

Inhaltsverzeichnis

1 Oracle 11g Installation auf Ubuntu 10.04 LTS ohne X-Windows.....	1
1.1 Voraussetzungen.....	1
1.2 Editor.....	1
1.3 Installation von zusätzlich benötigten Betriebssystempaketen.....	1
1.4 Vorbereiten des Betriebssystems für Oracle.....	2
1.4.1 Oracle Benutzer.....	2
1.4.2 Oracle Umgebung (sog. Environment) einrichten.....	2
1.4.3 Spezielle Linux Systemeinstellungen für Oracle RDBMS.....	3
1.5 X-Windows ohne lokales X verwenden.....	4
1.5.1 Variante 1.....	4
1.5.2 Variante 2.....	4
1.6 Installation des Oracle Servers.....	4
1.6.1 Auspacken der Archive.....	5
1.6.2 Start der Installation.....	5
1.6.3 Kontrolle der Installation.....	6
1.7 Installation einer Oracle Datenbank mit Skripten.....	7
1.7.1 ORADB1.sh.....	7
1.7.2 ORADB1.sql.....	7
1.7.3 CreateDB.sql.....	8
1.7.4 CreateDBFiles.sql.....	8
1.7.5 CreateDBCatalog.sql.....	8
1.7.6 xdb_protocol.sql.....	9
1.7.7 lockAccount.sql.....	9
1.7.8 postDBCreation.sql.....	9
1.7.9 init.ora.....	9
1.8 Kontrolle der Oracle Datenbankinstallation.....	10
1.9 Oracle Netzwerkkonfiguration.....	10
1.9.1 tnsnames.ora.....	11
1.9.2 sqlnet.ora.....	11
1.9.3 listener.ora.....	11
1.10 Automatischer Start der Oracle Datenbank.....	11
1.10.1 oracle.....	12
1.11 Neustart testen.....	13
1.12 Einrichtung des Klienten für andere Benutzer.....	13
1.13 FAZIT.....	13

1 Oracle 11g Installation auf Ubuntu 10.04 LTS ohne X-Windows

© Michael H.G. Schmidt (michael@schmidt2.de)

Wer schon mal eine Oracle Datenbank auf Linux installiert hat kennt das Problem: Es gibt eine Menge Linux Distributionen aber Oracle unterstützt im wesentlichen nur 2: RHEL (RedHat Enterprise Linux ©) und SLES (SuSE Linux Enterprise Server ©). Dies sind jedoch Server Versionen - der Support von aktueller Notebook/Desktop Hardware (WLAN, Sound etc. pp. ist unter diesen Linuxen verständlicherweise eher gering bis nicht vorhanden). Entwickler, Admins und enthusiastische Hobbyisten möchten jedoch Oracle manchmal auch auf "ihrem" OS (z.B.: Ubuntu) installieren.

Dies ist oft nicht ganz einfach (meistens wegen Problemen beim Verlinken der Oracle Binärdateien mit Betriebssystembibliotheken).

Das nächste Ärgernis beim installieren von Oracle RDBMS ist, daß Oracle seit einigen Jahren (und zwar seit Version 8i) ein funktionierendes X-Windows für die Installation von Oracle erwartet. Dies ist für viele Admins, die im RZ-Bereich arbeiten deshalb ärgerlich, weil X (Netzwerktechnisch) ein sehr unsicheres Protokoll fährt. Und wenn man dann noch gezwungen ist z.B. VPN & co. auf "dünnen" Leitungen zu "fahren" bekommt der alte Spruch www=World Wide Wait ein ganz neue Dimension.

Das folgende HOWTO soll dabei helfen diese Probleme zu umschiffen.

1.1 Voraussetzungen

Dieses Howto wurde unter Verwendung von Ubuntu 10.04.2 LTS (32 bit) mit Kernel 2.6.32 erstellt. Die richtige Version kann mit den folgenden Kommandos ermittelt werden:

```
cat /etc/lsb-release ; uname -r
```

Ergebnis:

```
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=10.04
DISTRIB_CODENAME=lucid
DISTRIB_DESCRIPTION="Ubuntu 10.04.2 LTS"
2.6.32-28-generic
```

Als Oracle Version wurde Oracle 11gR2 (32 bit) für Linux verwendet:

```
-rwxrwxrwx 1 michael michael 1285396902 2011-01-14 16:36 linux_11gR2_database_1of2.zip
-rwxrwxrwx 1 michael michael 995359177 2011-01-14 16:07 linux_11gR2_database_2of2.zip
```

Oracle RDBMS kann kostenfrei von [OTN](#) geladen werden. Dazu muss man sich allerdings im OTN (Oracle Technet registrieren). Die Registrierung im TechNet ist kostenlos. Auch die Mitgliedschaft ist kostenlos. Zu Test- und Weiterbildungs- und Entwicklungszwecken kann Oracle RDBMS ebenfalls kostenfrei eingesetzt werden. Allerdings ist der kommerzielle Einsatz kostenpflichtig. Über die genauen Lizenzbedingungen bitte selbst informieren (ich übernehme keine Haftung für obige Aussagen) !

- Speicher (RAM): Es sollten mind. 512MB RAM frei sein (besser 1-2 GB).
- Speicher (Swap): Es sollten ca. mind. 512MB Swap frei sein (besser 1-2 GB).
- Speicher (/tmp): Es sollten im Verz. /tmp mind. 400MB frei sein.

Kann wie folgt geprüft werden:

```
grep SwapTotal /proc/meminfo ; grep MemTotal /proc/meminfo ; df -h /tmp
```

1.2 Editor

Als Editor verwende ich normalerweise emacs. Wem emacs zu "kryptisch" und/oder zu "heavy" ist, der sollte zu nano greifen. In meinen Beispielen habe ich nano verwendet.

1.3 Installation von zusätzlich benötigten Betriebssystempaketen

Die Testumgebung wurde unter Oracle VirtualBox© als virtuelle Maschine und virtueller 16GB Platte aufgebaut. Ubuntu wurde via ISO Image neu installiert und dann auf den obigen Stand gepatcht. Danach wurden für die Installation der Virtualbox Gast Utilities noch die folgenden Pakete installiert:

```
sudo apt-get install build-essential linux-headers-generic linux-headers-2.6.32-28
sudo apt-get install libaio1 libaio-dev libelf-dev sysstat unixodbc unixodbc-dev pdksh
```

1.4 Vorbereiten des Betriebssystems für Oracle

1.4.1 Oracle Benutzer

Zuerst benötigen wir einen neuen Linux Benutzer für unseren Oracle Server. Dies ist (per Konvention) normalerweise der Benutzer "oracle". Ausserdem benötigt Oracle RDBMS eine Admin Gruppe. Diese muss immer "dba" heissen. Über die Gruppenzugehörigkeit zur Betriebssystemgruppe "dba" erhält der jeweilige Benutzer administrative Rechte auf alle Oracle Instanzen des jeweiligen Rechners. Benutzer, die nicht zur Gruppe "dba" gehören benötigen für die Oracle Administration ein Datenbankpasswort !

Es gibt noch weitere Gruppen, die laut Oracle Dokumentation vorhanden sein sollten. Das folgende Skript legt alles an:

```
sudo groupadd -g 10001 oinstall
sudo groupadd -g 10002 dba
sudo groupadd -g 10003 oper
sudo groupadd -g 10004 asm
sudo groupadd -g 10005 asmdba
sudo useradd -m -u 10001 -g oinstall -G dba,oper,asm,asmdba -d /home/oracle \
-s /bin/bash -c "Oracle software owner" oracle
```

Mit "sudo passwd oracle" ein Passwort für Benutzer Oracle setzen.

Bei Auftreten des folgenden Fehlers müssen andere Nummern für den Benutzer und/oder die Gruppe gewählt werden:

```
groupadd: GID '10001' already exists
...
...
```

Das Anlegen der Oracle Benutzer und Gruppen mit fest vorgegebener ID hat folgenden Grund: Im RZ Betrieb wird Oracle meist auf sehr vielen Maschinen installiert. Das Anlegen von Benutzern ohne Angabe einer ID führt dazu, daß Linux natürlich irgendeine Nummer verwendet. Beim Betrieb von Oracle & (z.B.) NFS führt dies aber zu vielen Problemen. Aus diesem Grund ist es besser einmalig einen Nummernkreis für Oracle Installationen festzulegen. So können später keine Konflikte mit anderen Benutzern entstehen. Natürlich kann man dies auch mit Hilfe eines Verzeichnisdienstes (z.B. NIS/NIS+ oder LDAP) realisieren.

Kontrolle:

```
id oracle
```

Ausgabe:

```
uid=10001(oracle) gid=10001(oinstall) Gruppen=10001(oinstall),10002(dba),10003(oper),10004(asm),10005(asmdba)
```

Benutzer Oracle für "sudo" ohne Passwort berechtigen. Mit folgendem Kommando die /etc/sudoers editieren:

```
sudo visudo
```

...und folgendes eintragen:

```
oracle ALL=NOPASSWD: ALL
```

1.4.2 Oracle Umgebung (sog. Environment) einrichten

Zum Benutzer Oracle wechseln:

```
sudo su - oracle
```

".profile" ('nano .profile' eingeben) editieren und am Ende folgendes einfügen:

```
### BEGIN Oracle Environment ###
export LANG=en_US.UTF-8 # Zeichensatz (notwendig für ü,ä,ö,ß usw.)
export LC_MESSAGES=C # Fehlermeldungen von Kommandos in Englisch
export EDITOR="nano -nw" # Variable "EDITOR" wird oft von sqlplus,cron u.a. verwendet.
export VISUAL="nano -nw" # s.o.
```

1.3 Installation von zusätzlich benötigten Betriebssystempaketen

```
alias vi='nano -nw'      # "nano" only;-)
alias vim='nano -nw'    # dito.

export ORACLE_BASE=/opt/oracle/product      # Oracle Basis Verzeichnis. Hier liegen z.B. die Server Protokolle
export ORACLE_HOME=$ORACLE_BASE/ora11gR2   # Oracle "Home" Verzeichnis. Hier liegen die Binaries.
export ORACLE_SID=ORADB1                    # Name der Instanz (im Normalfall auch der Name der Datenbank)
export PATH=$ORACLE_HOME/bin:$PATH         # Suchpfad erweitern (damit z.B. sqlplus gefunden wird.)
export ORACLE_TERM=xterm                   # optional.
export TNS_ADMIN=$ORACLE_HOME/network/admin # Pfad zur Oracle Netzwerkkonfiguration (optional).
export LD_LIBRARY_PATH=$ORACLE_HOME/lib    # i.d.R. nicht notwendig. schadet aber auch nicht.

ulimit -s 32768
### END Oracle environment ###
```

1.4.3 Spezielle Linux Systemeinstellungen für Oracle RDBMS

- Firewall: Sollte ein Connect von anderen Systemen gewünscht sein, muss Port 1521 (Standardport Oracle Netzwerkdienst) in beide Richtungen freigeschaltet werden.
- SELinux: Es sollte auf jeden Fall deinstalliert oder deaktiviert sein (man kann Oracle evtl. mit einigem Aufwand und ein bisschen "Jugend forscht" sicherlich auch mit SELinux betreiben. Aber soweit ich weiss ist es nicht supported und m.E. nach auch nicht sinnvoll, da im Betrieb die kuriosesten Fehler auftreten können, deren Ursache dann nur nach langer Suche oder gar nicht auf SELinux zurückzuführen sind).

In der Datei **/etc/security/limits.conf** (sudo nano /etc/security/limits.conf) folgendes am Ende einfügen:

```
### BEGIN ORACLE RAC shell limits ###
oracle soft nproc 2047
oracle hard nproc 16384
oracle soft nofile 4096
oracle hard nofile 65536
oracle soft memlock 3145728
oracle hard memlock 3145728
### END ORACLE RAC shell limits ###
```

Diese Einstellungen werden beim erzeugen von Prozessen des Benutzers Oracle aktiv. Das heisst: Wenn man sich bereits als Benutzer Oracle in den Desktop eingeloggt hat, dann sollte man sich nochmal neu einloggen.

In der Datei **/etc/sysctl.conf** (sudo nano /etc/sysctl.conf) folgendes einfügen:

```
### BEGIN ORACLE network settings ###

# default size of socket receive buffer in bytes (=4MB)
net.core.rmem_default=4194304

# maximum size of socket receive buffer in bytes (=4MB)
net.core.rmem_max=4194304

# default size of socket send buffer in bytes (=256KB)
net.core.wmem_default=262144

# maximum size of socket send buffer in bytes (=1MB)
net.core.wmem_max=1048576

# set local IP port range
net.ipv4.ip_local_port_range=9000 65500

### END ORACLE network settings ###

### BEGIN ORACLE kernel settings ###

# file handles
fs.file-max=6815744

# asynchronous IO
fs.aio-max-nr=1048576

# shared memory maximum allocation unit (2GB-1)
kernel.shmmax=2147483647

# SEMMSL SEMMNS SEMOPM SEMMNI
```

1.4.2 Oracle Umgebung (sog. Environment) einrichten

```
kernel.sem=250 32000 100 128

kernel.panic=5
kernel.shmmni=4096
kernel.shmall=2097152

# max. processes in OS including threads !
kernel.pid_max=32768

### END ORACLE kernel settings ###
```

Die Settings in /etc/sysctl.conf mit folgendem Befehl aktivieren:

```
sudo sysctl -p
```

Verzeichnisse für die Oracle Binaries und sonstigen Daten anlegen:

```
sudo mkdir -p /opt/oracle/product/ora11gR2
sudo chown -R oracle.oinstall /opt/oracle
```

WICHTIG ! Die Kernel Einstellungen für Oracle wurden für einen 32-Bit Kern erstellt. Die Einstellung "shmmax" z.B. sollte bei einem 64-Bit Kern mindestens auf 4GB-1 gesetzt werden. Bei einem 64-Bit System wird der Oracle Installer diese Einstellung auch bemängeln, wenn sie dann noch auf 2GB-1 steht. "pid_max" sollte bei 64-Bit Systemen auf 128K-1 (131071) gesetzt werden.

1.5 X-Windows ohne lokales X verwenden

Die Oracle Installation verläuft im Normalfall grafisch. Wir haben jedoch vor Oracle ohne Verwendung von X-Windows zu installieren ! Für die hartnäckigen Kommandozeilenhasser ist die folgende Kurzanleitung dieses Kapitels gedacht.

1.5.1 Variante 1

Auf einer Maschine, auf der kein Desktop läuft, oder keinen Desktop installiert hat muss man die Anzeige auf eine andere Maschine umleiten. Auf dieser zweiten Maschine muss dann ein X-Windows (Gnome o.ä.) laufen. Nach dem einloggen als Benutzer Oracle stellt man die Anzeige wie folgt ein:

```
export DISPLAY=xxx.xxx.xxx.xxx:0.0
```

Für xxx.xxx.xxx.xxx ist die IP der anderen Maschine einzusetzen. Auf der anderen Maschine muss nun der remote connect von X-Windows Programmen erlaubt werden. Unter Linux erledigt man das, indem man eine Shell öffnet und ein **xhost +** absetzt. Das Kommando quittiert dies mit:

```
access control disabled, clients can connect from any host
```

Auf dem Oracle Server testen wir dies mit "xterm". Shell öffnen und als User Oracle "xterm" eingeben. Jetzt sollte sich ein Fenster auf der anderen Maschine öffnen.

1.5.2 Variante 2

Umleiten der X-Fenster des Benutzers Oracle auf den eigenen, lokalen (!) Benutzer.

Im Normalfall hat bei einer Ubuntu Desktop Installation die Variable "DISPLAY" den Wert ":0.0". Der Grund ist: bei lokalen Verbindungen benötigt X keine IP-Adresse, wohl aber die Nummer des Displays (0.0). Wir können unter dem Benutzer Oracle das Display ebenfalls so einstellen, müssen aber darauf achten, daß der Benutzer Oracle auch die Rechte hat Fenster auf dem aktuellen Desktop zu öffnen.

Wir gehen deshalb wie folgt vor: Unter dem aktuellen Benutzer das Kommando **xhost +** absetzen. Dann unter dem Benutzer Oracle die Variable DISPLAY richtig einstellen: "**export DISPLAY=:0.0**". Danach kann mit **xterm &** geprüft werden, ob alles richtig eingestellt wurde (jetzt müsste sich ein neues Terminal Fenster öffnen, das aber dem Benutzer Oracle gehört !).

WICHTIGER HINWEIS: Den Oracle Installer ruft man dann im weiteren Verlauf ohne die Schalter "-silent -force -responseFile" auf.

1.6 Installation des Oracle Servers

Ab jetzt führen wir wieder alles mit dem neu angelegten "Oracle" Benutzer durch:

1.4.3 Spezielle Linux Systemeinstellungen für Oracle RDBMS

```
sudo su - oracle
```

1.6.1 Auspacken der Archive

```
unzip linux_11gR2_database_1of2.zip
unzip linux_11gR2_database_2of2.zip
```

1.6.2 Start der Installation

Damit die Installation vollautomatisch abläuft müssen wir zuerst im Verz. "database" eine sog. Antwortdatei (=Responsefile) für den Oracle Installer anlegen. Mit dem Editor Nano die Datei "ora11gR2.rsp" anlegen:

```
DECLINE_SECURITY_UPDATES=true
INVENTORY_LOCATION=/opt/oracle/oraInventory
ORACLE_BASE=/opt/oracle/product
ORACLE_HOME=/opt/oracle/product/ora11gR2
ORACLE_HOSTNAME=user-laptop
oracle.install.db.config.starterdb.automatedBackup.enable=false
oracle.install.db.config.starterdb.control=DB_CONTROL
oracle.install.db.config.starterdb.dbcontrol.enableEmailNotification=false
oracle.install.db.config.starterdb.enableSecuritySettings=true
oracle.install.db.config.starterdb.gridcontrol.gridControlServiceURL=
oracle.install.db.config.starterdb.installExampleSchemas=false
oracle.install.db.config.starterdb.memoryOption=false
oracle.install.db.config.starterdb.type=GENERAL_PURPOSE
oracle.install.db.DBA_GROUP=dba
oracle.install.db.InstallEdition=SEONE
oracle.install.db.isCustomInstall=false
oracle.install.db.OPER_GROUP=oper
oracle.install.option=INSTALL_DB_SWONLY
oracle.install.responseFileVersion=/oracle/install/rspfmt_dbinstall_response_schema_v11_2_0
SECURITY_UPDATES_VIA_MYORACLESUPPORT=false
SELECTED_LANGUAGES=en,de
UNIX_GROUP_NAME=oinstall
```

Wir installieren nur die Software ("INSTALL_DB_SWONLY") der Oracle Edition "SE one" ("oracle.install.db.InstallEdition=SEONE")

Jetzt starten wir die Installation wie folgt:

```
cd database
chmod 0755 runInstaller
unset DISPLAY
./runInstaller -ignorePrereq -ignoreSysPrereqs -silent -force -responseFile `pwd`/ora11gR2.rsp
```

ACHTUNG !!! ACHTUNG !!! ACHTUNG

Das Kommando "**unset DISPLAY**" ist extrem wichtig an dieser Stelle. Ansonsten kommt es zu folgendem Fehler und der Oracle Installierer verweigert seine Arbeit (!) :

```
Exception in thread "main" java.lang.NoClassDefFoundError
  at java.lang.Class.forName0(Native Method)
  at java.lang.Class.forName(Class.java:164)
  at java.awt.Toolkit$2.run(Toolkit.java:821)
  at java.security.AccessController.doPrivileged(Native Method)
  at java.awt.Toolkit.getDefaultToolkit(Toolkit.java:804)
  at javax.swing.UIManager.initialize(UIManager.java:1262)
  at javax.swing.UIManager.maybeInitialize(UIManager.java:1245)
  at javax.swing.UIManager.getUI(UIManager.java:851)
  at javax.swing.JPanel.updateUI(JPanel.java:104)
  at javax.swing.JPanel.<init>(JPanel.java:64)
  at javax.swing.JPanel.<init>(JPanel.java:87)
  at javax.swing.JPanel.<init>(JPanel.java:95)
  at oracle.sysman.oii.oiiif.oiiifo.OiiifoOCMUI.<init>(OiiifoOCMUI.java:125)
  at oracle.sysman.oii.oiiif.oiiifo.OiiifoOCMInterfaceManager.<init>(OiiifoOCMInterfaceManager.java:79)
  at oracle.sysman.oii.oiiif.oiiifo.OiiifoOCMInterfaceManager.getInstance(OiiifoOCMInterfaceManager.java:124)
  at oracle.install.iwv.db.driver.DBInstaller.run(DBInstaller.java:123)
  at oracle.install.commons.util.Application.startup(Application.java:869)
  at oracle.install.commons.flow.FlowApplication.startup(FlowApplication.java:164)
  at oracle.install.commons.flow.FlowApplication.startup(FlowApplication.java:181)
  at oracle.install.commons.base.driver.common.Installer.startup(Installer.java:265)
```

```
at oracle.install.iwv.db.driver.DBInstaller.startup(DBInstaller.java:114)
at oracle.install.iwv.db.driver.DBInstaller.main(DBInstaller.java:132)
```

Wenn alles ok ist antwortet der Oracle Installierer wie folgt:

```
Starting Oracle Universal Installer...
Checking Temp space: must be greater than 80 MB.   Actual 11488 MB   Passed
Checking swap space: must be greater than 150 MB.   Actual 729 MB   Passed
Preparing to launch Oracle Universal Installer from /tmp/OraInstall2011-02-17_08-35-02PM.
Please wait ...You can find the log of this install session at:
/opt/oracle/oraInventory/logs/installActions2011-02-17_08-35-02PM.log
```

Mit dem Kommando:

```
tail -f /opt/oracle/oraInventory/logs/installActions2011-02-17_08-35-02PM.log
```

kann nun in einem zweiten Terminal Fenster die Installation überwacht werden...

Nach erfolgreichem Abschluss der Installation antwortet der Oracle Installierer wie folgt:

```
The following configuration scripts need to be executed as the "root" user.
#!/bin/sh
#Root scripts to run

/opt/oracle/oraInventory/orainstRoot.sh
/opt/oracle/product/ora11gR2/root.sh
To execute the configuration scripts:
1. Open a terminal window
2. Log in as "root"
3. Run the scripts
4. Return to this window and hit "Enter" key to continue
```

Successfully Setup Software

Am Schluss müssen dann noch als Benutzer "root" zwei Skripte ausgeführt werden (siehe oben !). Allerdings gibt es genau an dieser Stelle ein Problem:

```
/opt/oracle/oraInventory/orainstRoot.sh: 8: /bin/awk: not found
```

Ursache für diesen Fehler ist, daß in dem von Oracle generierten Skript "awk" nur in einem einzigen Pfad (nämlich /bin) gesucht wird.

Das Problem lässt sich leicht fixen mit:

```
sudo ln -s /usr/bin/awk /bin/awk
```

Jetzt noch die beiden Oracle Skript ausführen:

```
sudo /opt/oracle/oraInventory/orainstRoot.sh
sudo /opt/oracle/product/ora11gR2/root.sh
```

1.6.3 Kontrolle der Installation

In der Protokolldatei sind die letzten Meldungen:

```
INFO: Terminated all background operations
INFO: Successfully executed the flow in SILENT mode
INFO: Finding the most appropriate exit status for the current application
INFO: Exit Status is 0
INFO: Shutdown Oracle Database 11g Release 2 Installer
```

Hier ist vor allem darauf zu achten, daß der Exit Status "0" ist.

Im Verzeichnis /opt/oracle/product/ora11gR2 sollten nun so einige neue Verzeichnisse zu sehen sein ;-). Ein "du -sh /opt/oracle/product/ora11gR2" sollte etwa 3,5GB melden !

Wir testen noch mit einem kurzen Aufruf von "sqlplus", ob wir die Binaries im frisch installierten Oracle Heimatverzeichnis ausführen können. Der Aufruf von sqlplus in der Form "sqlplus /NOLOG" sollte in etwa folgende Ausgabe liefern:

1.6.2 Start der Installation

SQL*Plus: Release 11.2.0.1.0 Production on Fri Feb 18 16:14:52 2011

Copyright (c) 1982, 2009, Oracle. All rights reserved.

SQL>

Falls man die Installation wiederholen möchte sollte man vorher unbedingt alles im Pfad /opt/oracle löschen und das Verzeichnis /opt/oracle/product neu anlegen (Nicht vergessen: Eigentümer auf "oracle.oinstall" mit chown setzen, falls man das als root gemacht hat !).

HINWEIS: Aus mir nicht bekannten Gründen hat der Installierer scheinbar ein Verzeichnis im ORACLEHOME vergessen. In diesem Fall wird beim start von "sqlplus" das Verz. **oradiag**oracle angelegt. Irgendwo dort gibt es dann auch ein **sqlnet.log** Protokoll, in dem Oracle auch meldet, welches Verz. fehlt. Wie auch immer - ein simples:

```
rm -rf oradiag_oracle
mkdir $ORACLE_HOME/log/diag/clients
```

verhindert, daß das Problem erneut auftritt.

1.7 Installation einer Oracle Datenbank mit Skripten

Nachdem nun der Oracle Server installiert ist benötigen wir noch eine Datenbank. Oracle Datenbanken kann man auf 2 Arten erzeugen:

1. mit dem grafischen Werkzeug "dbca" (Database Creation Assistant)
2. und per Skript

Da wir auf den Einsatz von X-Windows verzichten möchten kommt nur Möglichkeit 2 in Frage. Der einfachste Weg ist es, den dbca auf einem anderen System (mit funktionierendem X) aufzurufen und die Skripte vom dbca erzeugen zu lassen. Ich habe dies bereits getan und werde die vom dbca erzeugten Skripte hier auflisten.

Alle Schritte werden mit dem Benutzer Oracle ausgeführt. Die Skripte koennen mit "nano" gespeichert werden und sollten im Ordner "\$HOME/scripts" liegen.

Skriptordner anlegen:

```
mkdir $HOME/scripts
cd $HOME/scripts
```

1.7.1 ORADB1.sh

```
ORACLE_SID=ORADB1; export ORACLE_SID
ORACLE_BASE=/opt/oracle/product ; export ORACLE_BASE
ORACLE_HOME=$ORACLE_BASE/ora11gR2 ; export ORACLE_HOME
PATH=$ORACLE_HOME/bin:/usr/local/bin:/usr/bin:/bin ; export PATH
mkdir -p $ORACLE_BASE/diag/rdbms/oradb1/ORADB1/adump
mkdir -p $ORACLE_BASE/flash_recovery_area
mkdir -p $ORACLE_BASE/ora11gR2/dbs
mkdir -p $ORACLE_BASE/oradata
rm -f makedb.log
[ ! -f ORADB1.sql ] && {
    echo "ORADB1.sql nicht gefunden !"
    exit 1
}
$ORACLE_HOME/bin/sqlplus /nolog @ORADB1.sql
echo "Die folgende Zeile bitte in /etc/oratab einfüegen: \"ORADB1:/opt/oracle/product/ora11gR2:Y\""
```

1.7.2 ORADB1.sql

```
set verify off
ACCEPT sysPassword CHAR PROMPT 'Enter new password for SYS: ' HIDE
ACCEPT systemPassword CHAR PROMPT 'Enter new password for SYSTEM: ' HIDE
host $ORACLE_HOME/bin/orapwd file=$ORACLE_HOME/dbs/orapwORADB1 entries=10 force=y nosysdba=no
@CreateDB.sql
@CreateDBFiles.sql
@CreateDBCatalog.sql
@xdb_protocol.sql
@lockAccount.sql
@postDBCreation.sql
```

1.6.3 Kontrolle der Installation

1.7.3 CreateDB.sql

```
SET VERIFY OFF
connect "SYS"/"&&sysPassword" as SYSDBA
set echo on
spool makedb.log append
startup nomount pfile='./init.ora';
CREATE DATABASE "ORADB1"
  MAXINSTANCES 1
  MAXLOGHISTORY 1
  MAXLOGFILES 16
  MAXLOGMEMBERS 1
  MAXDATAFILES 100
  DATAFILE SIZE 32M AUTOEXTEND ON NEXT 8M MAXSIZE 1G
  SYSAUX DATAFILE SIZE 32M AUTOEXTEND ON NEXT 8M MAXSIZE 1G
  SMALLFILE DEFAULT TEMPORARY TABLESPACE "TEMP" TEMPFILE SIZE 32M
  AUTOEXTEND ON NEXT 8M MAXSIZE 2G EXTENT MANAGEMENT LOCAL
  SMALLFILE UNDO TABLESPACE "UNDOTBS1" DATAFILE SIZE 32M
  AUTOEXTEND ON NEXT 8M MAXSIZE 2G EXTENT MANAGEMENT LOCAL
  CHARACTER SET WE8ISO8859P15
  NATIONAL CHARACTER SET UTF8
  LOGFILE
    GROUP 1 SIZE 64M,
    GROUP 2 SIZE 64M,
    GROUP 3 SIZE 64M
  USER SYS IDENTIFIED BY "&&sysPassword"
  USER SYSTEM IDENTIFIED BY "&&systemPassword";

set linesize 2048;
column ctl_files NEW_VALUE ctl_files;
select concat('control_files='', concat(replace(value, ', ', ',', '')) || ', ''') ctl_files
  from v$parameter where name = 'control_files';
host cp -v ./init.ora $ORACLE_HOME/dbs/initORADB1.ora;
host echo &ctl_files >>$ORACLE_HOME/dbs/initORADB1.ora;
spool off
```

1.7.4 CreateDBFiles.sql

```
SET VERIFY OFF
connect "SYS"/"&&sysPassword" as SYSDBA
set echo on
spool makedb.log append
CREATE SMALLFILE TABLESPACE "USERS" LOGGING DATAFILE SIZE 5M AUTOEXTEND ON NEXT 1M
  MAXSIZE 2G EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;
ALTER DATABASE DEFAULT TABLESPACE "USERS";
spool off
```

1.7.5 CreateDBCatalog.sql

```
SET VERIFY OFF
connect "SYS"/"&&sysPassword" as SYSDBA
set echo on
spool makedb.log append
@?/rdms/admin/catalog.sql;
@?/rdms/admin/catblock.sql;
@?/rdms/admin/catproc.sql;
@?/rdms/admin/catoctk.sql;
@?/rdms/admin/owminst.plb;

connect "SYSTEM"/"&&systemPassword"
@?/sqlplus/admin/publd.sql;
spool off

connect "SYSTEM"/"&&systemPassword"
set echo on
spool makedb.log append
@?/sqlplus/admin/help/hlpbld.sql helpus.sql;
spool off
```

1.7.6 xdb_protocol.sql

```
SET VERIFY OFF
connect "SYS"/"&&sysPassword" as SYSDBA
set echo on
spool makedb.log append
@?/rdbs/admin/catqm.sql &&sysPassword SYSAUX TEMP yes;
spool off
```

1.7.7 lockAccount.sql

```
SET VERIFY OFF
set echo on
spool makedb.log append
BEGIN
  FOR item IN (
    SELECT USERNAME FROM DBA_USERS WHERE ACCOUNT_STATUS IN ('OPEN', 'LOCKED', 'EXPIRED')
    AND USERNAME NOT IN ('SYS','SYSTEM')
  )
  LOOP
    dbms_output.put_line('Locking and Expiring: ' || item.USERNAME);
    execute immediate
      'alter user ' ||
      sys.dbms_assert.enquote_name(sys.dbms_assert.schema_name(item.USERNAME),false) ||
      ' password expire account lock' ;
  END LOOP;
END;
/
spool off
```

1.7.8 postDBCreation.sql

```
SET VERIFY OFF
connect "SYS"/"&&sysPassword" as SYSDBA
set echo on
spool makedb.log append
execute DBMS_AUTO_TASK_ADMIN.disable();
select 'utl_recomp_begin: ' || to_char(sysdate, 'HH:MI:SS') from dual;
execute utl_recomp.recomp_serial();
select 'utl_recomp_end: ' || to_char(sysdate, 'HH:MI:SS') from dual;
shutdown immediate;

connect "SYS"/"&&sysPassword" as SYSDBA
startup mount PFILE='?/dbs/initORADB1.ora';
alter database archivelog;
alter database open;

connect "SYS"/"&&sysPassword" as SYSDBA
set echo on
create spfile='?/dbs/spfileORADB1.ora' FROM pfile='?/dbs/initORADB1.ora';
shutdown immediate;

connect "SYS"/"&&sysPassword" as SYSDBA
startup ;
spool off
exit;
```

1.7.9 init.ora

```
audit_file_dest=/opt/oracle/product/diag/rdbms/oradb1/ORADB1/adump
audit_trail=db
compatible=11.2.0.0.0
db_block_size=8192
db_create_file_dest=/opt/oracle/product/oradata
db_domain=""
db_name=ORADB1
db_recovery_file_dest=/opt/oracle/product/flash_recovery_area
db_recovery_file_dest_size=4G
diagnostic_dest=/opt/oracle/product
dispatchers="(PROTOCOL=TCP) (SERVICE=ORADB1XDB)"
log_archive_format=%t_%s_%r.dbf
memory_target=400M
```

1.7.6 xdb_protocol.sql

```
nls_language="AMERICAN"
nls_territory="AMERICA"
open_cursors=300
processes=150
remote_login_passwordfile=EXCLUSIVE
undo_tablespace=UNDOTBS1
```

Shell Skript ausführbar machen und starten:

```
cd scripts
chmod 0755 ORADB1.sh
./ORADB1.sh
```

Das Skript fragt zunächst nach neuen Passwörtern für SYS & SYSTEM, sowie noch einmal nach dem gerade vergebenen SYS Passwort und startet dann die Oracle Instanz. Danach wird die Datenbank angelegt und die Dictionaries(Systemkataloge) werden gefüllt.

Zunächst starten die Skripte eine Datenbankinstanz und legen eine leere Datenbank an. Danach huschen in schneller Folge die SQL Ausgaben von Oracle Skripten über den Schirm, die die Kataloge anlegen. Dieser Vorgang dauert zwischen 30-60 Minuten (je nach Leistung des Rechners).

Die Skripte legen ausserdem die Protokolldatei "makedb.log" an, die später noch auf Fehler kontrolliert werden muss.

1.8 Kontrolle der Oracle Datenbankinstallation

Sofern alles OK war sieht das Ende der Installation ungefähr so aus:

```
SQL> connect "SYS"/"&&sysPassword" as SYSDBA
Connected to an idle instance.
SQL> startup ;
ORACLE instance started.

Total System Global Area  418484224 bytes
Fixed Size                 1336932 bytes
Variable Size             255855004 bytes
Database Buffers          155189248 bytes
Redo Buffers               6103040 bytes
Database mounted.
Database opened.
SQL> spool off
SQL> exit;
Disconnected from Oracle Database 11g Release 11.2.0.1.0 - Production
Die folgende Zeile bitte in /etc/oratab einfüegen: "ORADB1:/opt/oracle/product/ora11gR2:Y"
```

Jetzt muss noch die Protokolldatei überprüft werden. Das macht man am besten mit folgendem Kommando:

```
egrep "^ORA-|^SP2-" *log | grep -v "does not"
```

Die folgenden Meldungen können ignoriert werden:

```
ORA-01921: role name 'EXP_FULL_DATABASE' conflicts with another user or role name
ORA-01921: role name 'IMP_FULL_DATABASE' conflicts with another user or role name
ORA-00439: feature not enabled: Bit-mapped indexes
ORA-00955: name is already used by an existing object
```

Alle anderen Fehler müssen auf jeden Fall im Logfile "makedb.log" gesucht, kontrolliert und evtl. behoben werden.

Jetzt können wir uns mit "sqlplus / as sysdba" in der neuen Datenbank einloggen. Die Abfrage "select name from v\$database" sollte als Ergebnis "ORADB1" liefern !

1.9 Oracle Netzwerkkonfiguration

Damit unsere Datenbank auch über Netzwerk erreichbar ist (Standardmäßig über Port 1521) müssen wir jetzt noch eine TNS-Net Konfiguration erstellen. Dazu benötigt man 3 Dateien:

1. tnsnames.ora = ist eine Art "hosts" Datei. Sie dient der Auflösung von Instanznamen zu Hosts und Datenbanken
2. sqlnet.ora = ist eine Art "nsswitch.conf". Diese Datei bestimmt, in welcher Reihenfolge die Namensauflösung durchgeführt wird.

3. listener.ora = konfiguriert den Oracle Horchprozess (= "Listener"). Legt Portnummer etc. pp. fest.

Alle Dateien werden in \$ORACLE_HOME/network/admin (= \$TNS_ADMIN) erstellt:

1.9.1 tnsnames.ora

```
ORADB1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = 127.0.0.1) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = ORADB1)
    )
  )
```

1.9.2 sqlnet.ora

```
NAMES.DIRECTORY_PATH= (TNSNAMES)
```

1.9.3 listener.ora

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROCL))
      (ADDRESS = (PROTOCOL = TCP) (HOST = localhost) (PORT = 1521))
    )
  )
```

Jetzt noch den Horcher starten:

```
lsnrctl start
```

Am Schluss testen wir noch, ob der Listener und unsere Netzwerkkonfiguration korrekt arbeitet:

```
tnsping ORADB1
```

Sollte folgendes Ergebnis liefern:

```
TNS Ping Utility for Linux: Version 11.2.0.1.0 - Production on 20-FEB-2011 02:02:36
Copyright (c) 1997, 2009, Oracle. All rights reserved.

Used parameter files:
/opt/oracle/product/orallgR2/network/admin/sqlnet.ora

Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP) (HOST = 127.0.0.1)
(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME = ORADB1)))
OK (20 msec)
```

1.10 Automatischer Start der Oracle Datenbank

Um den automatischen Start und Stop der Datenbank zu aktivieren müssen zwei Dinge erledigt werden:

1. Eintrag in /etc/oratab vornehmen
2. Skript in /etc/init.d installieren.

zu 1): mit nano "/etc/oratab" öffnen und folgenden Eintrag hinzufügen:

```
ORADB1:/opt/oracle/product/orallgR2:Y
```

zu 2) folgendes Skript mit "nano" als /tmp/oracle anlegen und ausführen:

```
sudo bash oracle install
```

1.10.1 oracle

```
#!/bin/sh
# -----
# email: michael@schmidt2.de
# @(#) $Id: Oracle_11g_Ubuntu.txt 70 2011-02-24 05:16:18Z shellforger $
# Init Script for oracle databases, Linux Version
# -----

OSTYPE=`uname -s | tr '[:upper:]' '[:lower:]'`
progrname=`basename $0`
whoami=`id | cut -d '(' -f2 | cut -d ')' -f1`
appname='ORACLE rdbms'

[ $OSTYPE = "linux" ] && {
    ECHO='echo'
    AWK='nawk'
    RCPATH=/etc
}

[ "$OSTYPE" != "@linux" ] && {
    $ECHO "FATAL: platform \"$OSTYPE\" not supported !"
    exit 1
}

usage () {
    cat <<-HERE

usage: $progrname { <start|stop|restart|install> }

HERE
    exit 1
}

logc () {
    $ECHO "${progrname}: $1"
}

self_install () {
    if [ $whoami != 'root' ] ; then
        $ECHO "ERROR: [self_install]: only root can do this."
        exit 1
    fi

    $ECHO "INFO: deleting old Skripts for $appname..."
    rm -f $RCPATH/rc0.d/K00$progrname
    rm -f $RCPATH/rc2.d/S99$progrname
    rm -f /etc/init.d/$progrname
    $ECHO "INFO: copying programs..."
    cp $0 /etc/init.d/$progrname 2>/dev/null
    chmod 0755 /etc/init.d/$progrname 2>/dev/null
    $ECHO "INFO: generating symlinks..."
    ln -s /etc/init.d/$progrname $RCPATH/rc0.d/K00$progrname 2>/dev/null
    ln -s /etc/init.d/$progrname $RCPATH/rc2.d/S99$progrname 2>/dev/null
    exit 0
}

daemon_restart () {
    daemon_stop
    daemon_start
}

daemon_start () {
    if [ $whoami != 'root' ] ; then
        logc "INFO: [start] only root can do this."
        exit 1
    fi
    logc "INFO: starting [$appname]"
    su - oracle -c "dbstart ; lsnrctl start"
}

daemon_stop () {
    if [ $whoami != 'root' ] ; then
        logc "INFO: [stop] only root can do this."
    fi
}
```

```

        exit 1
    fi
    logc "INFO: stopping [$appname]."
    su - oracle -c "lsnrctl stop ; dbshut"
}

# -----
# MAIN
# -----

[ $# -ne 1 ] && usage
[ $1 = "-?" -o $1 = "-h" -o $1 = "-H" ] && usage
[ "$1" != "start" -a "$1" != "stop" -a "$1" != "restart" -a "$1" != "install" ] && usage

mode=$1
[ $mode = "install" ] && self_install
eval daemon_$mode
#

```

1.11 Neustart testen

Dazu bitte folgende Kommandos absetzen:

```
sudo /etc/init.d/oracle restart
```

Jetzt sollte die Oracle DB sauber durchgestartet worden sein...

1.12 Einrichtung des Klienten für andere Benutzer

Um den Oracle Klienten für andere Benutzer verfügbar zu machen einfach die folgenden Zeilen in die ".bashrc" oder ".profile" des jeweiligen Benutzers eintragen:

```

export ORACLE_BASE=/opt/oracle/product      # Oracle Basis Verzeichnis. Hier liegen z.B. die Server Protokolle
export ORACLE_HOME=$ORACLE_BASE/ora11gR2   # Oracle "Home" Verzeichnis. Hier liegen die Binaries.
export ORACLE_SID=ORADB1                   # Name der Instanz (im Normalfall auch der Name der Datenbank)
export PATH=$ORACLE_HOME/bin:$PATH         # Suchpfad erweitern (damit z.B. sqlplus gefunden wird.)

```

Danach neu einloggen. Anschliessend steht "sqlplus" im Suchpfad zur Verfügung. Die Datenbank kann nun mit

```
sqlplus system/<systemPasswort>@ORADB1
```

genutzt werden. Als User Oracle reicht ein:

```
sqlplus / as sysdba
```

um die Datenbank zu administrieren.

Man sollte sich aber einen neuen Benutzer einrichten und möglichst nicht als SYS oder SYSTEM mit der Datenbank arbeiten...

1.13 FAZIT

Eine Oracle Installation vollkommen ohne GUI (X-Windows oder Windowfs) ist keine Hexerei - im Gegenteil: Das Ganze ist eigentlich sehr viel simpler als eine GUI basierte Installation. Vor allem kann man im Fehlerfall sehr viel einfacher erkennen, worin das Problem besteht. Bei einer Oracle GUI Installation habe ich in den vergangenen 10 Jahren so einige Haarsträubende Probleme des grafischen Installierers gesehen. Nicht selten blieb das Teil ohne Fehlermeldung stehen und verweigerte jegliche weitere Arbeit. Abstürze des OUI (Oracle Universal Installierer) sind ebenfalls keine Seltenheit. M.E. nach ist dieses sogenannte Oracle Werkzeug ein heisser Kandidat für die Ablage P. Dieses Howto hilft hoffentlich dem interessierten Leser ebenfalls in Zukunft den OUI zu meiden.