

## 第7章 探讨 Oracle 环境

本章要点：

- 创建Oracle环境
- 设计一个优化的弹性体系结构
- 配置Oracle环境
- 理解Oracle软件的环境变量
- 创建你的第一个数据库
- 探索Oracle数据库
- 研究不熟悉的环境

### 7.1 创建Oracle环境

Oracle数据库是一个强大但复杂的数据存储和检索系统，有能力支持庞大的工作量，同时保持性能、安全性和数据一致性的高水准。一个成功数据库的重要部分是在数据库服务器上软件安装和配置的方法。这一章研究服务器软件和配置在数据库环境中扮演的角色，并澄清在Oracle服务器安装中容易引起混淆的问题。这一章也探讨创建 Oracle环境和创建第一个数据库的策略。最后是如何熟悉你所不熟悉的 Oracle数据库。

当配置一个服务器安装 Oracle软件时，必须考虑到几个关键的因素。这些因素包括：设计的灵活性、管理的方便性与结构的简单性。如果在设计环境时将这些因素谨记在心，你得到的回报是获得有益于达到你的正常运行时间和性能目标的环境，同时避免一个拙劣设计的服务器结构可能产生的问题。

### 7.2 设计一个优化的弹性体系结构

创建和配置你的 Oracle环境的 Oracle标准叫做优化弹性体系结构，或 OFA标准。这是一套规则和指南，能使你很容易地创建和配置一个产品级的 Oracle服务器。使用OFA标准能够在不牺牲使用性、管理方便性和设计简单性的情况下，创建最灵活的 Oracle环境。

Oracle公司的 Cary Millsap先生在90年代早期编写了OFA标准。这个结构的权威性的白皮书能够在<http://www.europa.com/~orapub/index.html>中找到。Millsap先生的论文比这里的描述更加详细和深入，他的白皮书中涉及配置整个环境；然而，你所关心的是创建和配置用于支持Oracle数据库的操作系统目录结构和命名规定。任何对实现 Oracle安装的最佳实践感兴趣的 Oracle DBA系统集成者都应该阅读 Millsap先生的OFA标准论文。

Millsap先生用自己的话概括了OFA标准建议目的：“好的标准应该有坚实的基础，不要成为空中楼阁。”OFA标准建议正是这样的建议。你特定的环境也许具有一些要求和声明，要满足它们最好使用不包括在（或相反）OFA标准的方法。然而，仔细研究 OFA标准会发现最佳的实践方法，这些方法只有通过艰难的试验、错误和实践才能发现。

### 7.2.1 创建顶层目录

配置环境的第一步是决定命名和创建操作系统安装点以及你的 Oracle 数据库安装的顶层目录。由于具有许多配置项目，该步骤的实现在 UNIX 和 NT 操作系统上有所不同，但是基本的概念仍然是相同的。优化弹性体系结构第一条规则与这个主题相关，如下所示：

“OFA 规则 1——命名所有包含特定地点数据的安装点与 /pm 格式相匹配。这里的 p 是不代表任何安装点的内容的字符串常量，m 是长度固定、用来区分其他安装点的唯一键。”

换言之，创建的顶层安装点各不相同，但它们并不代表任何意思。字符部分（p）应该简短——理想情况是一至两个字符，用数字部分来区分各个不同的安装点。每个安装点应有相同的命名部分和相同的长度（例如 u01、u02、u03，或者 ora1、ora2、ora3）。

在 NT 服务器上，不同驱动器和卷标已经按照驱动器盘符分开。这使得使用前面描述的命名规则变得不必要。在每个驱动器上使用命名目录 orant（Oracle 缺省命名）指示 Oracle 应用和文件。

提示 为了减少混淆，UNIX 命名惯例完全使用这些解释。但是，除非另外声明，所有给定的结构和命名规定既适用于 UNIX 环境，也适用于 NT 环境。

在重新配置你的系统时，如果使用不是特定地点、特定应用或者特定硬件内涵的安装点，能获得很大的灵活性。考虑下面这个例子：Bob 是一个初学的 DBA，他在 UNIX 系统上命名安装点，指定每个安装点到物理磁盘卷标。这样，对 Bob 来说在驱动器上均衡他的数据文件 I/O 是很容易的。扫视一下路径，他就能说出他的主要数据表空间在 DISK01 上，而索引表空间位于 DISK05。但是，总会存在这种情况，他的数据库增长了，强迫他增加更多的磁盘和新的硬件。Bob 想利用他的新 RAID 控制器分离和镜像一些卷，这意味着他必须重新组织他的整个驱动子系统以完成这个目标。不幸的是，这也意味着，如果他想保持他的命名惯例，他必须重新命名他的安装点并改变所有他的备份脚本、应用等等，以指向新的路径。Bob 可以通过使用唯一的、但是本身没有意义的目录名来避免这种情况。

### 7.2.2 使用应用目录

实现遵循优化弹性体系结构准则的配置的好处之一是：使用相同的目录结构，能够很容易地安装多个应用和同一应用的多种版本。在 OFA 环境中，每一个应用都有它自己的主目录，用于存储二进制、配置文件和类似的运行这个应用所必须的文件。与此相关的另外两个 OFA 准则如下：

“OFA 准则 2——命名应用主目录应匹配的形式是 /pm/h/u，这里的 pm 是安装点的名字，h 从一个小的标准目录名集合中选择。u 是应用或者应用的所有者名。”

“OFA 准则 3——存储每一个版本 Oracle 服务器发布软件的目录应匹配的形式是 /h/product/v，这里 h 是 Oracle 软件拥有者的应用主目录，v 代表软件的版本。”

以下这个例子是执行这个规则的目录模板：

[/安装点]/APP/[应用主目录]/PRODUCT/[版本]/应用名

在这个例子中，你看到应用主目录存储在直接位于安装点之下的 APP 子目录中，接着使用分级命名来区分存储在目录下的应用，然后是 PRODUCT 目录和版本。使用命名规则的分层命名的例子如下：

```
/u01
  app
    finance
      [...]
    qse
      [...]
    oracle
      admin
      product
        7.1.6
          [...]
        7.3.3
          [...]
        8.0.3
          [...]
```

尽管这种方法可能看上去比需要的复杂得多，但是它带来的灵活性和易于管理性使它是值得采用的。正如你所看见的，在这个安装点之下安装有三个应用——finance、qse和oracle。finance和qse是第三方产品或者是主目录的增长应用，在主目录之下的目录结构与Oracle分层显示的格式相同。在Oracle应用目录中，你能看到product和admin子目录。在product目录里有三种版本的Oracle软件——7.1.6、7.3.3和8.0.3。任何改变或者一个新版本的Oracle软件的安装都不会影响到其他版本。

当用于执行测试和安装新版本的Oracle数据库时，用这种方式分开版本是十分重要的。显而易见的是你需要有能力很容易地安装一个新版本的软件而不影响已安装的产品。在你的新版本被测试后，搬移新版本是一个简单的事情，只需设置正确的环境变量为一个新目录即可。在你空闲时可以删除旧版本。

### 7.2.3 管理数据库文件

当DBA同时管理多个数据库时面临的困难成为创建OFA指南的必要性之一。需要一个方法将管理信息和数据文件组织成可管理和可预测的格式。你在前面所见的admin目录，被用作存储和管理与数据库相关的文件，并极大地改进了单人持续追踪多个数据库的能力。下一个与这个结构相关的OFA准则是：

“OFA规则4——对每一个db\_name=d的数据库，数据库的管理文件存储在/h/admin/d子目录中，这里的h是Oracle软件拥有者的登录主目录。”

下面是admin目录结构的例子：

```
/u01/app/oracle/admin
  PROD
    bdump
    udump
    cdump
    pfile
    sql
    create
```

在admin目录下的子目录解释如表7-1。

init.ora文件、追踪和转储文件、alter.log等等都存储在admin这个集中管理的目录中，这使管理由Oracle数据库产生的大量数据十分容易。你也可以在需要时增加目录来存储其他数据。

注意 在UNIX平台上创建一个从\$ORACLE\_HOME/dbs目录到admin/pfile目录的init.ora

文件链接。这保证Oracle缺省配置的完整性，并可以使你从OFA结构中受益。在NT平台上，要么在启动数据库和创建一个符合OFA规则的结构时，必须对init.ora文件指定全部路径，要么必须在缺省的\$ORACLE\_HOME/database目录中存储init.ora文件，来保持缺省的Oracle配置的完整性。

表7-1 admin 目录

目录名	用 途
bdump	后台转储文件（BACKGROUND_DUMP_DEST的值）
udump	用户转储文件（USER_DUMP_DEST的值）
cdump	内核文件（仅在UNIX中）
pfile	init.ora文件和任何其他数据库初始化参数
sql	数据库管理的SQL文件
create	用于创建最初的数据库和数据库对象的脚本

## 7.2.4 命名约定

当在一个数据库服务器上管理多个数据库时，文件命名约定是十分重要的。OFA标准对数据库文件的命名给出如下的规定：

/pm/q/d/control.ctl ——控制文件。

/pm/q/d/redon.log ——重做日志文件。

/pm/p/d/tn.dbf ——数据文件。

这些命名约定定义如下：

pm 是安装点名。

q是一个字符串，表明Oracle数据与所有其他文件分开。

d 是数据库的db\_name。

n是一个对给定文件类型有固定长度的区分键。

t是Oracle表空间名。

除了控制文件、重做日志文件或数据库 d相关的数据文件之外，不要在pm/q/d下存储其他的任何文件。

采用这个标准的严格程度取决于个人的偏爱，但是命名的约定这个思想不应该被忽视。用一个有意义的文件扩展名，在数据文件名中包括表空间名，并将所有的数据文件保持在它们各自不同的目录中。

一个供选择的保持最初命名约定的模板可能是 tn\_SID\_n.ext，这里tn是数据文件的表空间的缩写，SID是数据文件所属的数据库，n是数据文件号。

尽可能地保持表空间的缩写简短，如果可能，保持它为一个字母符号集。这样可以产生一致的报告。而且，重做日志文件的数字部分应该同时表明日志组和日志成员——使用数字和字符的组合，例如01a（第一个日志组，第一个成员）和02c（第二个日志组，第三个成员）等。命名使用的替代符号如表7-2所示。

正如你所见，为你的数据库文件实现一个命名约定可以产生不包含不确定意义的名字；通过查看它们的名字，你可以立即确定它们的功能。

OFA规则也表明数据库的数据文件、重做日志文件和控制文件应该存储在安装点之外的

专用目录中。下面列出的目录结构是遵循这个建议的目录结构，其中含有四个数据库的数据。

```
/u01
  app
  oradata
    PROD
      [...]
    PPRD
      [...]
    DEVL
      [...]
    TEST
      [...]
```

表7-2 数据库文件举例

文 件 名	解 释
syst_PROD_01.dbf	PROD数据库的第一个系统表空间数据文件
ctrl_TEST_02.ctl	TEST数据库的第二个控制文件
redo_PPRD_01a.log	PPRD数据库的第一个重做日志组的第一个成员
redo_PPRD_02c.log	PPRD数据库的第二个重做日志组的第三个成员
data_PROD_02.dbf	PPRD数据库的第二个数据表空间数据文件
initTEST.ora	TEST数据库的初始化参数文件

所有的数据文件、日志文件、控制文件和归档重做日志文件都存储在 oradata子目录下的目录中。每个数据库都有它自己的目录，仅有数据库文件存储在这些目录中。

### 7.2.5 综合

一个聪明人曾说过关于标准最重要的事情是灵活掌握。当描述 OFA指南时，这是特别适合的。没有特别好的创建 Oracle服务器环境的解决策略，因为不存在两个完全相同的环境。

清单7-1和清单7-2是在UNIX和NT下的OFA实现例子。这主要是给你指明思想，让你开始工作。当你重新工作时，你可能会发现一些特殊方法比那些现存的方法更适合你的特定需要。下面也给出各部分的解释。

清单7-1 实现OFA准则的UNIX例子

/根目录	
u01	u01安装点
app	应用目录
oracle	Oracle应用目录
admin	系统管理目录
PROD	PROD数据库目录
pfile	初始化参数文件
bdump	后台转储文件
udump	用户转储文件
cdump	内核文件
create	数据库创建脚本
sql	SQL脚本
PPRD	PPRD数据库目录
[...]	与PROD有相同的目录结构
TEST	TEST数据库目录
[...]	与PROD有相同的目录结构

product	应用文件目录
7.1.3	7.1.3版本文件
[...]	7.1.3版本的Oracle主目录
8.0.3	8.0.3版本文件
[...]	8.03版本的Oracle主目录
qse	QSE应用目录
admin	系统管理目录
[...]	QSE系统管理目录
product	应用文件目录
[...]	QSE版本和文件
oradata	Oracle数据库文件目录
PROD	PROD数据库文件
PPRD	PPRD数据库文件
TEST	TEST数据库文件
u02	u02安装点
oradata	Oracle数据库文件目录
PROD	PROD数据库文件
PPRD	PPRD数据库文件
TEST	TEST数据库文件
[...]	

清单7-2 实现OFA规则的NT例子

C:	驱动器盘符
oracle	Oracle软件目录
home	Oracle主目录
[...]	Oracle主目录内容
oradata	Oracle数据库文件
PROD	PROD数据库文件目录
[...]	PROD数据库文件
PPRD	PPRD数据库文件目录
[...]	PPRD数据库文件
admin	Oracle系统管理目录
PROD	PROD管理目录
[...]	与UNIX操作系统中的结构相同
PPRD	PPRD管理目录
[...]	与UNIX操作系统中的结构相同
D:	驱动器盘符
oracle	Oracle软件目录
oradata	Oracle数据库文件
PROD	PROD数据库文件目录
[...]	PROD数据库文件
PPRD	PPRD数据库文件目录
[...]	PPRD数据库文件
[...]	

使OFA标准起作用的一个关键因素是将应用与数据库文件分开。创建目录 admin和oradata 目录以保证所有特定的数据库文件与 Oracle软件分开。这对满足前面讨论的原则之一很必要：新软件版本的简单的和透明的分离。

Millsap先生的论文的最后一OFA准则如下：

“OFA规则6——仅在特定设计用来存储它们的文件中指出显式路径名，例如 UNIX /etc/passwd文件和Oracle oratab文件；仅在/etc/group 中指出组成员。”

如果遵循这个规则，这个规则能够帮助你避免犯每一个人都易犯的错误：快速创建一个应用并使之成为一个每天要用的常用工具，但在任何系统重新配置时都会崩溃。避免在一个壳脚本、批处理文件或应用程序中使用硬编码目录名。当使用 Oracle工具工作时，实现这一条比想象的要容易。任何目录或你可能需要用的值可能早已作为一个环境变量或注册键实现了。

### 7.3 配置Oracle环境

Oracle服务器是一个复杂的交互式程序和进程的集合。所涉及的软件环境变量很复杂，经常是难以理解。存在各种不同的参数控制 Oracle程序的行为、为配置或共享文件定位、定义使用的语言和字符集及执行其他的大量任务。在 UNIX机器上，这些参数作为环境变量进行存储，这些环境变量通常从称为用户登录脚本的 coraenv或oraenv文件中装入。在 16位的Windows计算机，这些参数存储在 oracle.ini文件中。在 32位Windows计算机（Windows 95、Windows NT等），这些配置参数被存储在注册表中。

环境变量文件或注册表键值在机器上第一次安装 Oracle产品时创建。其后安装的软件增添和更新这些配置。系统程序员或 DBA也能通过手工修改来调整这些配置、查找故障点和自定义Oracle环境。一个完整的软件配置参数的描述，能够在你的操作系统中特定的 Oracle文献中找到。花一小会儿时间熟悉你的操作系统中各种不同参数；许多问题都能通过细心地研究这些参数得以解决。

注意 不要混淆Oracle配置参数和一个数据库的初始化参数。Oracle配置参数应用于安装在一个计算机上的Oracle软件，而初始化参数构造单个数据库。

有几个你应该立即熟悉的环境变量，因为它们在运行的Oracle服务器中扮演着重要的角色。ORACLE\_HOME定义 Oracle软件的基本路径，这在本章的后面各节中将详细解释。ORACLE\_SID为你的会话设置缺省的数据库 SID（系统识别符），许多工具用ORACLE\_SID来决定所要连接的数据库。知道环境变量所设置的值是十分重要的——如果有错误的值，既可能导致在错误的数据库上执行，也可能导致使用错误版本的 Oracle工具。

### 7.4 理解Oracle软件的环境变量

当在你的服务器或客户机上安装 Oracle软件时，Oracle目录结构被创建用来存储 Oracle程序使用的二进制可执行文件、共享库、配置文件、追踪文件等。这个目录树跨不同服务器平台和Oracle软件版本被半标准化。服务器和客户端软件存储在相同目录中。

提示 在UNIX机器上，在同一 Oracle目录中不能混合不同版本的 Oracle软件。在Windows机器上，不能在同一个Oracle目录中混合16位和32位版本的Oracle软件。

Oracle软件通常使用为该任务特别创建的操作系统帐户安装在服务器上。这个缺省命名为Oracle的用户ID，必须有权力操作所有 Oracle目录（包括那些用于软件安装和 Oracle数据文件存储的目录）和文件。使用 OFA准则的服务器配置，用户主目录是 /mount/app/oracle。Oracle帐户阻止未授权用户访问是十分重要的问题。在许多环境中，访问这个帐户会危害整个数据库环境的安全。

以下部分解释 Oracle 目录结构的各个部分。特别是，你将看到 Oracle 主目录是什么，在 UNIX 和 NT 这两个平台上这些服务器结构有何不同，在 UNIX 和 NT 平台上不同版本的 Oracle 软件如何能够共存。

#### 7.4.1 ORACLE\_HOME 主目录

Oracle 目录树的顶层被称之为 Oracle 主目录。它在服务器中被定义为环境变量或注册表的设置被命名为 ORACLE\_HOME，它是一个必须的配置参数。许多 Oracle 的可执行文件使用 ORACLE\_HOME 变量来定位共享代码、配置文件等。例如，一个 UNIX 服务器上，一个 Oracle 程序可能在 \$ORACLE\_HOME/network/admin 中查找配置文件，而不是在 /u01/app/oracle/network/admin 中查找配置文件。

注意 有些 Oracle 环境变量（如 TNS\_ADMIN）忽视缺省的用于定位文件的 ORACLE\_HOME 目录。

ORACLE\_HOME 目录应该在数据库登录时设置，可以通过用户登录脚本、注册表值或 .ini 文件值设置，这取决于你的体系结构。其他环境变量值，如 LD\_LIBRARY\_PATH（在 UNIX 系统中）和 PATH，应该参照于先前设置的 ORACLE\_HOME 变量。这是 OFA 规则的要求之一：使硬编码目录值只被放置在一个系统管理位置上。

#### 7.4.2 在 UNIX 上运行 Oracle 多种版本

当启动一个数据库时，ORACLE\_HOME 变量值决定用哪个版本的 Oracle 服务器可执行文件和库来打开一个数据库。通过操作和控制 ORACLE\_HOME 值，你能定义使用哪个版本的 Oracle 软件。这就是多版本的 Oracle 数据库如何能够同时在同一服务器上运行的原因。下面显示了 Oracle 目录结构的例子（在 UNIX 服务器上）：

```
/u01/app/oracle/product
  7.1.6
    bin
    rdbms
    [...]
  7.3.3
    bin
    rdbms
    [...]
  8.0.3
    bin
    rdbms
    [...]
```

在这个例子中，三种版本的 Oracle 数据库安装在按照 OFA 准则定义的目录结构中。在任何时候，运行服务器的多个版本可执行文件的多个数据库能够运转。如果 DBA 登录并设置他的 ORACLE\_HOME 为 /u01/app/oracle/product/7.1.6，那么他启动的数据库运行 7.1.6 版本的服务器可执行文件；如果 DBA 登录并设置他的 ORACLE\_HOME 为 /u01/app/oracle/product/8.0.3，那么他使用 8.0.3 版本的服务器可执行文件。

当在 Oracle 环境中工作时需要记住两个版本：

- 1) 软件自身的版本(服务器可执行文件和库)。
- 2) 数据库版本。

这些版本是分开的和彼此之间是有区分的。Oracle不对服务器可执行文件和数据库执行版本检查。例如，用一个7.3.3版本的软件启动7.1.6版本的数据库时，你将不会接收到错误。当多个版本的Oracle安装在同一服务器上，必须注意做到不要混合软件和Oracle版本。

**注意** 在UNIX环境下，缺省的ORACLE\_HOME和相关的环境变量设置在数据库登录文件中，通过提示用户指出要使用的SID，在oratab文件中查找SID，并用在oratab文件中所查找的SID值设置ORACLE\_HOME。在Windows环境中，ORACLE\_HOME从注册表或.ini文件中读取值。

### 7.4.3 在NT环境下运行Oracle的多个版本

因为UNIX和NT服务器结构不同，在NT服务器上运行Oracle服务器多个版本的软件，处理的方法不同于UNIX。在NT服务器上一个实例作为带有后台进程的服务实现，同时服务进程产生大量线程。这个服务从Windows NT注册表获得它的所有结构信息。因为在NT服务器仅有唯一的注册表有一组值，在任何时候，仅有唯一的ORACLE\_HOME能够在服务器上定义。这意味着在NT服务器运行多个版本的Oracle软件不能通过操作ORACLE\_HOME环境变量来进行。

为了使多个Oracle版本存在单个NT服务器上，所有特定版本的目录、可执行文件和文件都应附加有主要的版本号。例如，如果在NT服务器上数据库版本是7.2.3，Oracle服务器可执行文件被命名为oracle72.exe，而7.3.3版本被命名为oracle73.exe。照此方法，RDBMS目录被命名为rdbms72或rdbms73，Oracle应用如tkprof或Export在7.2版本的数据库上被命名为tkprof72或在7.3版本的数据库为rxo73。

**注意** 由于这种命名约定，不支持有多个较小版本的Oracle软件同时安装。例如7.3.2版本和7.3.3版本的软件在它们的目录和文件中有相同的后缀73，因此它们不能在同一NT服务器上共存。因为容易引起混淆和存在事故发生的隐患，所以尽可能地避免在同一服务器上运行多个版本NT。

Oracle提供一个应用用来在16位Windows客户和Windows 95、Windows NT机器进行主目录的转换。这不同于用多个ORACLE\_HOME目录运行，这个程序所做的所有工作就是将init.ora或注册表中的目录引用从一个位置改变到另一个位置。这使你能在多个ORACLE\_HOME目录中安装和测试软件，但是它不能使程序从分开的ORACLE\_HOME目录中同时运行。

16位的Window软件使用oracle.ini文件设置Oracle软件配置变量。32位的Windows软件使用/HKEY/LOCAL\_MACHINE/SOFTWARE/ORACLE注册表键来设置Oracle环境变量。鉴于此，16位和32位版本的Oracle软件能够共存于32位的Windows操作系统中。建议使用不同的ORACLE\_HOME(缺省值是orawin和orawin95或orant)，通过它们各自的.ini文件和注册表设置来进行分开管理，避免在不同版本之间由于共享代码不符而引发的问题。

用于16位Oracle软件的驱动器和程序不能在32位版本上运行，反之亦然。你将经常看到与此相关的使用SQL\*Net的程序（如ODBC驱动）和依靠Oracle提供的支持库和DLL库的第三方编程工具。在这种情况下，同时运行16位和32位版本Oracle软件的唯一方法是保持每项工作平稳地运行。

#### 7.4.4 ORACLE\_HOME目录

如前所述，ORACLE\_HOME变量指向Oracle软件目录树的顶层。ORACLE\_HOME包含其他目录，这些目录中又包含其他目录和文件。在 ORACLE\_HOME内定位目录是在所有 Oracle 安装中相当标准的。每个目录包含用于特定的程序或功能的文件。

表7-3列出了能够在Oracle安装中找到的一些重要目录和它们的作用。

表7-3 重要的ORACLE\_HOME目录和用途

目 录 名	用 途
BIN	Oracle可执行文件和共享DLL库
DBS	Oracle服务器文件和init.ora文件的缺省目录
DATABASE	开始的数据库和缺省的Oracle数据库文件（ NT ）
RDBMS	数据库文件
RDBMS/ADMIN	创建系统数据库的对象、角色和视图的 SQL脚本
RDBMS/TRACE	缺省的数据库转储目录
NETWORK	Oracle网络文件
NETWORK/ADMIN	SQL*NET配置文件
NETWORK/TRACE	缺省的SQL*NET转储目录
PLSQL	PL/SQL配置文件和SQL脚本
DOC	联机文档
SYSMAN	Oracle企业管理器文件
PLUS	SQL*PLUS文件

通常，admin目录包含与系统管理对象有关的 SQL脚本，这些系统管理对象与它们所在的产品目录相关。对DBA来说最需要关心的目录是\$ORACLE\_HOME/rdbms/admin，在这个目录中你能看到用于运行生成数据库对象、包和过程等的 SQL脚本。\$ORACLE\_HOME/network/admin包含用于配置SQL\*Net通信所有必需的文件。\$ORACLE\_HOME/dbs包含sql.bsq文件，这个SQL文件在数据库创建期间用来运行创建数据字典表。

#### 7.4.5 其他重要的配置文件

在你安装Oracle服务器时应该了解几个其他文件。oratab、oraenv和coraenv文件仅能在UNIX服务器中见到。oratab是系统中每个数据库在其中占据一行的文本文件，每行包括数据库SID、特定数据库的Oracle主目录和一个用来表明数据库是否自动启动或者关闭的标志。有几个由Oracle提供的程序和脚本（如dbstart或dbshut）使用这个文件来识别数据库在服务器中的位置和它们的版本号。dbstart和dbshut也使用自动启动标志来选择在哪个数据库上执行它们的活动。

oraenv和corenv是UNIX环境中的脚本文件，用于定义和设置运行 Oracle程序和工具所必需的环境变量。oraenv文件用于sh或ksh环境，而文件 coraenv用于csh环境。oraenv和coraenv也访问oratab文件，以决定正确的ORACLE\_HOME，登录数据库键入的SID基于这个值。这些文件是当你开始自定义你的UNIX Oracle环境变量时，最初开始查找的位置。

在NT服务器中，你应该了解口令文件，它是由 oradim工具为每一个数据库创建的。这个文件包括内部用户帐户口令——这个帐户在数据库启动和关闭时使用。这个口令文件被命名为oraSID.pwd或pwdSID.ora，它被设置标志、隐藏并存储在 \$ORACLE\_HOME/database目录中。它用一定的安全配置来验证数据库的远端 DBA。另外，在NT上，这个口令文件在内部与

该SID的控制文件保持同步。

## 7.5 创建你的第一个数据库

由于你已经获得了关于 Oracle环境、Oracle数据库和实例的知识，现在该了解创建 Oracle数据库的过程了。

创建Oracle数据库包括一些离散的步骤：

- 1) 为一个新数据库创建初始化参数文件和系统管理目录。
- 2) 创建实例。
- 3) 创建数据库。
- 4) 运行数据库创建后的步骤。
- 5) 创建支持的数据库对象。
- 6) 安全化缺省帐户。
- 7) 更新系统文件以识别新数据库。

在以下部分对每一步进行解释。

### 7.5.1 创建初始化参数文件

创建一个新的数据库的第一步是复制缺省 init.ora文件（在UNIX平台上，能在\$ORACLE\_HOME/dbs中找到；在NT服务器上，在\$ORACLE\_HOME/database目录中）到唯一标识参数文件属于哪个特定数据库的文件。这个文件包括一个新建数据库的特定参数设置。init.ora文件按Oracle缺省的命名惯例是 initSID.ora；这里的SID是你为新数据库选择的系统标识符。如果在启动数据库时，一个参数文件没有显式地被指定，Oracle工具将查找用这个名字命名的文件名（用ORACLE\_SID来填写文件名的SID部分）。编辑复制的文件，用有意义的值取代缺省的参数值。至少要确保 controlfile参数、db\_block\_size参数和db\_name 参数是正确无误的。

现在该是创建用来存放 Oracle数据文件、后台和用户追踪文件和其他你要使用的管理目录结构的时候了。参见7.2节“设计一个优化的弹性体系结构”，其中有这个配置的详细信息。

### 7.5.2 创建实例

实例的创建是最容易引起误解的与 Oracle数据库系统管理有关的任务之一。正如前面所讨论的，实例是由后台进程和包含在服务器内存地址空间中的内存结构所组成的 Oracle RDBMS的特定部分。创建和运行实例在 UNIX和NT平台之间有很大差异。以下部分讨论涉及这两个环境的不同步骤。

#### 1. 在UNIX上创建实例

在UNIX平台上，Oracle实例不是一个永久的结构，而是每次数据库启动时使用 init.ora文件中特定的数据库启动参数来创建的。实例名使用环境变量 ORACLE\_SID的值。实例创建的确切时间发生在非安装阶段。当你启动数据库到非安装态时，你所做的工作仅只是分配系统全局区和启动服务器后台进程。取决于你的 UNIX平台，实例名（或SID，由ORACLE\_SID值定义）也有可能是后台进程任务名的一部分。

#### 2. 在NT上创建实例

对于创建实例和用实例工作，在NT服务器上所涉及到的内容要比 UNIX服务器上多很多。

由于NT的结构，所有UNIX控制台进程都使用NT服务执行。一个服务必须定义和存储在 NT注册表中之后，才能使它激活。因此对每一个你拥有的 NT数据库，你必须为数据库创建一个与主机实例相关的服务。不同于 UNIX服务器，不管实例是否被启动，NT Oracle实例存在于一个永久存在的物理操作系统结构中。

在多数安装中，每一个数据库被分配两个NT服务，只有其中的一个服务(即OracleServiceSID)是数据库操作所需要的。OracleServiceSID服务由嵌入它的定义的初始化参数文件创建，这就是为何当NT创建实例时知道查找哪个 init.ora文件和该实例是为哪个数据库创建的(使用在init.ora中的controlfile参数)。另一服务(OracleStartSID)是当NT服务器启动时提供的自动启动你的实例和数据库的一种方便方法。当 OracleStartSID被启动时，它自动启动OracleServiceSID并打开OracleServiceSID所指向的数据库。创建OracleStartSID服务在本章后面的各节讨论。

运行Oracle实例所需的服务使用 oradim工具创建。记住 oradim正如所有其他主要的 Oracle NT可执行文件一样，将在其上附加 Oracle版本的前两位数字，例如 oradim80.exe。oradim有两种操作方法——图形用户界面(GUI)接口和命令行方式。那些习惯使用 NT的用户将可能喜欢用GUI接口；而对UNIX管理员，更喜好用命令行方式。用无参数形式激活 oradim将自动启动GUI接口。GUI接口能够用来创建初始化服务和实例；而命令行选项能够有更多的控制能力和更强大的系统管理选项。

关于oradim值得注意的一件事是，它不能将错误信息输出到标准输出或打印到屏幕上，而是将所有的错误信息写入位于 \$ORACLE\_HOME/rdbms目录的 oradim.log文件中。在未来Oracle版本中将有望改变这种小小的不便。这种错误处理设计的原因已超出理解的范围。

当调用oradim GUI接口时，显示两个要求填入数据库参数的屏幕。跟随指示，oradim创建Oracle服务、创建数据库并运行 catalog.sql和catproc.sql(在7.5.4节“运行创建数据库后的步骤”中讨论)。确保改变给出的数据库名、实例名、数据文件和参数文件的缺省值。

运行字符命令的 oradim是使事情开始变得有趣。在命令行中，你能创建、编辑和删除已有的Oracle服务。对有经验的Oracle DBA来说，对命令行方式的oradim的喜爱要胜过GUI，因为它有更高级的控制和功能。对这个例子，假设使用 oradim命令行方式。

最基本的oradim命令行使用格式如下：

```
oradim -command -sid SID [flags]
```

在表7-4中描述了oradim命令。

表7-4 oradim 命令行选项

命 令	功 能
NEW	创建一个新的数据库实例服务
EDIT	编辑一个现有的实例服务
STARTUP	启动一个实例和/或数据库
SHUTDOWN	关闭一个实例和/或数据库
DELETE	删除实例服务

需要意识到的一个重要的事情是，实例服务完全独立于与它们相关的数据库。在 NT服务器中一个数据库能够存在，而不需要与之相关的实例一定存在。并且，在创建这个服务时使用的oradim版本指定在启动实例时将运行的 Oracle服务器可执行文件(oracle.exe)的版本。

要创建一个Oracle服务，发出如下命令：

```
oradim -new -sid [sid_name]
-pfile [init.ora directory]
-intpwd [internal password]
-startmode auto
```

这就创建了 OracleServiceSID和OracleStartSID，其中SID为设定的sid\_name值。auto的startmod创建 OracleStartSID文件。pfile参数给出用于配置由OracleServiceSID服务启动的实例的init.ora文件的位置。

更详细的使用oradim命令的信息和例子，参见特定的Oracle NT操作系统文献。

### 7.5.3 创建数据库

在配置好实例之后，你可以创建数据库，这就是你真正发出 CREATE DATABASE命令的时候。CREATE DATABASE 命令创建系统数据文件、表空间、控制文件、初始的重做日志组、系统回滚段和运行Oracle数据库所需要的内部数据库表。

CREATE DATABASE命令的语法能够在 Oracle Server Reference Manual中找到。在CREATE DATABASE命令中指定正确大小的数据文件是十分重要的，这个数据文件被分配给系统SYSTEM表空间。由于硬盘驱动器空间是相对不太昂贵的，将文件大小设置为比你希望使用的大两到三倍（至少 50~100MB）。如果数据库将大量使用存储包、过程、函数和触发器，那么需要分配更多的空间，因为这些对象的代码存储在 SYSTEM表空间中，会占用相当多的空间。同时将你的 MAX参数设置得大一些。如果没有明确声明 MAXDATAFILES、MAXLOGFILES等的状态值，将使用缺省值，通常缺省值十分低。唯一的缺陷是控制文件将比你所需要的大几个KB——这几乎算不上是一个缺陷。你还要指定字符集（CHARACTERSET）的正确值。它是在数据库安装之后不能改变的几个参数之一。

在运行CREATE DATABASE命令之前，确保在 init.ora文件中控制文件参数设置正确，你的ORACLE\_SID设置为正确的数据库名。如果你在 init.ora参数中设置了db\_name参数，要确保它与你的Oracle服务器的SID值相同。将SQL语句CREATE DATABASE保存在一个文本文件中，可以减少由于打字错误发生错误的概率和能够提供一定级别的自我参考文献。同时保证在NT控制面板运行Services applet来启动 OracleServiceSID服务并保证OracleServiceSID正在运行。

在UNIX上运行svrmgrl；在NT系统中对于7.3.x运行svrmgr23.exe；或对于8.0.x运行svrmgr30.exe。以internal连接到数据库。对7.3.x，你得到一个状态信息，表明你已“连接到一个空闲的实例”。这表明实例正在运行，但是在其上没有依附数据库。但是对于8.0.x，你仅接收到“已连接”消息，尽管实际上实例是空闲的。发出以下命令来启动非安装态的数据库：

```
startup nomount pfile=[init.ora file path and name];
```

当数据库开始运行你的CREATE DATABASE脚本时，完成这个命令所需要的时间主要取决于你所产生的系统数据文件的大小。当创建命令完成后，数据库自动打开。这时，你有一个工作的（虽然是空的）Oracle数据库。

### 7.5.4 运行数据库创建后的步骤

在数据库创建完成之后，有两个Oracle提供的SQL脚本必须立即运行：

```
catalog.sql
```

```
catproc.sql
```

在创建数据库之后，oradim GUI工具自动执行这些脚本。这两个脚本能够在 \$ORACLE\_HOME/rdbms/admin中找到。catalog.sql创建数据字典视图和V\$视图同义词，而catproc.sql运行安装过程化选项或PL/SQL对象和它支持的数据库结构需要的脚本。这两个脚本之中的任意一个运行失败，将导致奇怪的和不可预测的错误。

另一个你可能想运行的脚本是publd.sql。它创建用户的环境资源表，这些Oracle提供的表能够用来限制Oracle程序访问数据库。产品用户环境资源表在今天很少被使用，这是由于许多第三方应用对数据库的访问使用本地或ODBC驱动，它们不检查这些表对资源的限制。但是如果你没有创建这些表，每当不是SYSTEM用户的用户通过SQL\*Plus登录数据库时，他在屏幕上会得到一个令人讨厌的错误信息。当创建这些表时，应以SYSTEM用户登录。

最后，如果这个数据库涉及到复制，你需要运行 \$ORACLE\_HOME/rdbms/admin/catrep.sql。这个脚本创建支持复制选项所需要的过程和对象。

**提示** 对于数据库进程的创建，非常有意义的一点提示是在你运行数据库创建脚本之前，启动一个缓冲文件，然后当脚本运行之后，浏览它的错误。这将帮助你迅速查找错误。

### 7.5.5 创建数据库支持的对象

现在你有了一个数据库框架，你能够开始创建能够使你的数据库支持产品级工作量的支持对象。创建支持对象的关键是灵活性和可管理能力。如果你能将这两个目标记在心中，你的数据库安装将是成功的。

首先你需要做的是创建一个回滚段，在创建其他的你支持的数据库对象时需要用到这个回滚段。在创建时，你的数据库只有一个在系统表空间中创建的系统回滚段。这个回滚段仅只被用作涉及系统表空间的事务，它不能被删除或脱机。使用如下命令创建一个临时回滚段：

```
create rollback segment rbs_temp
storage (initial 10k next 10k minextents 3 maxextents 121)
tablespace system;
```

发出如下命令使回滚段联机：

```
alter rollback segment rbs_temp online;
```

现在创建你的产品表空间。至少需要创建列在表7-5中的表空间。

表7-5 数据库表空间的一些例子

表空间名	用 途
TEMP	存储临时数据库段
TOOLS	存储第三方或内建数据库工具对象
RBS	存储回滚段
RBS_02	存储用于装载大块数据的回滚段
USER_DATA	终端用户的缺省表空间
USER_INDEX	终端用户的索引表空间

你创建这些表空间的目标是分离操作性的和应用的数据和功能。TEMP、RBS和RBS\_02表空间包含的对象是Oracle数据库工作所必须的，而TOOLS、USER\_DATA和USER\_INDEX

表空间包含用户数据。参看本书第四部分“性能调整”，其中有更为详细的使用表空间分离数据、I/O和功能的信息。

同时，创建你的产品回滚段。在 RBS 表空间至少创建两个回滚段来支持 OLTP 事务。这些段应该相对较小，但有许多区间。将它们命名为 RBS##——例如，RBS01、RBS02 和 RBS03。在 RBS\_02 表空间创建一个用于装载大块数据的回滚段，这个段应该较大而有较少的扩展，称这个回滚段为 RBS\_LOAD##——例如 RBS\_LOAD01。使 OLTP 回滚段联机，将它们的名字添加到 init.ora 文件 rollback\_segments 列表中。

当你做完这些工作后，删除你早期创建的临时回滚段，使用如下命令：

```
drop rollback segment rbs_temp;
```

### 7.5.6 缺省帐户的安全性

一个新安装的数据库总有两个缺省的系统帐户——SYS 和 SYSTEM。SYS 的初始口令是 CHANGE\_ON\_INSTALL，SYSTEM 帐户的初始口令是 MANAGER。立即改变这些口令。如果你知道用 SYS 或 SYSTEM 登录破坏了多少个数据库，你会吓坏的。

并且，如果你安装了例子表，考虑改变 scott 用户的口令，从大家都知道的口令 tiger 变为其他的口令。

### 7.5.7 更新系统的配置文件

最后，你需要在你的系统中的所有配置文件中注册你新的数据库。如果这是在系统中安装的第一个数据库，你要设置你的 ORACLE\_SID 变量来反映这一点。在 NT 服务器中，在你设置这个值之前，需要先创建它。如果你在 UNIX 服务器上，你需要更新 oratab 以反映新的数据库信息。同时，将数据库放入 tnsnames.ora 和 listener.ora 文件中，以便通过 SQL\*Net 能连接它们。

现在（假如前面的所有步骤已成功地完成），你已有一个功能齐全的 Oracle 数据库，可以开始开发、测试和运行了。

## 7.6 探索 Oracle 数据库

任何使用 Oracle 工作过或长或短时间的人，都面临着管理、故障排除的任务，或者工作在对他说来完全陌生的数据库环境。能够使自己快速地熟悉常规配置并建立一个外部环境是十分重要的。要确保你的时间花在处理关键性工作上，而不在耗时、琐碎的事情上浪费时间，例如，你能用哪个回滚段进行你的数据装载，哪个表空间存放你的用户数据等。

在以下部分，你将看到如何从数据库中提取关于数据库的信息。这些节中的信息与在前面部分包括的信息相结合，这将使你能够完整地勾画出一个你并不熟悉的数据库环境。

### 7.6.1 查看数据库

第一件事是在数据库和它的对象上，获得一个句柄。这包括查看数据文件、表空间、重做日志、回滚段和对不同缓冲区的内存分配，以及运行的后台进程。存储在工作消息队列中的数据库工作，用户和分配给用户的权限也是相关的信息。

V\$dbase 和 V\$thread 数据视图被分别用来定位数据库信息和实例信息。V\$dbase 显示

数据库名和归档模式（数据库运行在归档模式，还是非归档模式）。V\$thread视图是并行服务器特有的视图，但是当它给出实例名时，它是有用的（在 NT服务器比在 UNIX环境有更多的说明）。你也可以查询v\$instance视图来看实例的启动时间，如果你使用 Oracle 7，以下查询转换v\$instance视图启动信息为可用的格式：

```
SELECT TO_CHAR (TO_DATE (a.value, 'J')
+ b.value/86400, 'HH24:MI:SS DD-MON-RR')
start_time
FROM v$instance a, v$instance b
WHERE a.key = 'STARTUP TIME - JULIAN'
AND b.key = 'STARTUP TIME - SECONDS';
```

这个查询有类似下列的输出结果：

```
START_TIME
-----
22:48:37 06-MAR-98
```

在Oracle 7中，这个相当精心设计的查询是必须的，因为启动时间的 Julian日期部分被存放v\$instance一列中，而日期的时间部分以秒为单位存储在分开的域中。

注意 v\$thread 包含OPEN\_TIME域，该域可能不同于v\$instance视图的时间显示。这是由于一个域显示实例被创建的时间，而另一个显示数据库被打开并能够进行访问的时间。

你也能通过查询v\$controlfile视图列出活动的控制文件，回滚段信息存储在dba\_rollback\_seg、v\$rollstat和v\$rollname中。dba\_rollback\_segs显示所有在数据库中的回滚段信息，而 V\$视图仅列出当前联机的回滚段。下面的查询列出联机回滚段信息：

```
SELECT a.segment_name, a.tablespace_name,
b.bytes, b.extents,
c.shrinks, c.extends, c.hwmsize
FROM dba_rollback_segs a, dba_segments b, v$rollstat c
WHERE a.segment_id = c.usn
AND a.segment_name = b.segment_name;
```

这个查询产生类似如下的输出：

```
SEGMENT_NAME      TABLESPACE_NAME
BYTES      EXTENTS
SHRINKS      EXTENDS      HWMSIZE
-----
-----
RBS01          ROLLBACK_DATA
102400          2
1              3      100352
RBS02          ROLLBACK_DATA
102400          2
0              0      100352
RBS03          ROLLBACK_DATA
102400          2
1              1      100352
SYSTEM          SYSTEM
204800          4
0              0      202752
```

重做日志信息被存储在 v\$log和v\$logfile视图中，以下查询显示出重做日志中为数据库创建的永久信息。

```
SELECT member, bytes, members, a.status
FROM v$log a, v$logfile b
```

```
WHERE a.group# = b.group#
```

```
ORDER BY member;
```

这个查询的输出与下面类似：

MEMBER	BYTES	MEMBERS	STATUS
C:\ORADATA\PPRD\REDO_PPRD_01A.LOG	204800	1	CURRENT
C:\ORADATA\PPRD\REDO_PPRD_02A.LOG	204800	1	INACTIVE

映射你的数据库文件和表空间是重要的。从 `dba_data_files` 中查询这个信息。你能从 `dba_tablespace` 中查询指定表空间信息和从 `V$datafile` 中查询指定的数据文件信息。

查询 `V$sgastat` 视图查看实例中的内存分配信息，以下查询可能对你有帮助：

```
SELECT name, bytes
FROM v$sgastat
WHERE name in
('free memory', 'fixed_sga', '
db_block_buffers', 'log_buffer',
'dictionary cache', 'library cache','sql area');
```

查询一个非常小的数据库返回的结果如下：

NAME	BYTES
free memory	2287676
fixed_sga	35208
db_block_buffers	409600
log_buffer	8192
dictionary cache	226376
library cache	247636
sql area	466040

`v$bgprocess` 视图能够用来快速列出正在运行的后台进程。如下查询会对你有帮助：

```
SELECT name FROM v$bgprocess WHERE paddr <> '00';
```

表 `dba_jobs` 含有通过 `DBMS_JOBS` 包按时间表未完成的作业的信息。你能使用如下查询列出时间表数据库作业：

```
SELECT log_user, job, what,
to_char(last_date, 'DD-MON-YY') l1, last_sec l2,
to_char(next_date, 'DD-MON-YY') n1, next_sec n2, failures
FROM dba_jobs
ORDER BY next_date DESC, next_sec DESC;
```

用户信息被存储在 `dba_users` 中，授权信息存放在 `dba_tab_privs`、`dba_sys_privs` 和 `dba_role_privs` 表中。你能通过查询 `role_tab_privs`、`role_sys_privs` 和 `role_role_privs` 来显示授给角色的特权信息。关于控制安全信息的更详细的信息可在第 23 章“安全管理”中获得。

## 7.6.2 查看数据库段

既然你已知道数据库结构信息存放的位置，那么查看一下用户数据对象信息。你所感兴趣的数据库段包括表、索引、视图、簇和其他数据存储对象。

`DBA_TABLES`、`DBA_INDEXES`、`DBA_SEGMENTS`、`DBA_SEQUENCES` 和 `DBA_OBJECTS` 数据字典视图提供了存储在数据库中各个不同数据库段中的信息。

在这些视图中的格式和信息是相当清楚的。`DBA_OBJECTS` 是很有趣的，因为它存储数据库中所有对象创建和最后修改（时间戳）的信息。`DBA_SEGMENTS` 是有用的，因为它保存了被每个数据库段使用的区间的大小和数量。请参看第 21 章“管理数据库存储”，其中有更

为详细的说明。

### 7.6.3 查看各种各样的数据库对象

你还应该查看一些其他的结构，包括存储的 PL/SQL对象、触发器、同义词和数据库链接。

包、过程和函数的源存储在 sys.source\$表中。当你创建对象时，所有的空行被忽略，每行作为一个分开的记录存储在这个表中。查询 dba\_source可以很容易地查出存储的包、过程和函数的文本内容。下面的查询产生指定的存储对象的源代码：

```
SELECT text
FROM dba_source
WHERE name = upper('&Object_Name')
ORDER BY line;
```

有关触发器的信息存储在 dba\_triggers表中。因为触发器体是作为一个长整型存储的，所以当你用 SQL\*Plus 查询这一列时，你需要设置 longsize 为足够长的大小。在这一视图上，TRIGGER\_TYPE 列既可以是行，又可以是语句，TRIGGERING\_EVENT 存储触发器的类型——before insert、after update 等等。

dba\_synonyms 列出数据库中的同义词信息。注意公共同义词，用 PUBLIC 作为拥有者。查询 dba\_db\_links 获得在数据库中定义的数据库链接的信息。关于 dba\_db\_links 有一个警告：如果这个链接是用连接的用户名和口令定义的，则口令被存放为明文。

## 7.7 研究不熟悉的环境

现在把包含在本章中的一些信息放入一个熟悉的参考框架中。假如你是一个签约的 DBA，刚到达一个新场所，前一个 DBA 已离开现场，而且他没有留下任何关于他的系统结构的支持文献，该怎么办？

以下两节探讨可能采取的行动——一个是在 UNIX 服务器上，另外一个在 NT 服务器上。它们应该会给你一些思路，你还要求助于本章前面所描述的信息。

### 7.7.1 研究 UNIX 环境

你主要的兴趣是判断这个 Oracle 软件是如何安装和配置的和在系统中安装了什么数据库。首先，获得用于安装 Oracle 软件的操作系统帐户的访问权。接下来，你需要定位可能的 ORACLE\_HOME 目录。如果在系统中仅有一个 ORACLE\_HOME，它应该已经放置在你的登录序列中。从你的环境变量中检查 ORACLE\_HOME 的值，同时查找和检查 oratab 文件以确定其中是否包含数据库和 ORACLE\_HOME 中的信息。这不是一个确保完全的清单，有可能 oratab 文件被手动更新，最新的信息不是数据库操作所要求的。另一个选择是搜索 Oracle 服务器可执行文件（oracle）或 rdbms 和 dbs 目录。在所有的 Oracle 服务器安装中，这是标准的。对所有可能的 Oracle 目录作一个详细清单。

接下来，判断服务器中有什么数据库。在每一个你所见的 Oracle 主目录中，检查 dbs 目录。在这个目录中，所有配置的数据库应该有一个 initSID.ora 文件。这个文件可能被链接到不同的管理目录，但是这些文件的列表显示所有已配置的数据库。再次声明，这并不一定意味着所有这些数据库都存在。创建和删除 init.ora 文件，绝大多数是由 DBA 手工进行的。并且不保证

init.ora文件在dbs目录中，由于dbs目录是Oracle程序缺省查找的地方，所以你很可能在那里找到它。

检查你的用户和系统登录的脚本来判断是否 Oracle环境已经进行过自定义修改。查看coraenv或oraenv是否正在被登录脚本调用和是否这些文件已经发生了其他方式的修改。如果coraenv或oraenv没有被调用，判断是否 Oracle环境变量正在被设置，检查系统的启动脚本（rc文件和与此类似的）来看什么程序或后台进程用系统根用户自动启动。不要忘记检查crontab文件或系统的时间表，以便Oracle拥有者和系统管理判断什么是按时间表规划并自动运行的作业。最后，检查UNIX内核参数来查看Oracle安装如何配置内存和内核。设置这些参数的具体位置和方法随UNIX厂商而有所不同，查看你的操作系统文献以获得更详细信息。

提示 使用UNIX的find命令来自动定位文件，否则需要手工查找。查找你的UNIX系统的联机手册（man）获得正确的语法。

### 7.7.2 研究NT环境

在NT服务器上，理解Oracle环境的特定配置要比UNIX服务器上直接一些。这是因为只有很少方面的配置要让DBA去费心完成。你的第一步是翻译环境变量，查看Oracle注册表。使用regedit查看HKEY\_LOCAL\_MACHINE\Software\Oracle键，这个字符串值告诉你ORACLE\_HOME的位置和其他的有用信息。接下来，在Control Panel中打开Services applet，查看所有Oracle服务。在系统中每个数据库都有一个Oracle服务，同时每个数据库也可能有一个Oracle启动服务。同时查看其他被安装的服务——TNS监听器、Web服务器，等等。

所有相关的配置值被存储在Oracle注册表键中。数据库的参数文件在\$ORACLE\_HOME/database目录下，这个目录也是数据库数据文件存储的缺省目录。使用Windows NT资源管理器帮你定位文件。一定要检查NT任务管理器和作业时间表来查看可能运行或已经在运行的作业。