

Oracle中Killsession的研究 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/143/2021\\_2022\\_Oracle\\_E4\\_B8\\_ADK\\_c102\\_143163.htm](https://www.100test.com/kao_ti2020/143/2021_2022_Oracle_E4_B8_ADK_c102_143163.htm) 我们知道,在Oracle数据库中,可以通过kill session的方式来终止一个进程,其基本语法结构为: alter system kill session sid,serial# . 被kill掉的session,状态会被标记为killed,Oracle会在该用户下一次touch时清除该进程. 我们发现当一个session被kill掉以后,该session的paddr被修改,如果有多个session被kill,那么多个session的paddr都被更改为相同的进程地址:SQL>

```
0select saddr,sid,serial#,paddr,username,status
from v$session where username is not null.SADDR SID SERIAL#
PADDR USERNAME STATUS-----
```

```
-----542E0E6C 11 314
542B70E8 EYGLE INACTIVE542E5044 18 662 542B6D38 SYS
ACTIVE
```

```
SQL> alter system kill session 11,314.System altered.SQL>
0select saddr,sid,serial#,paddr,username,status from v$session where
username is not null.SADDR SID SERIAL# PADDR USERNAME
STATUS-----
```

```
-----542E0E6C 11 314 542D6BD4
EYGLE KILLED542E5044 18 662 542B6D38 SYS ACTIVE
```

```
SQL> alter system kill session 11,314.System altered.SQL>
0select saddr,sid,serial#,paddr,username,status from v$session where
username is not null.SADDR SID SERIAL# PADDR USERNAME
STATUS-----
```

```
-----542E0E6C 11 314 542D6BD4
EYGLE KILLED542E2AA4 14 397 542B7498 EQSP
INACTIVE542E5044 18 662 542B6D38 SYS ACTIVE
```

```
SQL> alter
system kill session 11,314.System altered.SQL>
0select saddr,sid,serial#,paddr,username,status from v$session where
username is not null.SADDR SID SERIAL# PADDR USERNAME
STATUS-----
```

```
-----542E0E6C 11 314 542D6BD4
EYGLE KILLED542E2AA4 14 397 542B7498 EQSP
INACTIVE542E5044 18 662 542B6D38 SYS ACTIVE
```

```
SQL> alter
```

```
system kill session 14,397.System altered.SQL> 0select
saddr,sid,serial#,paddr,username,status from v$session where
username is not null.SADDR SID SERIAL# PADDR USERNAME
STATUS-----
```

```
-----542E0E6C 11 314 542D6BD4
EYGLE KILLED542E2AA4 14 397 542D6BD4 EQSP
```

KILLED542E5044 18 662 542B6D38 SYS ACTIVE 在这种情况下，很多时候，资源是无法释放的,我们需要查询spid，在操作系统级来kill这些进程.但是由于此时v\$session.paddr已经改变，我们无法通过v\$session和v\$process关联来获得spid那还可以怎么办呢? 我们来看一下下面的查询: SQL> SELECT

```
s.username,s.status, 2
x.ADDR,x.KSLLAPSC,x.KSLLAPSN,x.KSLLASPO,x.KSLLID1R,x.
KSLLRTYP, 3 decode(bitand (x.ksuprflg,2),0,null,1) 4 FROM
x$ksupr x,v$session s 5 WHERE s.paddr( )=x.addr 6 and
bitand(ksspaflg,1)!=0.USERNAME STATUS ADDR KSLLAPSC
KSLLAPSN KSLLASPO KSLLID1R KS
```

```
D-----
----- -- - 542B44A8 0 0 0 ACTIVE 542B4858 1 14
24069 0 1 ACTIVE 542B4C08 26 16 15901 0 1 ACTIVE 542B4FB8 7
46 24083 0 1 ACTIVE 542B5368 12 15 24081 0 1 ACTIVE 542B5718
15 46 24083 0 1 ACTIVE 542B5AC8 79 4 15923 0 1 ACTIVE
542B5E78 50 16 24085 0 1 ACTIVE 542B6228 754 15 24081 0 1
ACTIVE 542B65D8 1 14 24069 0 1 ACTIVE 542B6988 2 30 14571 0
1USERNAME STATUS ADDR KSLLAPSC KSLLAPSN
KSLLASPO KSLLID1R KS D-----
```

----- -- -SYS ACTIVE  
542B6D38 2 8 24071 0 542B70E8 1 15 24081 195 EV 542B7498 1 15  
24081 195 EVSYS INACTIVE 542B7848 0 0 0SYS INACTIVE  
542B7BF8 1 15 24081 195 EV16 rows 0selected. 我们注意,红字标  
出的部分就是被Kill掉的进程的进程地址.简化一点,其实就是  
如下概念:SQL> 0select p.addr from v\$process p where pid 1 2  
minus 3 0select s.paddr from v\$session  
s.ADDR-----542B70E8542B7498 Ok , 现在我们获得了进程地  
址 , 就可以在v\$process中找到spid , 然后可以使用Kill或  
者orakill在系统级来杀掉这些进程.实际上,我猜测: 当在Oracle  
中kill session以后, Oracle只是简单的把相关session的paddr 指向  
同一个虚拟地址.此时v\$process和v\$session失去关联,进程就此  
中断.然后Oracle就等待PMON去清除这些Session.所以通常等  
待一个被标记为Killed的Session退出需要花费很长的时间. 如果  
此时被Kill的process , 重新尝试执行任务 , 那么马上会收到进  
程中断的提示 , process退出 , 此时Oracle会立即启动PMON来  
清除该session.这被作为一次异常中断处理. 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)