

公司内部		
第	V1.0	版
共	7	页

网卡状态信息读取故障分析报告

文档编号： LINX_APP_TS_DS_V1.0

文档名称： 网卡状态信息读取故障分析报告

文档类别： 设计

密　　级： 公司内部

版本信息： V1.0

建立日期： 2010-11-01

创 建 人： 北京凝思科技有限公司研发部

审 核 人： 彭志航

批 准 人： 彭志航

批准日期：

保 管 人： 高严

存放位置： svn://

编辑软件： OpenOffice 3.2.0

文档修订记录

VER	REV	修订人	修订日期	简要说明	审核人	批准日期
V1.0	C	李瑾	2010-11-01	初次创建	彭志航	
VER（版本编号）：V——版本编号；R——修订编号。						
REV（修订状态）：C——创建；A——增加；M——修改；D——删除。						

文档审批记录

版本号	批准人	角色	审批日期	签字	备注
V1.0	彭志航				

目　录

1 故障出现环境.....1
2 故障还原.....1
3 故障产生原因.....2
4 解决方法.....2
5 验证.....3
6 压力测试.....3

表格

插图

1 故障出现环境

硬件环境：一台三网卡计算机

操作系统版本：凝思磐石安全操作系统 Rocky4.2-x86_64

应用程序名称：网卡监控程序 V1.2

出现现象：当 Daemon 程序“sys_nicmonitor”运行一段时间后，客户端提取网卡信息函数“get_nic_info”提取网卡信息失败。

日志输出：

```
2010-10-09 11:34:05.159 get_nic_info NOTICE: No information of bond0 !
2010-10-09 11:34:05.159 get_nic_info NOTICE: No information of eth0 !
2010-10-09 11:34:05.159 get_nic_info NOTICE: No information of eth2 !
2010-10-09 11:57:30.515 sys_nicmonitor NOTICE: bond0 flow is abnormal !
2010-10-09 11:57:30.529 sys_nicmonitor NOTICE: eth0 flow is abnormal !
2010-10-09 11:57:31.530 sys_nicmonitor NOTICE: bond0 flow is normal now !
2010-10-09 11:57:31.531 sys_nicmonitor NOTICE: eth0 flow is normal now !
2010-10-09 13:13:42.896 sys_nicmonitor NOTICE: bond0 flow is abnormal !
2010-10-09 13:13:42.896 sys_nicmonitor NOTICE: eth0 flow is abnormal !
2010-10-09 13:13:43.896 sys_nicmonitor NOTICE: bond0 flow is normal now !
2010-10-09 13:13:43.896 sys_nicmonitor NOTICE: eth0 flow is normal now !
2010-10-09 14:00:33.401 get_nic_info NOTICE: No information of bond0 !
2010-10-09 14:14:09.475 get_nic_info NOTICE: No information of bond0 !
```

2 故障还原

1、在裸机上安装操作系统凝思磐石安全操作系统 Rocky4.2-x86_64。

2、安装网卡监控程序 V1.2（步骤参考《网卡监控程序使用手册》）。

3、将三个网卡设置为连接状态，绑定网卡 eth0 和 eth3 为 bond0。

4、启动 Daemon 程序“sys_nicmonitor”

```
#./sys_nicmonitor
```

5、启动测试程序“getmem”。

```
#./getmem
```

6、大约 20 分钟未出现异常，判断出现事件为小概率事件。修改配置文件

```
#vi /conf/nic/sys_netcard_conf.txt
```

修改 write_interval 为 1 默认为 10（秒），修改应用程序每隔 1 秒读一次。

7、重新启动 Daemon 程序“sys_nicmonitor”和测试程序，不到 8 分钟故障出现。

3 故障产生原因

当多个进程之间共享内存的时候会出现资源竞争的问题。在程序中使用到信号灯来控制。第一个得到信号灯的进程同时也改变信号灯值，这样，系统调时包含了两个步骤：得到信号灯；改变信号灯值。由此可能导致一种竞争状态：第一个得到信号灯的进程在得到信号灯时，第二个进程又调用，并且发现信号灯已经存在，此时，第二个进程必须具有判断是否有进程正在对信号灯值是否改变的能力。

4 解决方法

在得到信号灯时我们加入了两个锁来控制信号灯，这样出现的情况有一下几种：

1、lock[0] = 0，信号灯为 0（熄灭）。这时表明没有进程访问共享内存。

2、当有进程访问时 lock[0] = 0，信号灯为为 0（熄灭）进程可改变信号灯值加 1（即点亮）lock[1] = 1 表明共享内存有人在访问。此时如果有进程访问时先判断 lock[0] = 0，

然后在判断信号灯是否为1，如果为1则继续等待，直到信号灯值为0。

3、 lock[0] = 0，信号灯为1。对共享内存进行操作，操作完成恢复信号灯为0。

这样如果两个进程同时得到信号灯时，再判断是否有进程对其操作，有进程操作则继续等待。

5 验证

修改代码（见源码）后，将新版本（V1.4）重新安装。启动Daemon每隔一秒钟向共享内存写一次，应用程序每隔一秒读一次，程序运行一天未出现BUG。

6 压力测试

同时启动三个应用程序每隔1毫秒从共享内存中读一次，测试一天未见异常。