

INSTALLING ORACLE 8i RELEASE 2 (8.1.6.1) ON RED HAT LINUX 6.x/7.x

Brian Lomasky, DBA Solutions, Inc.

Linux is an exciting new platform, since it is very stable, requires only a fraction of the hardware resources that Windows NT requires, is a true multitasking operating system, and oh yes, it's free!

Since Oracle 8i is now supported on the Intel/Linux platform, database developers and administrators can now take advantage of a robust and flexible solution for their database environments.

This paper will describe the operations which must be performed in order to perform a successful installation of Oracle 8i on Red Hat Linux 6.x.

It will also compare the desktop and tools environments between a Linux and a Windows NT system.

HARDWARE REQUIREMENTS:

RAM:

128 megs RAM (preferably 256 megs) is required to install and run Oracle 8i on Linux. 256 megs or more may be required to enable Java VM. You can see how much memory is installed by typing:

```
cat /proc/meminfo
```

and looking at the MemTotal value. The Linux 2.2 kernel can support up to 2 gigs RAM.

HARD DISK SPACE:

About 800 megs of disk space are needed to install Oracle 8i. (A minimal installation requires about 693 megs). Check this by typing:

```
df -k
```

This size does not include the size of the database.

(The Linux 2.2 kernel scales well up to 4 CPUs. POSIX thread support takes advantage of SMP systems. Linux can achieve SMP efficiencies approaching 70%).

SOFTWARE REQUIREMENTS:

OPERATING SYSTEM:

Red Hat 6.0, 6.1, or 6.2 (Linux Kernel 2.2)

For Red Hat 7.0, see the "Special Note For Red Hat 7.0" section at the end of this document.

(The following Linux distributions are also supported:

SuSe 6.3 or 6.4

TurboLinux 6.0.2

Caldera eServer 2.3

(The eServer requires the following patches from

<ftp://ftp.caldera.com/pub/.eServer/updates/2.3/current/RPMS/:glibc-2.1.2-2S.i386.rpm>,
glibc-devel-2.1.2-2S.i386.rpm, glibc-devel-static-2.1.2-2S.i386.rpm, glibc-localedata-2.1.2-
2S.i386.rpm)

(Red Hat seems to better supported by Oracle than other Linux distributions).

When performing the Red Hat installation:

1) Partition the hard disk as you see fit.

a) Make one or more mount points for the Oracle software and the database. You can optionally choose to adhere to the Oracle OFA standard. In order to achieve the highest level of performance, each of the mount points should be on separate partitions on distinct hard drives.

b) The total size of your swap partitions should be equivalent to 2 (to 3) times the installed RAM, or 400MB, whichever is larger.

2) If you decide to perform a Red Hat CUSTOM installation, be sure to install:

a) X Windows.

b) GNOME or KDE desktop.

c) Development packages.

3) Kernel memory parameters

The Oracle installation manual lists some kernel parameters which Oracle says need to be changed before installing the software.

The default kernel parameters that Red Hat uses when creating their kernel is adequate for Oracle, as most of them meet or exceed Oracle's requirements.

The only parameter that you might have to change is SHMMAX. The Red Hat value for this parameter is set to 32 meg (33554432). If your system has more than 64 meg of RAM and you really want a large SGA, you might consider increasing this parameter's value (although due to the complexity of doing so, this is not recommended unless you are experienced in rebuilding the Linux kernel).

If you want to change the value of SHMMAX, you must install the Red Hat kernel source RPMS from the Red Hat installation CD (so that the kernel can later be recompiled).

Log in as root and perform the following:

1) Insert the RedHat CD into your CDROM drive.

2) Mount it (if it does not automatically mount itself), by typing:

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
```

(This assumes that /dev/cdrom is a symbolic link to your CDROM device. If this is not the case, substitute your CDROM device specification for /dev/cdrom in the mount command).

3) Move to the /RedHat/RPMS directory:

```
cd /mnt/cdrom/RedHat/RPMS
```

4) Install the kernel source RPM:

```
rpm -ivh kernel-source-2.2.*.i386.rpm
```

5) Unmount the CD:

```
cd  
umount /mnt/cdrom
```

Change the SHMMAX parameter value by doing:

1) Edit the source file which defines the SHMMAX parameter value:

```
vi /usr/src/linux/include/asm/shmparam.h
```

2) Locate the line in the file that begins with the following:

```
#define SHMMAX 0x2000000
```

Note that 0x2000000 is 32 megs, expressed as a hexadecimal value. Oracle recommends that this value be set to one half of the installed memory.

Change the 0x2000000 value to your new value. A typical value might be one of the following:

For 64 meg, set this to 0x4000000

For 96 meg, set this to 0x6000000

For 128 meg, set this to 0x8000000

For 192 meg, set this to 0xC000000

For 256 meg, set this to 0x10000000

3) Save the file and exit from vi:

```
:wq
```

4) Recompile the kernel. Note that recompiling the kernel will require you to set more than 100 different options, and that choosing the wrong value for an option may leave you with an unbootable system! You should obtain a sufficient amount of support and experience before you undertake this complex operation. This is why it is recommended that you not change the SHMMAX parameter setting, unless you feel that it is absolutely necessary.

OPERATING SYSTEM PATCHES:

Use the latest kernel patches for Linux.

WINDOW MANAGER:

Use any supported window manager. Character mode installations are not supported.

SYSTEM LIBRARIES:

GNU C Tools egcs-1.1.2 (glibc 2.1)

Type:

```
rpm -q egcs
```

to see which version of egcs is installed.

TASKS WHICH NEED TO BE PERFORMED PRIOR TO INSTALLING ORACLE:

1) Log in as root and open a terminal window.

2) Ensure that the Oracle Installer can find the gmake command that it needs. Check this by typing:

```
ls -l /usr/bin/gmake
```

If gmake does not exist, create the symbolic link to make:

```
ln -s /usr/bin/make /usr/bin/gmake
```

3) Ensure that your server has an entry in the /etc/hosts file. (This allows Oracle to determine the IP address of the server).

4) If a previous installation of Oracle 8i failed, you may have to clean up some files that were left over from the failed installation. Failure to do so may cause this new installation to fail. To clean up these left over files, type:

```
rm -f /etc/oraInst.loc
mv /etc/oratab /etc/oratab.old
rm -Rf /tmp/orainstall
rm -f /usr/bin/dbhome
rm -f /usr/bin/oraenv
rm -f /usr/bin/coraenv
rm -f /usr/local/bin/dbhome
rm -f /usr/local/bin/oraenv
rm -f /usr/local/bin/coraenv
```

(The above files/directories may or may not exist. Ignore any errors relating to non-existent files or directories).

If ORACLE_BASE is already defined, type the following command to delete any left over files in that directory tree:

```
rm -Rf $ORACLE_BASE/*
```

5) Define a new directory tree for this installation of Oracle. If there is already an existing version of Oracle installed on the system, you must install this new version into a new ORACLE_HOME.

6) Create the Linux group to own the software. Oracle recommends that you create a group called oinstall to own the software, and a dba group to administer the software on a day-to-day basis:

```
groupadd oinstall
groupadd dba
```

7) Create the Oracle owner account, having a default group of oinstall and also be a member of dba. The oracle account is used to install the Oracle software and startup or shutdown the Oracle database.

```
useradd oracle -g oinstall -G dba oracle
passwd oracle
```

(The passwd command will prompt you twice to enter the desired password for the oracle account).

8) Create the mount points for the database/code tree. Example for an OFA-compliant directory structure:

```
mkdir -p /u01/app/oracle/product/8.1.6
mkdir -p /u02/oradata /u03/oradata /u04/oradata
chown -R oracle:oinstall /u01/app/*
chown -R oracle:oinstall /u02/* /u03/* /u04/*
```

9) Place the Oracle 8.1.6 Installation CD into the CDROM drive. Mount the Oracle 8i CDROM for subsequent usage:

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
```

(This assumes that /dev/cdrom is a symbolic link to your CDROM device. If this is not the case, substitute your CDROM device specification for /dev/cdrom in the mount command).

10) If using Net8, reserve a port for Net8 by editing the /etc/services file, and adding the following line:

```
listener 1521/tcp # Oracle Net8 listener
```

11) Log out of the root account and log back in as oracle.

12) Open a terminal window.

13) These instructions assume that your shell is: /bin/bash. Your shell can be viewed by typing:

```
echo $SHELL
```

If this is not your shell, substitute the appropriate file name for all following references to .bash_profile

14) Edit the .bash_profile file in your home directory, and insert the following lines into this file (Note that the lines beginning with a # are comments, and can be omitted):

```
# The definition for ORACLE_BASE must match the beginning of the first mount point
# defined in the above step 8:
ORACLE_BASE=/u01/app/oracle; export ORACLE_BASE
# The definition for ORACLE_HOME must match the first
# mount point defined in the above step 8:
ORACLE_HOME=$ORACLE_BASE/product/8.1.6; export ORACLE_HOME
# If you get an error saying that you cannot get to Oracle from a user account which is
# different from the oracle one, you do not have LD_LIBRARY_PATH set correctly. As
# an alternative to setting LD_LIBRARY_PATH, you can add the fully-qualified path
# to $ORACLE_HOME/lib to the /etc/ld.so.conf file, and then run ldconfig (as root).
LD_LIBRARY_PATH=$ORACLE_HOME/lib; export LD_LIBRARY_PATH
ORACLE_SID=ORCL; export ORACLE_SID
# NLS_LANG specifies the storage character set to be used when installing or creating a
# database. If you are using a language other than US English, change the "american" in the
# following NLS_LANG definition to your desired language. National Language Support is
# described in detail in the "Oracle 8i Server Concepts" and "Oracle 8i Server Reference"
# manuals. Some possible values for the NLS_LANG definition can be one of the following:
# Language          NLS_LANG value
# -----
# US English        american
# Arabic            arabic
# Brazilian Portugese  "brazilian portugese"
# Canadian French   frc
# Czech             czech
# Danish            danish
# Dutch             dutch
# Finnish           finnish
# French            french
# German            german
# Hungarian         hungarian
# Icelandic         is
# Italian           italian
# Japanese          japanese
# Korean            korean
# Lithuanian        lt
# Mexican Spanish   esm
# Norweigan         norweigan
# Polish            polish
# Portugese         portugese
```

```

# Russian            russian
# Simplified Chinese  "simplified chinese"
# Slovak             slovak
# Swedish            swedish
# Thai              th
# Traditional Chinese "traditional chinese"
# Turkish            turkish
# (Include the quotes in the NLS_LANG value, if they are included in the above listing)
NLS_LANG=american; export NLS_LANG
ORACLE_TERM=vt100; export ORACLE_TERM
PATH=$ORACLE_HOME/bin:$PATH; export PATH
CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib
# For each <product>, set the CLASSPATH environment
# variable to include any product-specific jlib directories:
# CLASSPATH=$CLASSPATH:$ORACLE_HOME/<product>/jlib
# If using OCI and thin JDK 11x JDBC drivers, append the
# following path to the previous CLASSPATH definition:
# $ORACLE_HOME/jdbc/lib/classes11.zip
# If using OCI and thin JDK 102 JDBC drivers, append the
# following path to the previous CLASSPATH definition:
# $ORACLE_HOME/jdbc/lib/classes102.zip
# If installing JDBC, uncomment the following line:
# LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/jdbc/lib
# If using interMedia Text, uncomment the next line:
# LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/ctx/lib
export CLASSPATH
#
# The definition for umask is needed only if umask does not default to 022. The current
# umask value can be displayed by typing the command: umask
umask 022
# On the server where the Oracle database will be installed, set the DISPLAY environment
# variable to the machine name (or IP address) of your own workstation (NOT the system
# where Oracle is being installed on!), the X server, and the screen being used. If you are
# not sure of what the X server and the screen should be set to, use 0 for both. If you get an
# Xlib error similar to "Failed to connect to server" or
# "Connection refused by server" when starting the OUI, enter the following command in your
# workstation's session: xhost +<server_machine_name>
# The following example assumes that you're installing the Oracle server on your workstation;
# Substitute the correct machine name, IP address, X server, and screen for the following
# `uname -n` if this is not so:
DISPLAY=`uname -n`:0.0; export DISPLAY

```

(In order for another local user (other than oracle) to access the database, they will need the same definitions in their .bash_profile file, as were created for the oracle software owner. An alternative is to place shell scripts in /etc/profile.d, as these scripts are automatically executed system-wide by all users).

15) ORA_NLS33 specifies the directory under which Oracle will store its *.nlb files (which define languages, territories, character sets, and linguistic sorting orders), and is necessary only if ORACLE_HOME has multiple versions of directories under which *.nlb files will be placed. The default is \$ORACLE_HOME/ocommon/nls/admin/data.

Specify ORA_NLS33 by including the following line in the .bash_profile file:

```
ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data; export ORA_NLS33
```

16) Log out, and then log back in as oracle, in order to reset your environment to the values you just specified in the .bash_profile file that you just edited.

17) Verify the correct setting of your PATH, by typing:

```
echo $PATH
```

Ensure that the following paths are in your PATH definition, in addition to the \$ORACLE_HOME/bin that you just added to the PATH:

```
/bin
```

```
/usr/bin
```

```
/usr/local/bin
```

(If any of these are missing, edit the .bash_profile script and append them to the PATH definition. After that is done, go back and restart with step 16.

18) Ensure X windows is configured correctly on the local system, by typing:

```
xclock &
```

If you get an Xlib error similar to "Failed to connect to server" or "Connection refused by server", enter the following command in your workstation's session (NOT from a window that is logged onto your database server):

```
xhost +server_machine_name
```

INSTALL ORACLE:

Oracle must be installed from a GUI console. This is accomplished by using the normal GNOME or KDE desktop.

1) Start the Oracle installer, by typing:

```
cd /mnt/cdrom  
./runInstaller
```

If the installer does not start, please review all previous steps, as the most likely cause is that a previous failed install has not been cleaned up or the environment variables are missing or incorrect.

a) A GUI screen appears which welcomes you to the Java-based Oracle installer. Click the Next button.

b) Verify that the Source path says:

```
/mnt/cdrom/stage/products.jar
```

The Destination path should be the same as your \$ORACLE_HOME environment variable. Click Next.

c) At the "Available Products" screen, select "Oracle 8i Enterprise Edition 8.1.6.1.0". Click Next.

d) At the "Installation Types" screen, select "Custom". Click Next.

e) At the "Available Product Components" screen, select the desired components to be installed. Click Next.

f) At the "Component Locations" screen, Click Next to use the current volume as the location for the displayed software components.

g) At the "Privileged Operating System Groups" screen, Click Next to select the oinstall group to be the OSDBA and OSOPER group.

h) You will then be asked whether you wish to create a database as part of the installation. If you choose NO, the Database Configuration Assistant (dbassist) will be present in the window that appears at the end of the installation, however, due to Oracle Bug 1098669, you cannot run it from this window. The workaround is to run the dbassist command from \$ORACLE_HOME/bin at the end of the install. Click Next.

i) The Summary screen will be displayed. This lists everything you're about to install. Click the Install button.

j) The Oracle software installation will start.

k) After the files have been copied, you will be prompted to run the \$ORACLE_HOME/root.sh script as root:

Open up a new terminal session.

Log in as root.

Type the following command to move to your \$ORACLE_HOME directory (This must match the definition contained in oracle's .bash_profile script for ORACLE_HOME):

```
cd /u01/app/oracle/product/8.1.6
```

Type the following command:

```
./root.sh
```

Press <ENTER> when prompted for your local bin directory.

Exit the terminal window by typing: `exit`

l) Click the OK button in the window that prompted you to run the root.sh script.

m) The Net8 Configuration Assistant (netca) will start. Check the "Perform typical configuration" box. Click Next.

n) The Database Configuration Assistant (dbassist) will start, if you have chosen to run it as part of the installation. See the below "Create the Database" section for details. Click OK after the database has been created. Confirm that you want to exit at the next screen. Click the Next button. When you exit the database assistant, you will go back into the Oracle Universal Installer to complete the installation.

o) The "End of Installation" screen will be displayed. Click Exit to exit the installer. Click Ok to confirm that you really want to exit the installer.

2) Check the log files in /tmp for any errors.

3) Read the appropriate release notes in \$ORACLE_HOME/relnotes.

CREATE THE DATABASE:

Note that creating the database can take a long time (i.e. hours) on a slow system (mostly due to the fact the Oracle JServer is loaded into the database).

- 1) Start the Database Configuration Assistant, by typing:
`dbassist &`
- 2) Select the "Create a Database" option and click Next.
- 3) When prompted for "Select the type of database to create", select "Custom".
- 4) You can accept the defaults on the next four screens, however, if you have installed JServer, you must increase the 10M RBS tablespaces to 50M, to prevent the JServer from running out of extents. Click Next on each screen. Stop at the screen that asks you for the Global Database Name.
- 5) When prompted to enter a Global Database Name, enter a name that uniquely identifies this database. The name must be eight or less characters. Check that the displayed Oracle SID is the same as you entered earlier in the ORACLE_SID environment variable definition in your .bash_profile file. Click Next.
- 6) Accept the defaults on the next six screens. Click Next until you get to the screen that asks you to "Create database now".
- 7) When you reach the screen "Review the following database information...", fill out the details as requested. Click Next.
- 8) Ensure that "Create database now" is selected, then click the Finish button.
- 9) Click "Yes" to confirm that you want to proceed.
- 10) When the database creation is finished, you will get a message saying so. Click "OK".

(If you ever want to manually create a database, you can use the scripts in \$ORACLE_HOME/rdbms/install/create to create the database. Refer to the README file in that directory to determine which files must be executed and in what order).

TASKS WHICH NEED TO BE PERFORMED AFTER INSTALLING ORACLE:

Oracle creates two accounts for administering the database:

```
sys, having a password of change_on_install
system, having a password of manager
```

As these passwords are widely known, you should change them as soon as possible, by typing the following commands:

```
sqlplus system/manager
alter user sys identified by 'your-new-password';
alter user system identified by 'your-new-password';
exit
```

Start the Oracle Net8 Listener, if you want to access the Oracle server via Net8. (The Oracle installation creates typical Net8 configuration files, which will work for most TCP/IP installations, but you may need to change them depending on your network configuration). Type:

```
lsnrctl start
```

Note that Oracle recommends NOT using prespawnd dedicated servers in Oracle 8.1.6.

Check to see if the Oracle Server is running:

```
ps -ef | grep pmon
```

If no process is displayed, you can manually start the database by typing the following commands:

```
sqlplus /nolog
connect internal
startup
exit
```

Verify that the Oracle server has been successfully started and that the database is accessible. Type:

```
sqlplus system/manager
select count(*) from dba_tables;
exit
```

The "select count" statement should display a count in excess of 100 tables. If so, the database has been successfully created.

Fix the dbstart script:

Due to an Oracle Bug 1154931, the supplied \$ORACLE_HOME/bin/dbstart script does not start any 8.1.6 instances if they are down. dbstart will report "Database <SID> warm started" rather than starting the instance. This is due to a change in the banner that Oracle displays in svrmgrl or SQL*Plus.

For the Oracle Enterprise Edition, a workaround of changing the dbstart script to look for the string "Oracle8i Enterprise Edition" instead of "PL/SQL" can be used to fix this problem:

1) Save a copy of the 8.1.6 \$ORACLE_HOME/bin/dbstart script.

2) Edit the script and change the 'awk' lines from:

```
/PL\SQL (Release|Version)/ {substr($3,1,3) ;
print substr($3,1,3)}'\`
```

to:

```
/PL\SQL: (Release|Version)/ {
print substr($3,1,3); exit }
/Oracle7 Server (Release|Version)/ {
print substr($4,1,3); exit }
/Oracle8i Enterprise Edition (Release|Version)/ {
print substr($5,1,3); exit }'\`
```

3) Fix the definition for LD_LIBRARY_PATH from:

```
LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${ORACLE_HOME}/lib
```

to:

```
LD_LIBRARY_PATH=${ORACLE_HOME}/lib:${LD_LIBRARY_PATH}
```

Note: This workaround will only work for Oracle8 Enterprise Editions. Similar workarounds may be used for other editions.

Configure the server to automatically start the Oracle database (and Net8) at system startup and shutdown the Oracle database at system shutdown:

Automating database startup is optional, but automatic shutdown is recommended because it guards against improper shutdown of the database.

1) Configure Oracle to automatically start the Oracle database whenever the dbstart script is executed, and to automatically shutdown the Oracle database whenever the dbshut script is executed. Type:

```
vi /etc/oratab
```

On the last line, change the third parameter from N to Y to auto-start/stop the database.

2) Log in as root.

3) Create a file containing commands to startup or shutdown the Oracle database. Type:

```
vi /etc/rc.d/init.d/dbora
```

The file should look similar to the following:

```
#!/bin/bash
#
# /etc/rc.d/init.d/dbora
#
# Oracle database startup/shutdown script for Linux
#
# Set ORA_HOME to be equivalent to the ORACLE_HOME from
# which you wish to execute dbstart and dbshut
#
ORA_HOME=/u01/app/oracle/product/8.1.6
PATH=$PATH:$ORA_HOME/bin; export PATH
#
# Set ORA_OWN to the user id of the owner of
# the Oracle database in ORA_HOME
#
ORA_OWN=oracle
#
# Set LD_LIBRARY_PATH so Oracle can find the
# dynamic link libraries it needs (mainly for lsnrctl)
LD_LIBRARY_PATH=$ORA_HOME/lib; export LD_LIBRARY_PATH
#
if [ ! -f $ORA_HOME/bin/dbstart -o ! -d $ORA_HOME ]
then
    echo "Oracle startup: cannot start"
    exit
fi
case "$1" in
start)
    ps -ef | grep -v grep | grep ora_.*_* > /dev/null
    if [ $? -ne 0 ]
    then
        #
        # Start the Oracle database
        #
        su $ORA_OWN -c $ORA_HOME/bin/dbstart
    fi
    #
    # Start Net8:
    #
    su $ORA_OWN -c "$ORA_HOME/bin/lsnrctl start"
    #
    ;;
stop)
    # Stop the Oracle databases
    ps -ef | grep -v grep | grep ora_.*_* > /dev/null
    if [ $? -eq 0 ]
    then
        su $ORA_OWN -c $ORA_HOME/bin/dbshut
    fi
    ;;
esac
```

4) Set required permissions on the oracle file:

```
chmod 744 /etc/rc.d/init.d/dbora
```

5) Link the file to the startup/shutdown directories, so that the operating system will run the above script when the system is started or shutdown. Type:

```
ln -s /etc/rc.d/init.d/dbora /etc/rc.d/rc5.d/S99dbora
ln -s /etc/rc.d/init.d/dbora /etc/rc.d/rc0.d/K01dbora
```

6) Log out from root and log back in as oracle.

INIT.ORA PARAMETER SETTINGS:

There is a template init\$ORACLE_SID.ora file contained in \$ORACLE_BASE/admin/\$ORACLE_SID/pfile/init\$ORACLE_SID.ora. This file contains settings for small, medium, and large databases, with the settings for medium and large databases commented out. Edit this file to make whatever parameter changes are required for your database. Any changes you make will only take effect when the database is shutdown and then restarted.

*PRO*C SETTINGS:*

If using Pro*C/C++ precompilers, you must customize the \$ORACLE_HOME/precomp/admin/ottcfg.cfg and \$ORACLE_HOME/precomp/admin/pcscfg.cfg files for your environment, as described in the "Pro*C/C++ Precompiler Programmer's Guide" manual and your C compiler documentation. The following lines are a typical pcscfg.cfg file for Pro*C. (Replace <your-oracle-home> with your ORACLE_HOME definition and enter the 2 lines of the sys_include definition on one long line):

```
sys_include=(<your-oracle-home>/precomp/public,
             /usr/lib/gcc-lib/i386-redhat-linux/egcs-2.91.66/include/, /usr/include)
include=(<your-oracle-home>/precomp/public)
include=(<your-oracle-home>/rdbms/demo)
include=(<your-oracle-home>/rdbms/public)
include=(<your-oracle-home>/network/public)
include=(<your-oracle-home>/plssql/public)
include=(<your-oracle-home>/otrace/public)
ltype=short
```

RMAN:

If using RMAN, ensure that the correct rman executable is being run. Type:

```
which rman
```

If the output shows /usr/bin/rman, this indicates that \$ORACLE_HOME/bin is not found before /usr/bin in the PATH environment variable. (Under a default RedHat 6.x install, a KDE program called rman (a file viewing / editing program) is installed under /usr/bin). You can also execute Oracle's RMAN by typing: \$ORACLE_HOME/bin/rman

ORACLE 8.1.6 ON LINUX RESTRICTIONS:

Linux does not have support for very large files (files larger than 2gig).

Linux does not support raw devices, unless your operating system documentation indicates that it does support raw devices.

Third-party filters to extract pertinent ASCII information from documents being indexed are not provided by Oracle InterMedia Text for Linux.

The Parallel Server option is not supported.

The Visual Information Retrieval Cartridge option is not supported.

Only the secure socket layer (SSL) authentication is supported for the Advanced Security option.

The OS cartridge for the Data Gatherer is not available.

The Precompiler options (Pro*COBOL, Pro*Ada, Pro*FORTRAN) are not supported.

Legato Storage Manager is not supported.

Modified system header files for Pro*C/C++ are located in \$ORACLE_HOME/precomp/public with the Oracle 8.1.6 for Linux distribution.

You can not upgrade 8.0.5.x databases to 8.1.6.x using the Oracle Data Migration Assistant (ODMA); Use exp/imp utilities instead. (Upgrading 8.1.5.x to 8.1.6.x databases is supported).

Thin and OCI drivers for JD 1.2 are not supported. (Drivers for JDK 1.1 are supported).

KNOWN PROBLEMS WITH ORACLE 8.1.6.1 ON LINUX:

SQL*Loader is unable to load LOB data.

Jobs submitted to the Intelligent Agent by OEM may not complete.

BUG 1210242 - Do not use TIMED_STATISTICS *AND* SQL_TRACE against 8.1.5 or 8.1.6.1. If you do, cursor's will not be shared, and the VERSION_COUNT's in V\$SQLAREA will be very high. This causes lots of CPU time to be wasted traversing the tables of child library cache objects, and probably library cache latch contention as well. Fixed in 8.1.6.2 and 8.1.7.

Bug 995765 - RMAN fails when attempting to allocate a channel for backup and recovery. Patch for 8.1.6.1 available from <ftp://oracle-ftp/server/patchsets/unix/Linux/bug995765>.

Fixed in 8.1.6.2 and 8.1.7.

RED HAT ENTERPRISE EDITION FOR ORACLE 8i:

Supports raw I/O.

Supports large process memory usage (3gig, instead of 1gig).

Supports 64-bit file I/O. Allows files greater than 2gig.

Supports vectored I/O. Performs reads/writes from an offset into a large file.

SPECIAL NOTE FOR RED HAT 7.0 (AND PROBABLY ANY GLIBC-2.1.9X BASED DISTRIBUTION):

There is a generic incompatibility across different major versions of GNU libc that does not guarantee that a binary linked against a version of glibc will keep working if relinked in an environment of a later major glibc release.

Since Oracle 8.1.6.1 is linked against glibc-2.1.3 and Red Hat 7.0 ships with glibc-2.1.9x, Oracle 8.1.6.1 fails to start on RedHat Linux 7.0 after relinking.

Possible Symptoms are:

Starting up the database fails with ORA-3113 after some time spent looping in `semop()` and a core file is generated. The stack trace taken with GDB is the following:

```
#0 0x404c74e1 in __kill ()
#1 0x8469156 in slcra ()
#2 0x846cd01 in ssexhd ()
#3 0x4048ed60 in pthread_sighandler
#4 <signal handler called>
#5 0x846cd12 in ssexhd ()
#6 0x4048ed60 in pthread_sighandler
#7 <signal handler called>
#8 0x846cd12 in ssexhd ()
#9 0x4048ed60 in pthread_sighandler
#10 <signal handler called>
#11 0x846cd12 in ssexhd ()
#12 0x4048ed60 in pthread_sighandler
#13 <signal handler called>
#14 0x8a6f27c in sskgmstat ()
```

After installing Oracle, when `dbassist` runs, it hangs at "Initializing Database 80%" or it will stop near 0% if the database does not start. Examining the processes (via `top`) shows the `svrmgrl` process taking up 99.9% of the CPU and 2.5MB of memory.

One workaround is:

1) Install the following RedHat 7.0 packages:

```
compat-egcs-6.2-1.1.2.9
compat-glibc-6.2-2.1.3.2
```

2) Modify all occurrences of `CC` and `LINK` definitions in all `env_*.mk` makefiles in the `$ORACLE_HOME` software tree from either `$(LDCCOM)` or `gcc` to `i386-glibc21-linux-gcc`. Example:

Change the lines such as:

```
CC=cc
LINK=$(LDCCOM)
LINK=$(PURECMDS) gcc $(LD_FLAGS) $(COMPOBJS)
```

to:

```
CC=i386-glibc21-linux-gcc
LINK=i386-glibc21-linux-gcc
LINK=$(PURECMDS) i386-glibc21-linux-gcc $(LD_FLAGS) $(COMPOBJS)
```

3) You will also need to modify the `genclntsh` shell script and substitute:

```
LD="ld -shared -L${ORACLE_HOME}/lib"
LD_RUNTIME="-R${ORACLE_HOME}/lib"
LD_OPT="-h ${CLNT_LIB}"
```

with:

```
LD="i386-glibc21-linux-gcc -shared -L${ORACLE_HOME}/lib"  
LD_RUNTIME="-Wl, -R${ORACLE_HOME}/lib"  
LD_OPT="-Wl, -h${CLNT_LIB}"
```

4) Rebuild libclntsh.so by running 'genclntsh' and relink oracle, svrmgrl, etc using the usual relink commands. You'll get warnings in the relink process, but they can be ignored.

If trying to install IAS 8i, the relink script provided with IAS to relink the executables after editing the make files won't work. You have to go into each individual product's lib directory and issue a "make -f ins_<prod>.mk install" to relink.

This has been tested on env_rdbms.mk, env_sqlplus.mk, and env_network.mk, but should be working on the other makefiles as well.

Another workaround is:

1) Downgrade to RedHat 6.2.

Another workaround is:

1) Get a copy of the oracle kernel, and utilities such as svrmgrl, imp, exp, etc from a RedHat 6.2 box and place them in \$ORACLE_HOME/bin.

Note that this workaround will not allow you to relink the Oracle kernel and/or utilities; The resulting binaries will not run under a glibc-2.1.9x system.

Another workaround is:

1) If you haven't relinked multiple times, it's enough to copy back the old binary versions in \$ORACLE_HOME/bin (such as oracleO, impO, etc) to oracle, imp, etc.

Note that this workaround will not allow you to relink the Oracle kernel and/or utilities; The resulting binaries will not run under a glibc-2.1.9x system.

Another workaround is:

1) Install the following RedHat 7.0 packages (if not already installed):

```
compat-egcs-6.2-1.1.2.9  
compat-glibc-6.2-2.1.3.2
```

2) Before installing Oracle 8.1.6.1, install the retro compiler as the default cc and gcc compiler, by typing the following commands, as root:

```
cd /usr/bin  
mv gcc gcc.linux7  
ln -s i386-glibc21-linux-gcc gcc
```

3) After installing Oracle 8.1.6.1, as oracle, type:

```
cd $ORACLE_HOME/bin  
./genclntsh  
./relink all
```

(If you want to restore the Red Hat 7.0 version of the compiler, remove the gcc link and cp the gcc.linux7 file back to gcc).

Another reported (but un-verified) workaround is:

- 1) Get the following RedHat 6.x packages (if not already installed):

```
glibc-2.1.3-*.rpm
```

- 2) Extract the following files from the RPM package and copy them to \$ORACLE_HOME/lib:

```
libc-2.1.3.so
```

```
libpthread.so
```

```
libdl.so
```

```
ld-linux.so.2
```

(Those were symlinks to some like libXX-2.1.3.so - Copy the targets)

- 3) Create \$ORACLE_HOME/lib/libc.so (assuming your ORACLE_HOME is /usr/oracle):

```
GROUP ( /usr/oracle/lib/libc-2.1.3.so /usr/oracle/lib/ld-linux.so.2 /usr/lib/libc_nonshared.a )
```

- 4) Relink all oracle executables. Type:

```
cd $ORACLE_HOME/bin
```

```
./genclntsh
```

```
./relink all
```

(The person that did this did not install any java components. All references to the java option were removed from oracle libraries.)

Another reported (but un-verified) workaround is:

- 1) Install Oracle 8.1.6 on RedHat 6.2.

- 2) Create a tar/gzip image of the entire Oracle directory.

- 3) Copy the tar/gzip image to the RedHat 7.0 system and unzip it into the same directory paths as existed on the RedHat 6.2 system.

- 4) Copy the /etc/ora* file(s) from the RedHat 6.2 system to the RedHat 7.0 system.

Another reported (but un-verified) workaround is:

- 1) Install Oracle 8.1.6 on RedHat 7.0. Do not create the database.

- 2) Run the following script to change all references to cc and gcc to /usr/bin/i386-glibc21-gcc:

```
#!/bin/sh
cd $ORACLE_HOME
LIST=`rgrep -rl -x mk "gcc" .`
for FILE in $LIST ; do
    # Make a backup
    if [ ! -f $NAME.orig ]; then
```

```

        cp $NAME $NAME.orig
    fi
    # Change cc and gcc for /usr/bin/i386-glibc21-gcc
    sed -e "s/gcc/\usr/bin/i386-glibc21-linux-gcc/
        s/CC=cc/CC=\usr/bin/i386-glibc21-linux-gcc/" \
        $NAME > $NAME.new
    mv $NAME.new $NAME
done

```

3) Change the genclntsh script so that LD uses a different link script, searching for /usr/i386-glibc21-linux/lib first.

4) Relink all oracle executables. Type:

```

cd $ORACLE_HOME/bin
./genclntsh
./relink all

```

5) Create the Oracle database.

(This idea is based on the Oracle 8.0.5 patch script).

SOME IMPORTANT ORACLE BUGS AFFECTING 8.1.6:

Histograms on partition tables with any empty partitions causes the optimizer to crash on any query for the affected table.

Bug 962964.

Workaround: Drop statistics and rebuild without using histograms.

Fixed in 8.1.6.2 (or 8.1.7).

Rebuild of partitioned index in parallel mode crashes with ORA-12801 + ORA-27062

More than 30 bugs in PQO, which is probably causing this error.

Workaround: Remove the PARALLEL clause on the index builds.

Fixed in 8.1.6.3?

Export failing with ORA-1000 (too many open cursors)

Occurs on full exports of databases that use locally managed tablespaces.

Bug 1244182.

Workaround: Increase the OPEN_CURSORS parameter.

Fixed in 8.1.6.3 (but may need an additional patch?)

LINUX AND WINDOWS NT DESKTOP AND TOOLS ENVIRONMENT COMPARISON:

As Linux is very stable, requires only a fraction of the hardware resources that Windows NT requires, and performs exceedingly well, you might want to consider implementing Linux as your workstation desktop environment. This section is intended to make you aware of the many desktop applications available for Linux.

Linux provides a wealth of tools and multiple GUI environments for use.

The two most common GUI desktop environments are GNOME and KDE. Each provides a highly customizable interface, similar to what you are familiar with on a Windows NT system.

In order to configure a Linux system, the linuxconf program functions in a similar manner to NT's Control Panel applets.

Additionally, there are command-line utilities for all of the system configuration options, allowing you to include any of them into your own standard or custom environment-setting scripts.

The following is a list of some of the currently-available common software packages for Linux:

Backup & Recovery utilities:	tar, cpio, afio, tob, taper, cdbl, dump, restore
CAD/CAM graphics drawing:	xfig
CD encoder:	bladeenc
CD players:	gtcd, cdp, xplaycd
CD ripper:	cdparanoia, cdda2wav, ripperX
CD-RW recorder:	cdtux, cdrecord, xcdroast
Databases:	postgres, mysql, oracle
DOS emulators:	dosemu
Drawing package:	dia
DVD players:	oms
Email:	mail, mutt, Netscape mail, mh
Encryption:	openssh, ssh, ssh2, mcrypt
Fax:	FaxMail
File Manager:	gmc, mc
File sharing with remote systems:	nfs, samba
Firewalls:	portsentry, ipchains
Graphics Image editor:	gimp
Graphics viewers:	xanim, aktion, gphoto, xnview, nview, xzgv, xv, ee, gqview, display, uudeview, compupic
Help text display:	locate, man, apropos
Instant Messaging:	gnomeicu, licq
Internet Browser:	Netscape Communicator, lynx, wget
Menu Editor:	gmenu
Microsoft Exchange client:	tradeclient
MP3 players:	freeamp, xmms, mpg123, amp
MPEG players:	mtv, mtvp, mpeg_play, gtv, plaympeg
Multimedia players:	xanim, aktion, mtv, mtvp
News Readers:	pan, netscape
Office Suite:	soffice
Painting programs:	xpaint, kpaint
Palm Pilot utilities:	gpilot, pilot-xfer, jpilot, pilot_applet
PDF viewers:	acroread, xpdf
Personal Information Managers (PIM):	gnome-pin, ical, korganizer, jpilot
Photo editor:	photopaint
PPP dialer/configuration:	rp3, rp3-config
Presentation Graphics:	soffice

Security monitors/auditing:	tiger, logcheck, tripwire
Spam filters:	junkbuster
Spelling checkers:	look, ispell, spell
Spreadsheet:	soffice, gnumeric
System configuration utilities:	linuxconf, gnomecc, sysctl
System performance utilities:	gtop, top, sar, vmstat, iostat
Terminal emulators:	gnome-terminal, rxvt, konsole
Text editors:	ed, emacs, gvim, vi, vim
Text to Audio converters:	catspeech
User account creation and modification:	useradd, userdel, usermod
Virus scan:	antivir, amavis, avguard, tkave
Windows emulators:	vmware, wine
Word Processing:	soffice, xwp, abiword

Note that for many of software application categories, there are different software packages available, allowing you to choose the software that best fits your specific needs.

There are thousands of free software packages, which are easily available, for use in solving your specific application requirements.

If you want to write your own custom applications, there are many programming languages and tools available for use. Some of them are: C/C++, Fortran, awk, python, perl, grep, tcl, tk, gtk

The Linux shells provide a robust and flexible environment for designing, customizing, and automating your scripts.

May 3, 2001

Paper 183