHREAD_READ_GUARDPAGES={ON OFF}</code> <code>export AIXTHREAD_READ_GUARDPAGES</code> 更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 <code>AIXTHREAD_READ_GUARDPAGES={ON OFF}</code> 命令添加到 <code>/etc/environment</code> 文件中可进行永久性更改。
诊断：	N/A
调整：	N/A

参考[线程环境变量](#)。

10. AIXTHREAD_RWLOCK_DEBUG (AIX 4.3.3 及更高版本)

用途：	维护读写锁列表以供调试器使用。
值：	缺省值：ON。范围：ON , OFF。
显示：	<code>echo\$AIXTHREAD_RWLOCK_DEBUG</code> 此参数内部开启，因此初始缺省值无法通过 <code>echo</code> 命令看到。
更改：	<code>AIXTHREAD_RWLOCK_DEBUG={ON OFF}</code> <code>export AIXTHREAD_RWLOCK_DEBUG</code> 更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 <code>AIXTHREAD_RWLOCK_DEBUG={ON OFF}</code> 命令添加到 <code>/etc/environment</code> 文件中可进
诊断：	将此参数设置为 ON 可使调试线程应用程序更加容易，但可能会强加一些开销。
调整：	如果程序包含大量活动的读写锁，并且频繁创建和破坏读写锁，那么维护读写锁列表可能产生更大的开销。将该变量设置为 OFF 将禁用该列表。

参考：[线程调试选项](#)

11. AIXTHREAD_SUSPENDIBLE (带有 5300-03 的 AIX 5.3 及更高版本)

用途：	在将下列例程与 <code>pthread_suspend_np</code> 或 <code>pthread_suspend_others_np</code> 例程一起使用的应用程序中防止死锁： <ul style="list-style-type: none"><code>pthread_getrusage_np</code><code>pthread_cancel</code><code>pthread_detach</code><code>pthread_join</code><code>pthread_getunique_np</code><code>pthread_join_np</code><code>pthread_setschedparam</code><code>pthread_getschedparam</code><code>pthread_kill</code>
值：	缺省值：OFF。范围：ON , OFF。
显示：	<code>echo\$AIXTHREAD_SUSPENDIBLE</code> 此参数内部开启，因此初始缺省值无法通过 <code>echo</code> 命令看到。
更改：	<code>AIXTHREAD_SUSPENDIBLE={ON OFF}</code> <code>export AIXTHREAD_SUSPENDIBLE</code> 更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 <code>AIXTHREAD_SUSPENDIBLE={ON OFF}</code> 命令添加到 <code>/etc/environment</code> 文件中可进行永久性更改。
诊断：	此变量会带来少许性能损失。
调整：	仅在上述函数与 <code>pthread_suspend_np</code> routine 或 <code>pthread_suspend_others_np</code> 例程一起使用时才应启用此变量。

参考：[线程调试选项](#)

12. AIXTHREAD_SCOPE (AIX 4.3.1 及更高版本)

用途：	控制争用作用域。值为 P 表示基于进程的争用作用域 (M:N)。值为 S 表示基于系统的争用作用域 (1:1)。
值：	缺省值：S。可能的值：P 或 S。
显示：	echo\$AIXTHREAD_SCOPE 此参数内部开启，因此初始缺省值无法通过 echo 命令看到。
更改：	AIXTHREAD_SCOPE={P S}export AIXTHREAD_SCOPE 更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 AIXTHREAD_SCOPE={P S} 命令添加到 /etc/environment 文件中可进行永久性更改。
诊断：	如果分派的线程比期望的少，那么尝试系统作用域。
调整：	AIX 4.3.2 上的测试表明，某些应用程序使用基于系统的争用作用域 (S) 可能性能要好 环境变量只对那些创建时具有缺省属性的线程有影响。当 pthread_create 的 attr 参数采用缺省属性。

参考：[线程环境变量](#)

13. AIXTHREAD_SLPRATIO (AIX 4.3 及更高版本)

用途：	控制预留给睡眠线程的内核线程数。
值：	缺省值：1:12。范围：两个正数值 (k:p)，其中 k 是应预留给 p 个休眠 pthread 的内核线程数。
显示：	echo\$AIXTHREAD_SLPRATIO 此参数内部开启，因此初始缺省值无法通过 echo 命令看到。
更改：	AIXTHREAD_SLPRATIO=k:pexport AIXTHREAD_SLPRATIO 更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 AIXTHREAD_SLPRATIO=k:p 命令添加到 /etc/environment 文件可进行永久性更改。
诊断：	N/A
调整：	一般来说，支持睡眠线程需要的内核线程是很少的。因为它们一般是一次醒一个。这就节约了内核资源。

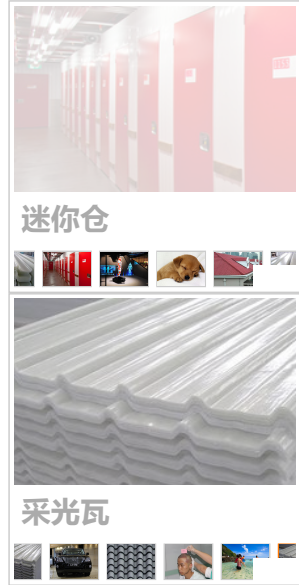
参考：[进程范围争用作用域的变量](#)

14. AIXTHREAD_STK=n (带有 4330-09 推荐维护包的 AIX 及更高版本)

用途：	十进制字节应该分配到每个 pthread。此值可由 pthread_attr_setstacksize 例程覆盖。
值：	缺省值：对于 32 位应用程序为 98 304 字节，对于 64 位应用程序为 196 608 字节。范围：0 到 268 435 455 的十进制整数，并且该值将向上取整到最近的页 (目前为 4 096)。
显示：	echo\$AIXTHREAD_STK 此参数内部开启，因此初始缺省值无法通过 echo 命令看到。
更改：	AIXTHREAD_STK=sizeexport AIXTHREAD_STK 更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 AIXTHREAD_STK=size 命令添加到 /etc/environment 文件中可进行永久性更改。
诊断：	如果失败程序的分析结果是由于堆栈溢出，缺省的栈大小应该增加。
调整：	如果尝试达到 32 位应用程序上的 32 000 线程限制，那么有必要减小缺省堆栈大小。

15. MALLOCBUCKETS (AIX 4.3.3 及更高版本)

用 在缺省内存分配器中启用基于存储区的扩展，对于发出大量较小分配请求的应用程序，该扩展可提高其性





途：	能。
值：	<div>MALLOCTYPE=buckets</div> <div>MALLOCBUCKETS=[<div>number_of_buckets:n bucket_sizing_factor:n blocks_per_bucket:n bucket_statistics:[stdout stderr pathname]],...</div>]</div> <div>下表显示了 MALLOCBUCKETS 的缺省值。</div> <div>MALLOCBUCKETS 选项</div> <div><div>缺省值</div><div>number_of_buckets¹</div><div>16</div><div>bucket_sizing_factor (32 位)²</div><div>32</div><div>bucket_sizing_factor (64 位)³</div><div>64</div><div>blocks_per_bucket</div><div>1024⁴</div></div> <div>注：</div> <div><div>1. 允许的最小值是 1，最大值是 128。</div><div>2. 对于 32 位的情况，bucket_sizing_factor 的专有值必须是 8 的倍数。</div><div>3. 对于 64 位的情况，bucket_sizing_factor 的专有值必须是 16 的倍数。</div><div>4. 选项 bucket_statistics 在缺省情况下被禁用。</div></div>
显示：	<div>echo\$MALLOCBUCKETS; echo\$MALLOCTYPE</div>
更改：	<div>使用用来导出环境变量的特定于 shell 的方法。</div>
诊断：	<div>如果 malloc 性能较差，且发出了很多较小的 malloc 请求，那么此功能可提高性能。</div>
调整：	<div>为了启用 malloc 存储区，环境变量 MALLOCTYPE 一定要设置为'bucket'。</div> <div>MALLOCBUCKETS 环境变量可用于更改 malloc 存储区的缺省配置，尽管缺省值对于大多数应用程序已经足够。</div> <div>number_of_buckets:n 选项可用于指定每个堆的可用存储区数，其中 n 为存储区的数量。指定的 n 值会用于所有有效的堆。</div> <div>bucket_sizing_factor:n 选项可用于指定存储区缩放大小因子，其中 n 为存储区缩放大小因子，单位为字节。</div> <div>blocks_per_bucket:n 选项可用于指定每个存储区中最初包含的块数，其中 n 为块数。这个值用于所有存储区。当所有块被分配出去后，这个 n 值也可在自动扩大存储区时用来确定每个存储区要增加多少块。</div> <div>选项 bucket_statistics 会使 malloc 子系统输出一个关于每个进程终止时调用 malloc 系统而 malloc 存储区正被启用的 malloc 存储区的统计摘要。这个摘要会显示存储区配置信息和处理每个存储区时分配内存的需要量。如果通过 malloc 很多堆被启用，每个存储区的分配需求量就是处理那个存储区所有堆的分配需求量之和。</div> <div>存储区统计摘要位于下列输出目标文件中，由选项 bucket_statistics 设定。</div> <div><div>stdout</div><div>标准输出</div><div>stderr</div><div>标准错误</div><div>pathname</div><div>用户指定路径名</div></div> <div>如果提供了用户指定路径名,统计输出会附加到已存在的文件内容后面。应当避免本应成为另一进程输入内容的某进程输出成为了以标准输出形式存在的终端输出。</div>
参考：	<div>Malloc Buckets</div>

16. MALLOCMULTIHEAP (AIX 4.3.1 及更高版本)

用 途：	在进程私有段内控制堆数
值：	缺省值：对于 AIX 4.3.1 和 AIX 4.3.2 为 16，对于 AIX 4.3.3 及更高版本为 32。范围：1 到 32 之间的正数。
显示：	echo\$MALLOCMULTIHEAP 此参数内部开启，因此初始缺省值无法通过 echo 命令看到。
更改：	MALLOCMULTIHEAP=[[heaps:n considersize],...] export MALLOCMULTIHEAP 更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 MALLOCMULTIHEAP=[[heaps:n considersize],...] 命令添加到 /etc/environment 文件中可进行永久性更改。
诊断：	在 malloc 锁（位于 F 段）中寻找锁争用或少于需要的可执行线程。
调整：	较小数量的堆数有助于减小进程大小。通过在启动应用程序之前导出 MALLOCMULTIHEAP 量，某些大量使用 malloc 子系统的多线程用户进程可获得更好的性能。 潜在的性能优化最可能出现在多线程 C++ 程序中，因为无论调用构造函数或析构函数这些都要用到 malloc 子系统。 在多线程用户进程运行于 SMP 系统时，任何性能优化都是可以得到的，尤其是在使用系（M:N ratio of 1:1）时。但是，在某些情况下，增强在其他条件下以及在单处理器上可能也很明显。 如果选项 considersize 被设定，交替堆选择算法将来选择有足够空间处理需求的有效 sbrk() 调用的数量，这可能使进程的工作集大小达到最小。但是这个算法需要更多一点的处理时间。

参考：线程环境变量

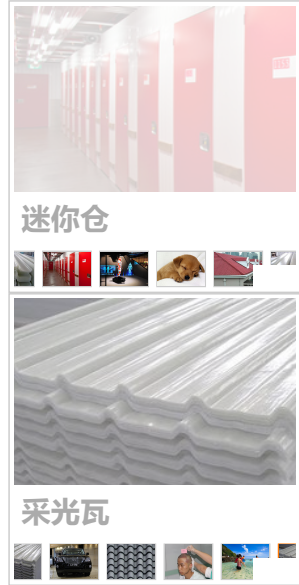
17. NUM_RUNQ

用 途：	更改运行队列的缺省数目。
值：	缺省值：在运行时找到的活动处理器的数量。范围：正整数。
显示：	echo\$NUM_RUNQ 此参数内部开启，因此初始缺省值无法通过 echo 命令看到。
更改：	NUM_RUNQ=export NUM_RUNQ 更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 NUM_RUNQ=n 命令添加到 /etc/environment 文件中可进行永久性更改。
诊断：	N/A
调整：	N/A

参考：线程环境变量

18. NUM_SPAREVP

用 途：	设置在 pth_init 期间将通过 malloc 分配的 vp 结构的数量。
值：	缺省值：NUM_SPARE_VP。范围：正整数。
显示：	echo\$NUM_SPAREVP 此参数内部开启，因此初始缺省值无法通过 echo 命令看到。
更改：	NUM_SPAREVP=export NUM_SPAREVP 更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 NUM_SPAREVP=n 命令添加到 /etc/environment 文件中可进行永久性更改。
诊断：	N/A





断：	
调	N/A
整：	

参考：[线程环境变量](#)

19. SPINLOOPTIME

用	在转到其他处理器（仅对于 libpthreads）前控制时间量来重试繁忙的锁。
途：	
值：	缺省值：单处理器为 1，多处理器为 40。范围：正整数。
显	echo\$SPINLOOPTIME
示：	此参数内部开启，因此初始缺省值无法通过 echo 命令看到。
更	SPINLOOPTIME=nexport SPINLOOPTIME
改：	更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 SPINLOOP' 到 /etc/environment 文件中可进行永久性更改。
诊	如果线程经常休眠（大量空闲时间），那么 SPINLOOPTIME 可能不够高。
断：	
调	如果存在 pthread 互斥对象争用，那么在多处理器系统上将该值从缺省值 40 进行增大可
整：	

参考：[线程环境变量](#)

20. STEP_TIME

用	调整在激活超时期间用来创建 VP 的次数。
途：	
值：	缺省值：DEF_STEPTIME。范围：正整数。
显	echo\$STEP_TIME
示：	此参数内部开启，因此初始缺省值无法通过 echo 命令看到。
更	STEP_TIME=export STEP_TIME
改：	更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 STEP_TIME=n 命令添加到 /etc/environment 文件可进行永久性更改。
诊	N/A
断：	
调	N/A
整：	

参考：[线程环境变量](#)

21. VP_STEALMAX

用	调整可窃取的 VP 的数量，或者关闭 VP 窃取。
途：	
值：	缺省值：无。范围：正整数。
显	echo\$VP_STEALMAX
示：	此参数内部开启，因此初始缺省值无法通过 echo 命令看到。
更	VP_STEALMAX=export VP_STEALMAX
改：	更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 VP_STEALMAX=n 命令添加到 /etc/environment 文件中可进行永久性更改。
诊	N/A
断：	
调	N/A
整：	

参考：线程环境变量

22. YIELDLOOPTIME

用	在忙锁（仅用于 libpthreads ）上分块前控制使用处理器的次数。假如有另一个优先级较高的可执行内核
途：	线程，处理器就会去处理那个内核线程。
值：	缺省值：0。范围：正数值。
显	echo\$YIELDLOOPTIME
示：	此参数内部开启，因此初始缺省值无法通过 echo 命令看到。
更	YIELDLOOPTIME=nexport YIELDLOOPTIME
改：	更改会在该 shell 中立即生效。在从这个 shell 注销之前更改一直有效。将 YIELDLOOPTIME=n 命令添加到 /etc/environment 文件中可进行永久性更改。
诊	如果线程经常休眠（大量空闲时间），说明 YIELDLOOPTIME 可能不够高。
断：	
调	如果不希望线程在等待锁时转入休眠状态，那么将该值从缺省值 0 进行增大可能有益。
整：	

参考：线程环境变量


顶 踩
0 0

上一篇 栈区，堆区，全局区，文字常量区，程序代码区 详解
下一篇 LINUX C中如何定义可变参数的宏


相关文章推荐

- AIX 网络参数调整命令
- AIX系统维护大全
- NFS 性能调优
- Java虚拟机参数
- 案例研究: 调优 WebSphere Application Server V7...

- Distributed System: MapReduce 可调参数与优化...
- AIX的vmstat命令详解(原创)
- web test LoadRunner docs / loadrunner license、...
- 优化 AIX 7 网络性能:NFS 监控和调优
- no 命令



迷你仓





采光瓦





茶杯犬多少钱



迷你仓



卖狗网



龟江湖



奥迪r8二手车



幼儿园滑梯



一室一厅出租

奥

猜你在找

- 【直播】机器学习&深度学习系统实战（唐宇迪）

【直播回放】深度学习基础与TensorFlow实践（王琛）

【直播】机器学习之凸优化（马博士）

【直播】机器学习之概率与统计推断（冒教授）

【直播】TensorFlow实战进阶（智亮）
- 【直播】Kaggle 神器：XGBoost 从基础到实战（冒教授）

【直播】计算机视觉原理及实战（屈教授）

【直播】机器学习之矩阵（黄博士）

【直播】机器学习之数学基础

【直播】深度学习30天系统实训（唐宇迪）

查看评论

暂无评论


您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

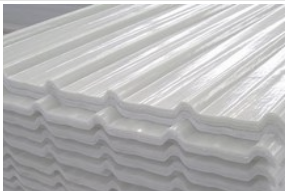
[网站客服](#) [杂志客服](#) [微博客服](#) webmaster@csdn.net 400-660-0108 | [北京创新乐知信息技术有限公司 版权所有](#) | [江苏知之为计算机有限公司](#) |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved 



迷你仓



采光瓦

