

Jessica要努力了。。

阿里巴巴2018届实习生招聘，有兴趣的可以把简历和岗位发我邮箱
1486657732@qq.com

博客园 首页 新随笔 联系 订阅 管理 随笔 - 942 文章 - 0 评论 - 63

昵称：Jessica程序猿
园龄：2年11个月
粉丝：93
关注：26
+加关注

<	2017年5月						>
日	一	二	三	四	五	六	
30	1	2	3	4	5	6	
7	8	9	10	11	12	13	
14	15	16	17	18	19	20	
21	22	23	24	25	26	27	
28	29	30	31	1	2	3	
4	5	6	7	8	9	10	

搜索

找找看

谷歌搜索

- 常用链接
- 我的随笔

我的评论

我的参与

最新评论

我的标签

- 随笔分类(955)
- C++(79)

C++ template(5)

C++ 容器(28)

C++构造函数(6)

C++面向对象编程(7)

C++训练(71)

C++重载与类型转换(9)

careercup(87)

Effective C++(3)

ext2文件系统系列(6)

Linux共享内存使用常见陷阱与分析

所谓共享内存就是使得多个进程可以访问同一块内存空间，是最快的可用IPC形式。是针对其他通信机制运行效率较低而设计的。往往与其它通信机制，如信号量结合使用，来达到进程间的同步及互斥。其他进程能把同一段共享内存段“连接到”他们自己的地址空间里去。所有进程都能访问共享内存中的地址。如果一个进程向这段共享内存写了数据，所做的改动会即时被有访问同一段共享内存的其他进程看到。共享内存的使用大大降低了在大规模数据处理过程中内存的消耗，但是共享内存的使用中有很多的陷阱，一不注意就很容易导致程序崩溃。

超过共享内存的大小限制？

在一个linux服务器上，共享内存的总体大小是有限制的，这个大小通过SHMMAX参数来定义（以字节为单位），您可以通过执行以下命令来确定 SHMMAX 的值：

```
1. # cat /proc/sys/kernel/shmmax
```

如果机器上创建的共享内存的总共大小超出了这个限制，在程序中使用标准错误perror可能会出现以下的信息：

```
1. unable to attach to shared memory
```

解决方法：

1、设置 SHMMAX

SHMMAX 的默认值是 32MB 。一般使用下列方法之一种将 SHMMAX 参数设为 2GB：

通过直接更改 /proc 文件系统，你无需重新启动机器就可以改变 SHMMAX 的默认设置。我使用的方法是将以下命令放入 />etc/rc.local 启动文件中：

```
1. # echo "2147483648" &gt; /proc/sys/kernel/shmmax
```

您还可以使用 sysctl 命令来更改 SHMMAX 的值：

```
1. # sysctl -w kernel.shmmax=2147483648
```

最后，通过将该内核参数插入到 /etc/sysctl.conf 启动文件中，您可以使这种更改永久有效：

```
1. # echo "kernel.shmmax=2147483648" &gt;&gt; /etc/sysctl.conf
```

2、设置 SHMMNI

我们现在来看 SHMMNI 参数。这个内核参数用于设置系统范围内共享内存段的最大数量。该参数的默认值是 4096 。这一数值已经足够，通常不需要更改。

您可以通过执行以下命令来确定 SHMMNI 的值：

```
1. # cat /proc/sys/kernel/shmmni
2. 4096
```

3、设置 SHMALL

最后，我们来看 SHMALL 共享内存内核参数。该参数控制着系统一次可以使用的共享内存总量（以页为单位）。简言之，该参数的值始终应该至少为：

```
1. ceil(SHMMAX/PAGE_SIZE)
```

SHMALL 的默认大小为 2097152 ，可以使用以下命令进行查询：

```
1. # cat /proc/sys/kernel/shmall
2. 2097152
```

SHMALL 的默认设置对于我们来说应该足够使用。

注意：在 i386 平台上 Red Hat Linux 的 页面大小 为 4096 字节。但是，您可以使用 bigpages ，它支持配置更大的内存页面尺寸。

多次进行shmat会出现什么问题？

flashcache(2)
gdb调试(2)
git(3)
hbase(1)
iscsi(4)
JAVA(5)
Leetcode(168)
linux内存管理(11)
linux内核(23)
Linux内核分析及编程(9)
Linux内核设计与实现(1)
maven
nginx(2)
python(1)
shell 编程(20)
Spring(1)
STL源码剖析(11)
UNIX 网络编程(24)
unix环境高级编程(24)
Vim(2)
web(9)
阿里中间件(1)
操作系统
测试(13)
程序员的自我修养(7)
大数据处理(3)
动态内存和智能指针(4)
泛型编程(2)
概率题(2)
论文中的算法(11)
面试(81)
软件安装(31)

当首次创建共享内存段时，它并不能被任何进程所访问。为了使共享内存区可以被访问，则必须通过 `shmat` 函数将其附加(`attach`)到自己的进程空间中，这样进程就与共享内存建立了连接。该函数声明在 `linux/shm.h`中：

```
1. #include
2. #include
3. void *shmat(int shmid, const void *shmaddr, int shmflg);
```

参数 `shmid` 是 `shmget()` 的返回值，是个标识符；
参数 `shmflg` 是存取权限标志；如果为 0，则不设置任何限制权限。在 `中`定义了几个权限：

```
1. #define SHM_RDONLY 010000 /* attach read-only else read-write */
2. #define SHM_RND 020000 /* round attach address to SHMLBA */
3. #define SHM_REMAP 040000 /* take-over region on attach */
```

如果指定 `SHM_RDONLY`，那么共享内存区只有读取权限。
参数 `shmaddr` 是共享内存的附加点，不同的取值有不同的含义：
?如果为空，则由内核选择一个空闲的内存区；如果非空，返回地址取决于调用者是否给 `shmflg` 参数指定 `SHM_RND` 值，如果没有指定，则共享内存区附加到由 `shmaddr` 指定的地址；否则附加地址为 `shmaddr` 向下舍入一个共享内存低端边界地址后的地址 (`SHMLBA`，一个常址)。
Ø通常将参数 `shmaddr` 设置为 `NULL`。

`shmat()` 调用成功后返回一个指向共享内存区的指针，使用该指针就可以访问共享内存区了，如果失败则返回 `-1`。

其映射关系如下图所示：

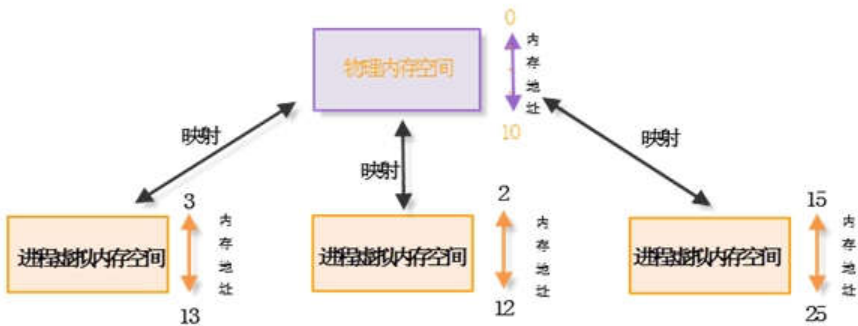


图1.1 共享内存映射图

其中，`shmaddr`表示的是物理内存空间映射到进程的虚拟内存空间时候，虚拟内存空间中该块内存的起始地址，在使用中，因为我们一般不清楚进程中哪些地址没有被占用，所以不好指定物理空间的内存要映射到本进程的虚拟内存地址，一般会让内核自己指定：

```
1. void ptr = shmat(shmid, NULL,0);
```

这样挂载一个共享内存如果是一次调用是没有问题的，但是一个进程是可以对同一个共享内存多次 `shmat`进行挂载的，物理内存是指向同一块，如果`shmaddr`为`NULL`，则每次返回的线性地址空间都不同。而且指向这块共享内存的引用计数会增加。也就是进程多块线性空间会指向同一块物理地址。这样，如果之前挂载过这块共享内存的进程的线性地址没有被`shmdt`掉，即申请的线性地址都没有释放，就会一直消耗进程的虚拟内存空间，很有可能会最后导致进程线性空间被使用完而导致下次`shmat`或者其他操作失败。

解决方法：

可以通过判断需要申请的共享内存指针是否为空来标识是否是第一次挂载共享内存，若是则使用进行挂载，若不是则退出。

```
1. void* ptr = NULL;
2. ...
3. if (NULL != ptr)
4. return;
5. ptr = shmat(shmid,ptr,0666);
```

附：

