

UNIX和WINDOWS2000上的ORACLE的差异 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/285/2021_2022_UNIX_E5_92_8CWIN_c67_285514.htm 摘要 ORACLE是广为人知的Unix硬件平台上的领先的数据库系统。ORACLE用户和管理员因此熟悉Unix平台上的ORACLE架构以及它上面的工具和技巧，并从他们的数据库得到最大的收益。相反，Windows上的ORACLE架构就不那么的被广为了解。这篇文章从一个DBA的角度考察了两个操作系统之间的关键的异同点。简介在看了几本令人失望的这方面的书之后，我们写了这篇文章。那些书的通病是试图做太多的事情 - - 在细节上讲述Windows和ORACLE。我们的这篇文章假定读者熟悉Unix平台上的ORACLE DBA的工作。因此本文将分析两个平台上的ORACLE的关键的差异而不是从头教你ORACLE的技巧。我们不想把它作为你的一份详尽的指导或者是手册的替代品，事实上它可能鼓励你阅读一些手册。作为数据库服务器平台，它只会涉及一些Unix和Windows上相关的优点，这就是本文的目的。范例 这个例子使用linux上的ORACLE 8i，实例名字叫作eighti。Windows 2000上面的ORACLE 8i的实例名字叫作atei。客户端对ORACLE的访问 当客户端连接到ORACLE时，通常的来说ORACLE服务器的平台与客户端的应用无关。这实际上很难说清。ORACLE DBA 和系统管理人员更关心操作系统平台，他们有的时候会基于需求（如运行时间和可扩展性）选择平台。更通常的情况下,他们接受（或是接手）给定的平台并学习从中得到最大受益。关于WINDOWS 2000值得一提的是Windows 2000是从Windows NT升级而来。在这

两个操作系统之间有很多的相似点，Windows 2000 也有些新的特性。微软从NT4.0的升级途径见下表。两个系统间有很多相似点：

----- NT 4.0 Windows 2000

----- NT 4.0 Workstation Windows 2000 Professional NT
4.0 Server Windows 2000 Server NT 4.0 Enterprise Edition
Windows 2000 Advanced server Unix Windows 2000 Datacenter
server

----- ORACLE后台进程 下面这句话对于用过ORACLE的人来说是会很熟悉的：“每一个运行着的ORACLE数据库都对应一个ORACLE实例，当一个数据库在数据库服务器（不考虑机器的类型）上启动的时候，ORACLE分配一块叫做System Global Area (SGA)的内存区域并启动一个或者多个ORACLE进程。SGA和ORACLE进程合起来称作ORACLE实例。”——摘自 ORACLE 8i Concepts [4 L Leverenz, 1999]。处理后台进程是放在首位的，也是不同的操作系统之间最明显的差异。ORACLE在UNIX上的后台进程 任何连接到UNIX的用户都可以很容易的察看ORACLE的后台进程：
% ps -ef|grep
eighti|grep -v grep oracle8 18451 1 0 16:37:18 ? 0:00
ora_pmon_eighti oracle8 18453 1 0 16:37:19 ? 0:00 ora_dbw0_eighti
oracle8 18457 1 0 16:37:19 ? 0:04 ora_ckpt_eighti oracle8 18461 1 0
16:37:19 ? 0:00 ora_reco_eighti oracle8 18455 1 0 16:37:19 ? 0:02
ora_lgwr_eighti oracle8 18459 1 0 16:37:19 ? 0:01 ora_smon_eighti

```
oracle8 19168 19167 0 16:43:46 ? 0:00 oracleeighti
(DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq
))) 最后一行的ORACLE进程与一个SQL*Plus会话相关，其他的
进程都是后台进程。在ORACLE中我们可以通过输
入SQL*Plus会话察看这些进程：SELECT sid, spid, osuser,
s.program FROM v$process p, v$session s WHERE p.addr=s.paddr
; SID SPID OSUSER PROGRAM
```

```
----- 1
18451 oracle8 oracle@saic02 (PMON) 2 18453 oracle8
oracle@saic02 (DBW0) 3 18455 oracle8 oracle@saic02 (LGWR) 4
18457 oracle8 oracle@saic02 (CKPT) 5 18459 oracle8 oracle@saic02
(SMON) 6 18461 oracle8 oracle@saic02 (RECO) 7 19168 oracle8
sqlplus@saic02(TNS V1-V3) 7 rows 0selected. 每一个后台进程都
有一行，还有一行信息是与SQL*Plus会话相关的，SPID对应
相应的UNIX进程号。在WINDOWS2000上的ORACLE后台进
程回到WINDOWS上，从操作系统中察看后台进程有些困难
。从任务管理器中可能会看到运行着的应用（任务管理器的
察看方法：在任务栏点击右键选择"任务管理器"）。在服务器
上ORACLE可以是可用的，运行着的应用却是不可见的。
进程表的确显示一个进程叫做ORACLE.EXE，察看alert log 显示
ORACLE的所有后台进程都是启动的：PMON started with
pid=2 DBW0 started with pid=3 LGWR started with pid=4 CKPT
started with pid=5 SMON started with pid=6 RECO started with
pid=7 要看实际的后台进程，需要运行额外的软件，例如，进
程察看器。该软件可以从Windows 2000 CD中得到（Windows
NT的话可以从资源包中得到）。在Windows 2000上
```

，ORACLE实例是作为一个单一的Windows 2000进程（ORACLE.EXE）实现的。这个进程包括实例所需要实现的每个任务的线程。因此一个线程对应每个ORACLE后台进程。ORACLE.EXE进程作为一个服务运行，可以从控制面板的服务中察看到ORACLEServiceSID。其他的服 务也可以这样控制。这允许ORACLE在没有用户登录服务器的时候也持续的运行。对于共享主处理器资源的所有的进程来说，ORACLE能够达到高速、低负荷的上下文切换。在Unix下显示ORACLE中的进程，我们也可以通过输入简单的SQL语句来达到。为了显示PID列，SQL语句做了些轻微的改动。要注意PID匹配警告日志中报告的值。

```

SELECT s.sid, p.pid, p.spid
       signaled, s.osuser, s.program
FROM v$process p, v$session s
WHERE p.addr=s.paddr.
SID PID THREADID OSUSER
PROGRAM -----
1 2 1088 SYSTEM ORACLE.EXE 2 3 1172 SYSTEM ORACLE.EXE
3 4 1180 SYSTEM ORACLE.EXE 4 5 1192 SYSTEM ORACLE.EXE
5 6 1212 SYSTEM ORACLE.EXE 6 7 1220 SYSTEM ORACLE.EXE
7 8 1200 Administrator SQLPLUSW.EXE 7 rows 0selected.
每一个后台进程都有一行，还有一行信息是与SQL*Plus会话相关。
程序名字并没有指明后台进程的名字，和在Unix中一样，这些名字可以通过和v$bgprocess连接得到。
SELECT s.sid SID,
       p.spid THREADID, p.program PROCESSNAME, bg.name NAME
FROM v$process p, v$session s, v$bgprocess bg
WHERE p.addr = s.paddr AND p.addr = bg.paddr AND bg.paddr = '00'.
SID
THREADID PROCESSNAME NAME -----
----- 1 1088 ORACLE.EXE PMON 2 1172

```

ORACLE.EXE DBW0 3 1180 ORACLE.EXE LGWR 4 1192

ORACLE.EXE CKPT 5 1212 ORACLE.EXE SMON 6 1220

ORACLE.EXE RECO 6 rows 0selected. 断开会话 提交SQL命令
ALTER SYSTEM DISCONNECT SESSION可以断开会话。有的

时候需要在操作系统级别断开会话，在UNIX上，通过kill命令实现，前面例子中的SQL会话可以通过输入UNIX命令断开

： kill -9 19168 在Windows 2000上可以用orakill断开一个会话。orakill是Windows平台上的ORACLE的一个特定命令，默认安装在\$ORACLE_HOME\bin下。在命令行下输入orakill可以察看它的用法。前面例子中的SQL*Plus会话可以通过输入如下的命令断开：

orakill atei 1200 Kill of thread id 1200 in instance atei successfully signaled. 在Windows 2000中，如果一个断开的会话标记为 "marked for kill"但是没被删除，orakill会终止它。

要记住杀掉一个后台进程总不是个好主意，尤其是Windows上，会导致进程崩溃，甚至导致数据库不可用。Windows 2000注册表和其他的Windows 2000中的应用那样，ORACLE的大多数的设定都在注册表中。应该看

看HKEY_LOCAL_MACHINE SOFTWARE\ORACLE下面都有什么。这些参数中的一些在后面会详细讨论。和ORACLE服务相关的参数和其他的服务一样存贮在同样的位置：

HKEY_LOCAL_MACHINE SYSTEM\CurrentControlSet services。环境变量 在Unix中两个最重要的变量是ORACLE_HOME和ORACLE_SID。一旦这些变量设定的话，应用就可以运行并连接到本地数据库。通常也把\$ORACLE_HOME/bin包含在\$PATH中以便在使用ORACLE工具（如：sqlplus）的时候免去输入全路径的麻烦。Windows 2000中可以打开命令行设

定ORACLE_SID 变量再联接到本地数据库。其他的值可以从注册表中得到。 MULTIPLE ORACLE HOMES Windows 2000 全面支持多个ORACLE home。以前在Windows NT上这是个主要的问题，一直到ORACLE8.0.4以后才开始支持。最初得对这一点的支持很差劲。ORACLE Home Selector, ORACLE8i 的一个新的应用工具, 改变环境路径，使选择的ORACLE home 路径作为主的home。只是简单的改变系统路径，把ORACLE选择的BIN目录放在启动路径中。每一个BIN目录都有一个ORACLE.KEY文件，指明在注册表中ORACLE程序在哪里可以找ORACLE_HOME 和其他的环境变量。如果在服务器上面只有一个数据库，通常在注册表中设定ORACLE_SID。不过，不要设定 ORACLE_HOME，对于ORACLE产品来说根本不需要，可能会导致问题。文件系统多 ORACLE home的支持允许在Windows上面实现 Unix 的OFA 标准。这极大的简化了从Unix的过渡。OFA目录树的顶层的名字有差异，不过主要的子目录和文件名字在两种操作系统中都是一致的。 Unix NT ORACLE_BASE

```
/oracle/app/oracle D:\Oracle ORACLE_HOME
/oracle/app/oracle/product/8.1.7 D:\Oracle\Ora817 Admin
directories /oracle/app/oracle/admin D:\Oracle\Admin Database
files /db01/oradata/SID D:\Oracle\Oradata ID /db02/oradata/SID
F:\Oracle\Oradata ID /db03/oradata/SID G:\Oracle\Oradata ID
```

服务管理器 从ORACLE 8i开始，服务管理器的名字在不同的平台上都一致了，都叫做svrmgrl。以前在 Windows NT 上ORACLE的执行文件名字随着版本变动而改变，对于那些在多平台上工作的人来说这很令人讨厌，尤其是在使用一些

命令 (imp、exp等) 的时候。

ORACLE 服务器版本 Windows 服务器管理器可执行文件

----- 7.3

svrmgr23 8.0 svrmgr30 8.1 svrmgrl

----- 要

注意server manager 正在逐步被淘汰(译者注：9i中彻底淘汰了svrmgrl)，一些额外的功能被加到了SQL*Plus 中。 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