ORACLE 数据库启动时，经历了三个过程：(用命名如下）
startup nomount;
alter database mount; alter database open;
当然数据库关闭时也是经历了三个相反的过程：
alter database close;
alter database dismount;
shutdown;（shutdown 后面跟了四个参数：normal;immediate;transactional;abort)
四种方式关闭数据库的比较：
(NO YES)

关闭方式
允许新的连接
等待活动会话终止
等待活动事务终止
强制进行 checkpoint,关闭所有文件

分析第一个过程 startup nomount:

这个过程数据库首先到参数文件（pfile/spfile)中读取数据库的设置，创建实例.

数据库所在的操作系统版本：

```
[oracle@localhost ~]$ lsb_release -a
LSB
Version:      :core-3.1-ia32:core-3.1-noarch:graphics-3.1-ia32:graphi
cs-3.1-noarch
Distributor ID: EnterpriseEnterpriseServer
Description:      Enterprise Linux Enterprise Linux Server release 5.5
(Carthage)
Release:           5.5
Codename:          Carthage
```

数据库版本：

```
SQL> SELECT * FROM v$version where rownum=1;


BANNER
--------------------------------------------------------------------
-----------
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
```

因为 spfile 是二进制文件，不能直接读取，在 linux 中，可以用命令 String 转储出来：

```
[oracle@localhost dbs]$ strings spfileorcl3939.ora
orcl3939.__db_cache_size=54525952
orcl3939.__java_pool_size=4194304
orcl3939.__large_pool_size=4194304
```

```
orcl3939.__oracle_base='/u01/app/oracle'#ORACLE_BASE set from
environment
orcl3939.__pga_aggregate_target=171966464
orcl3939.__sga_target=251658240
orcl3939.__shared_io_pool_size=0
orcl3939.__shared_pool_size=176160768
orcl3939.__streams_pool_size=4194304
*.audit_file_dest='/u01/app/oracle/admin/orcl3939/adump'
*.audit_trail='db'
*.compatible='11.2.0.0.0'
*.control_files='/u01/app/o
racle/oradata/orcl3939/control01.ctl','/u01/app/oracle/flash_recovery
_area/orcl3939/control02.ctl','/u01/app/oracle/oradata/orcl3939/contr
ol03.ctl'
*.db_block_size=8192
*.db_domain='localdomain'
*.db_name='orcl3939'
*.db_recovery_file_dest='/u01/app/oracle/flash_recovery_area'
*.db_recovery_file_dest_size=4039114752
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=orcl3939XDB)'
*.instance_name='ORCL3939'
*.local_listener='(ADDRESS=(PROTOCOL=TCP)(HOST = local
host.localdomain)(PORT = 1521))'
*.memory_target=423624704
*.open_cursors=300
*.processes=150
*.remote_login_passwordfile='EXCLUSIVE'
*.service_names='a,b,c,d'
*.trace_enabled=TRUE
*.undo_tablespace='UNDOTBS1'
```

spfile 文件中你可以看到数据库在 nomount 时做了些什么，根据参数文件的内容，创建了 instance,分配了相应的内存区域，启动了相应的后台进程。

我们再看告警日志文件（alert_.log）：读取了参数文件，启动了实例

```
Starting up:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing
options.
Using parameter settings in server-side spfile
/u01/app/oracle/product/11.2.0/dbhome_1/dbs/spfileorcl3939.ora
```

```
System parameters with non-default values:
   processes                  = 150
   memory_target              = 404M
   control_files              =
"/u01/app/oracle/oradata/orcl3939/control01.ctl"
   control_files              =
"/u01/app/oracle/flash_recovery_area/orcl3939/control02.ctl"
   control_files              =
"/u01/app/oracle/oradata/orcl3939/control03.ctl"
   db_block_size              = 8192
   compatible                 = "11.2.0.0.0"
   db_recovery_file_dest      = "/u01/app/oracle/flash_recovery_area"
   db_recovery_file_dest_size= 3852M
   undo_tablespace            = "UNDOTBS1"
   remote_login_passwordfile= "EXCLUSIVE"
   db_domain                  = "localdomain"
   instance_name              = "ORCL3939"
   service_names              = "a,b,c,d"
   dispatchers                = "(PROTOCOL=TCP)
(SERVICE=orcl3939XDB)"
   local_listener             = "(ADDRESS=(PROTOCOL=TCP)(HOST =
localhost.localdomain)(PORT = 1521))"
   audit_file_dest            =
"/u01/app/oracle/admin/orcl3939/adump"
   audit_trail                = "DB"
   db_name                    = "orcl3939"
   open_cursors               = 300
   diagnostic_dest            = "/u01/app/oracle"
   trace_enabled              = TRUE
Thu Apr 02 14:59:41 2015
PMON started with pid=2, OS id=5989
Thu Apr 02 14:59:41 2015
VKTM started with pid=3, OS id=5991 at elevated priority
VKTM running at (10)millisec precision with DBRM quantum (100)ms
Thu Apr 02 14:59:41 2015
GEN0 started with pid=4, OS id=5995
Thu Apr 02 14:59:41 2015
DIAG started with pid=5, OS id=5997
Thu Apr 02 14:59:41 2015
DBRM started with pid=6, OS id=5999
Thu Apr 02 14:59:41 2015
PSP0 started with pid=7, OS id=6001
Thu Apr 02 14:59:41 2015
DIA0 started with pid=8, OS id=6003
```

```
Thu Apr 02 14:59:41 2015
MMAN started with pid=9, OS id=6005
Thu Apr 02 14:59:41 2015
DBW0 started with pid=10, OS id=6007
Thu Apr 02 14:59:41 2015
LGWR started with pid=11, OS id=6009
Thu Apr 02 14:59:41 2015
CKPT started with pid=12, OS id=6011
Thu Apr 02 14:59:41 2015
SMON started with pid=13, OS id=6013
Thu Apr 02 14:59:41 2015
RECO started with pid=14, OS id=6015
Thu Apr 02 14:59:41 2015
MMON started with pid=15, OS id=6017
Thu Apr 02 14:59:41 2015
MMNL started with pid=16, OS id=6019
starting up 1 dispatcher(s) for network address
'(ADDRESS=(PARTIAL=YES)(PROTOCOL=TCP))'...
starting up 1 shared server(s) ...
ORACLE_BASE from environment = /u01/app/oracle
```

数据库根据参数创建实例之后，后台进程依次启动，注意上面输出中包含了 PID 信息以及 OS ID 两个信息，PID 代表该进程在数据库内部的标识符编号，而 OS ID 则代表该进程在操作系统上的进程编号。

我们可以通过 oracle 中的动态视图 v$process，可以把后台进程和操作系统的进程想关联起来：

```
SQL> select addr,pid,spid,username,program from v$process;
ADDR PID SPID USERNAME PROGRAM
ADDR            PID
SPID                                                    USERNAME
              PROGRAM
-------- ---------- ------------------------------------------------------------------------
------------------------------------------
------------------------------------------------------------------------------------------------
------------------------------------------------
393B9444          1                                                                     PSEUDO
393B9F1C          2
5989                                                    oracle
          oracle@localhost.localdomain (PMON)
393BA9F4          3
5991                                                    oracle
          oracle@localhost.localdomain (VKTM)
393BB4CC          4
5995                                                    oracle
          oracle@localhost.localdomain (GEN0)
```

393BBFA4 5
5997 oracle
oracle@localhost.localdomain (DIAG)
393BCA7C 6
5999 oracle
oracle@localhost.localdomain (DBRM)
393BD554 7
6001 oracle
oracle@localhost.localdomain (PSP0)
393BE02C 8
6003 oracle
oracle@localhost.localdomain (DIA0)
393BEB04 9
6005 oracle
oracle@localhost.localdomain (MMAN)
393BF5DC 10
6007 oracle
oracle@localhost.localdomain (DBW0)
393C00B4 11
6009 oracle
oracle@localhost.localdomain (LGWR)
393C0B8C 12
6011 oracle
oracle@localhost.localdomain (CKPT)
393C1664 13
6013 oracle
oracle@localhost.localdomain (SMON)
393C213C 14
6015 oracle
oracle@localhost.localdomain (RECO)
393C2C14 15
6017 oracle
oracle@localhost.localdomain (MMON)
393C36EC 16
6019 oracle
oracle@localhost.localdomain (MMNL)
393C41C4 17
6021 oracle
oracle@localhost.localdomain (D000)
393C4C9C 18
6023 oracle
oracle@localhost.localdomain (S000)
393C5774 19
6757 oracle

oracle@localhost.localdomain (TNS V1-V3)

已选择 19 行。

分析第二个过程 mount:

告警日志中：

alter database mount

Thu Apr 02 15:33:03 2015

Successful mount of redo thread 1, with mount id 3864558315

Database mounted in Exclusive Mode

Lost write protection disabled

Completed: alter database mount

有参数文件中，找到了 control_file 的位置并锁定控制文件：

SQL> show parameter control_files;

NAME TYPE VALUE

---

control_files string /u01/app/oracle/oradata/orcl39

39/control01.ctl,

/u01/app/ora

cle/flash_recovery_area/o

rcl39

39/control02.ctl,

/u01/app/ora

cle/oradata/orcl3939/cont

rol03

.ctl

这三个控制文件的大小一样，3 个控制文件最好放在不同的物理磁盘上，往控制文件中写信息的时候并发同时写，所以 3 个控制文件的内容是相同的，但是读取的时候，只读取第一个，如果 3 个控制文件有一个出错了，oracle 就不能启动了。在实际的生产工程中，不建议放在同一磁盘上，这样不利于数据库遇到磁盘介质损坏的恢复。

控制文件中包含了联机重做日志文件和数据文件的位置。

分析第三个过程 open:

由于控制文件中记录了数据文件，日志文件的位置，检查点信息等重要的信息，在 open 阶段时，数据库根据控制文件中记录的这些信息找到这些文件，然后进行检查点及完整性检查。如果没有问题可以启动数据库，如果存在不一致或者文件丢失则需要恢复数据库。关于数据库的一致性性检查在这里不做阐述。

在这三个过程中，每个过程可以查些什么动态性能视图（动态性能视图是在数据库启动时自动创建）：

nomount:

只是启动了实例，启动实例的信息主要来自参数文件，参数文件中记录的信息可以查询，可以查：

v$parameter,v$spparameter,vsga,vsgastat,vbh,vinstance,v$option,v$version,v$process,v$session

mount:

此时控制文件被读取，和控制文件相关的视图可以查询，这要有：

v$thread,v$controlfile,v$database,v$datafile,v$logfile,v$datafile_header

open:

open 之后，所有的动态性能视图都可以查询。