Oracle for Linux Installation HOWTO

Table of Contents

Table of Contents

Oracle for Linux Installation HOWTO

Stephen Darlington, `<stephen@zx81.org.uk>`

$Revision: 1.14 $, $Date: 2000/06/19 21:07:43 $

With this HOWTO, and a little luck, you will be able to get "Oracle 8i Enterprise Edition for Linux" installed, create a database and connect to it from a remote machine. The main focus of this guide is RedHat Linux 6.0, although it should work more−or−less unchanged for any other recent distribution.

1.Introduction

2.Starting off

3.The installer

4.Creating a database

5.Configuration

6.Final Words

1.Introduction

1.1 What's in here?

Linux is well known for being difficult and, generally, user hostile. Being a bit of a Unix fan I'm not sure whether I agree with that or not.

Oracle is similar I guess. Initially it's difficult to get to grips with, but it's difficult to work with any other RDBMS when you're used to it.

Combine the two, remember that 8i is only the second production release, and you realise that this isn't going to be straight−forward, even if you're familiar with both.

I am, but I had problems. Many problems were my own stupidity or hubris, but I document them for completeness.

1.2 Who is this HOWTO for?

First, this document is for people who want to install Oracle 8i version 8.1.5 on Linux. It does not cover any earlier versions, although it should work with similar later versions such as 8.1.6.

If you want to install 8.0, I recommend you try Linux Journals guide, and if you want to install any of the previous versions you're going to have to use the SCO version and follow Paul Haigh's Oracle Database HOWTO.

If you're trying to install the 'right' version, what level of background knowledge will you need?

Perhaps the easiest way is if I explain a little of my background, clearly if yours is similar we're going to be on the same wave−length. I've used a lot of Unix and Oracle over the last few years. At home I've been running Linux since 1994 and I've been using Solaris and HP−UX on−and−off since 1992. I first came across Oracle in 1996 and have worked with both versions 7 and 8. I'm mainly a developer, but I have done DBA and sysadmin−type work.

In summary, I can find my way around a Unix box and I know much of the Oracle terminology. You'll need both to brave the rest of this document. But don't worry if you have a different background, follow this guide closely and keep asking questions. The Linux community are a helpful bunch, just don't expect an answer if you haven't at least made an effort to solve the problem yourself.

1.3 New versions of this document

Things move quickly in the world of Linux and Oracle, meaning that this document can quickly get out of date. If this document is more than a month or two old, I suggest you take a look at my web site for an update.

1.4 Disclaimer

You get what you pay for. I offer no warranty of any kind, implied or otherwise. I'll help you where I can but legally you're on your own.

1.5 Credits and Thanks

This HOWTO has been written by Stephen Darlington. It couldn't have been created without the constant stream of questions and answers on Oracle Technet and the Usenet news−groups. So thanks to the people that keep posting and sorry that I can't credit you all individually!

Thanks to the following people, in no particular order, for their contributions to this document: Ton Haver, Guy Cole, Iain Frerichs, Albert Braun, Steve Morando, Krill Kokoshka, Brain Slesinsky, Galen G Burk and Bill Gathen.

I welcome any constructive feedback on this HOWTO and any general Linux or Oracle issues. Email me at

stephen@zx81.org.uk.

1.6 Licence

2.Starting off

2.1 Overview

In this section, we'll set up Linux so that you're in a position to get Oracle 8i from the CD that they sent you into your hard−disk.

The Oracle installation process begins when you've built your PC, installed Linux, configured it and connected it to your network.

2.2 Prerequisites

Hardware

I think that the most important part of the prerequisites is not to underestimate them. Oracle is a very big and complex application and you won't get the best out of it if you skimp too much on the hardware.

My biggest mistake was to assume that Oracle were joking when they said that you need 128Mb of RAM. I've installed Oracle a couple of times on Sun servers with that much, why would I need more on a CISC machine?

Believe Oracle not my gut. My machine with 32Mb of Ram ground on for less than half an hour before I realised that it was hopeless.

I was trying to use the bare minimum of hardware, and that's generally a bad idea. If you can't afford the hardware you certainly won't be able to afford the licences!

Things to look for on a production server are many disks, possibly RAIDed, and fast CPU's. Database access is relatively easy to break down into smaller parallel phases so having a number of processors really does help.

On the other hand, any machine that can run Linux and that has enough memory should be in with a chance. My other machine, the one I used for the rest of this document, is fine as a development machine. It is a Celeron 466Mhz with 128Mb of memory, an 8Gb hard disk, an Intel graphics card and a DM9102 network card.

2.3 Linux setup

Choice of distribution

Oracle seem to have done most of their development on RedHat Linux 6.0. For a fuss−free installation, do the same. I've heard horror stories about trying to get it installed on other distributions.

However, anything *like* RedHat should also do the trick. A recent version of Mandrake or SuSE should be fine (SuSE, in fact, are fairly active in supporting Oracle), and newer versions of RH pose no problems either.

Distribution Setup

Now that you've decided on which RedHat−like distribution you're going to use, you'll need to work out which options to set and which of the vast number of packages need to be installed to make Oracle work.

Firstly you need two to three times the amount of memory you have for your swap space. (You'll need around 200Mb of memory, real or virtual, just to run the installer!) Note that contrary to popular opinion, Linux swap partitions can be larger than 128Mb.

The arrangements of your other partitions can also be important. Make sure that the Oracle software is on a different partition to your operating system, and make sure that the Oracle datafiles are on yet another partition. The idea here is to make sure that your datafiles do not get fragmented. (In a live environment, you're likely to have a number of disk with Oracle spread across them. There are a number of good books that you consult for more information on this.)

As for the software, I took the easy option and installed just about everything. You certainly need all the 'base' packages, X Windows (the installation routine is a Java GUI) and the development tools regardless of whether you intend doing any coding or not. Compared to the size of Oracle and your databases a Linux distribution is tiny, probably less than a gigabyte. It's worth installing it all for an easy life!

Kernel parameters

The documentation suggests that you make changes to the Linux kernel so you can get more shared memory. Since this is so difficult in Linux (unlike most commercial Unix's you have to recompile the kernel), the approach I took was to go ahead with the installation anyway. The default RedHat Linux settings worked, although you may have to change them for a larger development or production system.

Note that some people have had to recompile the kernel to get Oracle to work at all. I guess it must depend on the other software that you're running on the same machine.

Follow the instructions in the Oracle documentation (on the installation CD in HTML format) and the Linux Kernel HOWTO to build your new kernel.

Users and groups

Using LinuxConf (or whatever other method you feel comfortable with), you need to add a new group called "dba" and a new user called "oracle", which should belong to your newly created "dba" group.

You can make any other user a DBA by putting them in the DBA group. If you have several DBA's this is probably a good idea for auditing purposes.

Installing the right Java Virtual Machine

If you check the official documentation, you'll find that Oracle recommend the Blackdown Java Runtime Environment version 1.1.6v5. That's what they mean. Don't think 'newer versions will be less buggy' as the installer probably won't work. And don't think, 'I'll be developing software so I'll just get the JDK,' as that won't work either.

There is one caveat to using this version of the JRE: the Oracle installer seems to be hard–coded to expect the JRE executable to be at `/usr/local/jre/bin/jre`. While this is inconvenient, it does not mean that you have to install it there.

I performed the following steps to get a working copy of the JRE:

1. Download the Java Runtime Environment from the Blackdown website
2. Move to where you want to install the JRE:

   ```
   cd /usr/local
   ```
3. Uncompress the archive:

   ```
   bzip2 -d -c jre-1.1.6-v5-glibc-x86.tar.bz2 | tar xvf -
   ```
4. Create a symbolic link between where Oracle thinks it is and where it actually is:

   ```
   ln -s jre116_v5 jre
   ```

2.4 Starting off questions and answers

Do I really need 128Mb RAM?

I would recommend that you do use 128Mb of RAM or more. I think it would be difficult to get any serious work done with less.

However, if you disable the Java option and set all the shared memory settings to be relatively small, there's no reason why it shouldn't work. I've heard success stories with 64Mb. You're probably not going to get away with 32Mb, though.

There is a caveat. You may only need half of what Oracle recommends to run the thing, but to install it their number starts to make sense. I've heard reports of the installer using 150Mb of memory and I've seen it well over 120Mb myself. If you have 64Mb or less of memory, make sure you have lots of swap space and patience.

An alternative if you absolutely can't add more memory: install Oracle on another, bigger machine and copy across the `$ORACLE_HOME` directory. You'll need to make sure that you have all the same users and groups (preferably with the same numeric codes) and take special care with SUID executables like `$ORACLE_HOME/bin/oracle/`.

Does it work with RedHat 6.1 or above?

I'm still running 6.0 myself, so all I can say is that a number of people have claimed success with this configuration.

At the time of writing, Oracle 8i has been certified with RedHat 6.0 and "Certification for other distributions is currently in progress" (Oracle 8i Patch FAQ).

Does it work with Debian/SuSE/Mandrake/some other distribution?

Oracle specify the Linux kernel version 2.2 or above and GLIBC version 2.1 with any window manager. In theory, any distribution that meets these requirements should work.

In practice, Oracle may not support it and you may have more problems trying to complete the installation. Unless you have a very good reason to do otherwise I suggest you stick to RedHat 6.0 with all the patches you can get hold of.

Does it work with development kernels?

There's no obvious reason why it shouldn't work –– I used 2.3.19 for a while because it supported my network card and the stable kernel at the time didn't –– but unless there's a pressing need it's certainly safest to stay well clear. I switched back to the stable series as soon as the driver was included.

Where do I get Oracle from?

Firstly, if you're brave, have a very fast Internet connection or inexhaustible patience (and unmetered access) you can download it from Oracle Technet. Beware: it's nearly 200Mb, and 8.1.6 is even bigger.

A better option is to get the CD. Oracle sometimes offer to send you a free development CD when you join Technet. It's certainly worth spending some time looking round their web site for that. Alternatively, you can buy them from the Oracle Store for around $40. It includes lots of other software too and comes on 15 discs.

3.The installer

3.1 How?

Generally, following the documentation is a good idea. It's not that bad and you'll get much better support from Oracle if you have. (I ended up breaking things –– and knowing it would –– by following the documentation for Oracle Applications. It was the only way to get decent support.)

This document is going to give an overview, but you should still have their documentation available.

3.2 What do I tell the installation program?

As part of the installation Oracle will ask a number of questions. Generally they're not too difficult but let's see what I entered and why.

1. Many people make the mistake of following Oracle's documentation and, therefore, fail at the first hurdle. Don't execute `runInstaller` as it almost always fails. Instead move to `install/linux` on the CD and run `runIns.sh` while logged in as 'oracle'.
2. It should show a title screen. Click 'Next.'
3. It should ask you to enter the source directory of the installation files ('jar' file) and your Oracle installation directory. You should be able to leave the former alone. The Oracle home directory is where you want to install the software. According to the installation documentation is should be somewhere on `/u01`, but I ignored that and put it in `/home/oracle`. Oracles advice, in this respect, is usually worth following. Click 'Next' when you've entered the details.
4. Now it should ask you for the DBA group. This is the Unix group you created in the last section and is probably 'dba'. Enter the details and click 'Next.'
5. This time it wants you to log in as 'root' and run `/tmp/OraInstall/orainstRoot.sh`. Do as it says. (You may have to run `pdksh` or `bash` in the 'Bourne compatibility mode' to get it to complete successfully.) When you're done click 'Retry.'
6. You're now given the option of what to install. Your best bet here is 'Oracle Enterprise Edition,' as this includes just about everything (table 3.1 in the Oracle documentation tells you exactly what it installs). Make sure the right radio button is selected and click 'Next.'

7. It should now allow you to choose what you install with much finer granularity. Unless you're particularly constrained by disk space or know exactly what you need, I'd recommend leaving it exactly as it is and clicking 'Next.' The Universal Installer won't let you make any silly choices so don't worry too much if you unselect something. You can always add it back in later.
8. For any products that you've asked it to install, the installer will allow you to change where it puts them. Again, only if you have a good reason to should you change it. Click 'Next' when you're done.
9. It now goes away and installs all the pieces of software you asked it to. This will probably take quite a while and will use far more memory than is reasonable.
10. It should ask you if you want to create a database. Select 'no'. There are two reasons for this: it often doesn't work and, even when it does, it's very slow (it seems to fire up another JVM, leaving X, the Oracle back–end and *two* virtual machines in memory; not good with 128Mb of memory).
11. The installer should now ask you about the network protocols that you want Oracle to support. The boxes all came up blank for me. I don't know what's supposed to be in there, but I clicked 'Next' and found that everything worked.
12. All the hard stuff is complete now. All the products you want should be installed and are ready to go. Congratulations.

3.3 Installing the patch

Unfortunately, the CD that Oracle sent you was probably version 8.1.5.0.0. As with almost all first releases there are problems with that version (problems include empty files, so they're quite serious) and a patch, to version 8.1.5.0.2 is essential. You'll certainly need it to progress to the "Configuration" section of this HOWTO. The patch described here is a cumulative patch, i.e., it includes all the files required to move from version 8.1.5.0.0 to 8.1.5.0.2.

The file you need is on the Oracle web site and is relatively easy to install.

1. This is probably the first of many patches, so create a directory called "patches" somewhere convenient (mine is in `$ORACLE_HOME`).
2. Download the file into it.
3. Create somewhere to put the files:

   ```
   mkdir /tmp/orapatch
   cd /tmp/orapatch
   ```
4. Uncompress the file:

   ```
   tar zvxf $ORACLE_HOME/patches/linux815patches.tgz
   ```
5. Run the shell script that's now in the current directory:

   ```
   ./linux_815patches.sh
   ```

Note that it's important not to uncompress the file in the current directory. The patch installer checks that the correct number of files are present and fails if there are not the right number. Of course, if it finds the patch archive it finds too many files!

3.4 Setting up your environment

Add the following lines to your ".profile" (or whatever the equivalent is for your shell):

```
. oraenv
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
```

Quite why the Oracle installer doesn't do this I have no idea.

If you see "`[: integer expression expected before -lt`" the next time you log in, it's because 'oraenv' is expecting your ULIMIT to be an integer rather than the default 'unlimited.' I've seen no ill effects by ignoring the error, but you can fix it by setting the ULIMIT to something finite.

3.5 Installations questions and answers

The installation program exits with 'CreateOUIProcess()'

Firstly, make sure that you're running the right version of the JVM. I don't know what Oracle do with their software, but it's very dependent on the version you use.

Secondly, it might help if, instead of running `runInstaller` from the root of the CD, you move into `install/linux` and run the `runInst.sh` shell script instead.

This problem seems more common on RedHat 6.1 than 6.0 and could be something to do with a newer C library.

I've also heard reports that if you have the wrong version of Gnome's usual window manager, Enlightenment, you might get this problem. Upgrade or switch to another environment such as KDE or Fvwm2.

The installer just flashes on the screen and then vanishes

This is not an uncommon occurrence. Usually it means that you're running an old version of Enlightenment. Upgrading or switching to another environment should fix the problem.

A similar problem is the installation program vanishing at some later point in the process, often around 80% of the way through. The consensus seems to be that Oracle ran out of memory. You should increase the amount of swap space your machine has, anything over 200Mb should be sufficient.

Strange Java errors when I start the installation program?

Which version of the Java Virtual Machine are you using? People have claimed success with other versions, but most of the problems that I had disappeared when I downgraded to JRE 1.1.6v5, the one that Oracle recommends in their documentation.

Two other things that are worth mentioning: make sure you use the JRE and not the JDK and, secondly, you should be using "green" threads. Unless you've set THREADS_FLAG to 'native' you almost certainly have the correct setting.

The installation program 'Segmentation Fault's

You do have GLIBC 2.1 don't you?

Problems loading shared libraries

The error message that I'm talking about looks a bit like this:

```
error in loading shared libraries: libclntsh.so.8.0: cannot open
shared object file: No such file or directory
```

This is the same as NT complaining that it can't find a DLL. It's very easy to fix. Simply add the following line to the end of your ".profile" if you're using a Bourne−like shell (ask a local guru if you don't know):

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
```

Or use the following line if you're using a CSH−like shell:

```
setenv LD_LIBRARY_PATH "$LD_LIBRARY_PATH
$ORACLE_HOME/lib"
```

I don't use the C−Shell, so independent verification of this command would be appreciated.

Pro*C doesn't work

The answer to this took quite a bit of tracking down, although the answer *is* on the Oracle web site if you look hard enough.

The default configuration of Pro*C doesn't know where to find all its libraries, so you need to tell it. After installation `$ORACLE_HOME/precomp/admin/pcscfg.cfg` is empty, but it needs to contain the following:

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

(The first four lines above, from `sys_include` to `include)` should all be on the same line in the file.)

The Oracle documentation doesn't mention this, but you also need to edit `$ORACLE_HOME/precomp/lib/env_precomp.mk`. On the line that defines `CCPSYSINCLUDE`, put the following:

```
CCPSYSINCLUDE=sys_include='($(ORACLE_HOME)/precomp/public,
/usr/lib/gcc-lib/i386-redhat-linux/egcs-2.91.66/include,
/usr/include/g++-2, /usr/include)'
```

This works for RedHat 6.0, but may need tweaking for other distributions or later versions of RH.

I installed the patch but it made things worse!

This is tricky, barely documented by Oracle and common across all their products and installation programs. It's about time they did something about it!

Often what happens is as follows: you install Oracle Enterprise Edition and, as Oracle tells you, you dash off and install all the available patches. Then you decide you need the pre–compilers and install Oracle Programmer from the same CD.

Before you installed Pro*C your database worked, and now it doesn't.

The problem is that the versions of the pre–compilers that you installed were not patched and some of the Oracle server code relies on the fixes; Oracle's installer is so stupid that it will overwrite newer version of the same code.

The solution is not pretty. Since you can't extract an individual file from the CD you need to install the whole thing again, this time adding Oracle Programmer before the patch.

Oracle thinks I don't have enough disk space

There's something wrong with the installation program. Assuming you *do* have enough space it will install okay.

4.Creating a database

4.1 Overview

Hopefully you followed the advice from the previous section and didn't create a database.

For most people, I can probably outline the process in a couple of words: "Run 'dbassist'." Unless this is the first time you've ever run Oracle, none of the questions should really phase you.

For completeness, I'll document what I did but I'd best say what I was aiming for first. Bottom line: this is neither a production system nor a 'serious' (i.e., several people, full time) development box. I installed 8i to play around and see what was new or different from 8 and older versions.

This means that when 'dbassist' offered an easy option I took it. And when it suggested using a different disk, or at least a different partition, I declined. My `$ORACLE_HOME` is `/home/oracle`. All the data files and software are in there, all on one partition.

4.2 Step−by−step guide

1. Bring up a command prompt and type:

   ```
   dbassist
   ```
2. My machine tells me that "JNLS Exception: oracle.ntpg.jnls.JNLSException. Unable to find any National Character Sets." According to Oracles 8i Patch FAQ, this is a known problem (884001) and can safely be ignored.
3. Select the "Create a database" radio button and press "Next"
4. There are two options: Typical and Custom. If you knew exactly what you were doing you probably wouldn't be reading this and could comfortably select Custom. I'm not going to cover that. Instead I'll assume you select "Typical" and press "Next"
5. Next it asks whether you want to copy the database from your CD or to create the data files. Whenever I tried the first option, Oracle couldn't find my CD player (you just installed from it!). So I recommend choosing the second option. It's not difficult, it probably just takes longer
6. It's probably safe to select 'Hybrid' when it asks you what environment the database will operate in
7. Now it asks you how many users will be using your database at any given time. I put five.
8. Next it asks you what products you want to install in your new database. Again, you know what you want better than me!
9. Oracle needs a "Global Database Name" and a "SID" now. The database name is like a fully qualified domain name (but different). If you're the Oracle guru you'll know what to put, if not your organisation might have some conventions. I called mine 'dev1' (both the SID and database name).
10. Now, do you want to create the database 'now' or should you let it save the information to a shell script? With 128Mb of RAM I found the former option painful. I created the shell script, quit out of X and anything else using a lot of memory and then ran the script. Much more snappy.
11. I didn't notice this in any of the documentation, but your database won't work properly without it! The database that 'dbassist' creates is fine, but by default the user rollback segments are left off−line. (Read: non−system users can't perform any operation that requires transactions.) Type:

    ```
    cd $ORACLE_HOME/dbs
    ```
    You now need to edit a file called "init<SID>.ora" ("initdev1.ora" in my case). About half−way down the file is a commented out line looking something like this:

```
# rollback_segments = (r01, r02, r03, r04)
```

Uncomment this line (remove the hash), save the file and you're done.

12. This is a kind of meta−step. You have a database and you should be able to start it up, but you probably don't know what any of the system passwords are! There are two that you need to know. The first is the SYSTEM password. This defaults to 'MANAGER'. (It seems to be conventional to put Oracle passwords in uppercase. In fact passwords are not case sensitive.) I recommend you change it straight away by typing password at the SQL*Plus prompt. (For people expecting an ALTER USER command, this is new to the version of SQL*Plus supplied with 8i.) The other password that you need to know is the one for SYS. It defaults to 'CHANGE_ON_INSTALL' and you should do exactly what it says!
13. Final step. This one gets rid of the annoying 'no profile' warnings you get when you log into SQL*Plus. Log into SQL*Plus as user 'system' (sqlplus system/<password>). Then type:

    ```
    @?/sqlplus/admin/pupbld.sql
    ```

    The question−mark is an alias for the $ORACLE_HOME directory.
14. This is an optional step used to define the default editor for SQL*Plus (it defaults to ed so you do!). Open $ORACLE_HOME/sqlplus/admin/glogin.sql in your favourite editor and add define_editor=<editor name> to the end.

And that's it. You should now have an operational database that you can log into using SQL*Plus.

4.3 Questions and answers

Is it really that easy?

Yes and no. If you're just playing around, building a database for yourself to learn the new features of 8i, then 'yes.' The database the above instructions will build is complete and will work fine.

However, if you know anything about Oracle, you will quickly realise that the default configuration is appallingly bad. If you're making a serious, production system I recommend you use the "Custom" option.

Even for my toy system I did some tweaking. I increased the sizes of most of the table−spaces and changed them so that they didn't grow automatically (I hate software when it tries to be too clever).

Is it really necessary to put all the files on different disks?

No and it will work fine if you don't, but I don't recommend putting all your files on the same disk nevertheless.

Spreading the files over a number of disks, even it's just the data files on one and the rollback segments on another, will have a significant performance advantage. Read an Oracle DBA book if you need further information.

I can't start dbassist

Caused by several zero−length files in the initial installation. Following the patch procedure will fix this problem.

I get "ORA−01034: ORACLE not available"

To cut a long story short, your `$ORACLE_SID` is probably set incorrectly or not at all. Make sure it's set to the same value you gave 'dbassist' and that it's value is exported (i.e., `export ORACLE_SID` in any Bourne compatible shell).

I get "ORA−01012: Not logged in"

This is a very common error, and there are a number of different things that cause it.

Firstly you'll want to make sure that you're not creating a Shared Server configuration (sometimes known as MTS). Create a database using Dedicated Server and convert it later.

If that's not it, check your `NLS_LANG` environment variable. The easiest option is to unset it. If you really want to use it, make sure that you have it exactly right. Make sure you don't transpose any '1's (one's) for 'l's (the twelfth letter of the alphabet)!

Can datafiles only be 1Gb in size?

'dbassist' won't let you create a datafile bigger than 1Gb. I believe this to be a bug as Linux has no problem with files up to 2Gb.

5.Configuration

5.1 Overview

Congratulations, you have Oracle running on your Linux box. You have created a database and can connect to it using SQL*Plus.

Of course, this is not the end of it. Ideally, you'd be able to connect to it as another Unix user or from a completely different machine. That is what this section is for.

5.2 Connecting as another user

Some of the details in this section are a little sketchy as this is not a configuration that I personally use. However, performing one of the following steps should work:

- ```
  . oraenv
  ```
  if you run a Bourne−like shell (like Bash or pdksh)
- ```
  source coraenv
  ```
  if you prefer the C−Shell

When running "oraenv" I get an error if I use 'bash', the default Linux shell. It seems not to cause any problems so don't worry. You can always use 'pdksh' if it *does* worry you.

5.3 Connecting from another machine

I remember this being very complex with earlier versions of Oracle, but just seemed to work here. I'm sure that must mean that I did something wrong, forgot something I did or that there's a massive security hole.

This is what I remember doing:

1. Logging into Linux as user 'oracle'
2. Make sure that "oraenv" has been executed (i.e., your `$ORACLE_HOME` is set correctly)
3. Type:

   ```
   lsnrctl start
   ```

On your client machine all you need to do now is point it at the right machine and database instance.

If you want more control over the process, the "Net8 Configuration Assistant" ('netec') should be able to help.

5.4 Connecting to another machine

This used to be very difficult in many earlier version of Oracle, involving editing many text files, most of which had an fantastically complex syntax.

But in 8i, if you've got your JVM working, then all you need is the "Net8 Easy Config" program. Follow these steps to allow your machine to connect to a database on another machine:

1. Start "Net8 Easy Config" by typing `netec` at the command prompt while logged in as 'oracle.'
2. After a short delay while Java gets its act together, the welcome screen appears. It should be asking what you want to do. Leave the radio buttons on the left alone (the default is 'create') and enter the name of the database in the text box. Click 'Next' when you're done.

3. Select one of the protocols it offers. Unless you know differently, this should probably be 'TCP/IP' which is the default. Press 'Next.'
4. Enter the hostname (or IP address) of the remote machine. The port number probably doesn't need changing. Press 'Next.'
5. Select the type of database (8i or other) using the radio buttons and enter the name in the appropriate text box. Press 'Next.'
6. You can now test that the information you've enter makes sense to Oracle. I found that 'netec' has a tendency to crash if some of the details are wrong. Press 'Next' when you're sure that it all works. You can keep pressing the 'Back' button to go back and correct any information.
7. If you're happy with all the information you've entered, you can press the 'Finish' button and that's it!

If you want more control over the process, you may need to use the "Net8 Assistant" −− a big window with many confusing options −− which can be started with the `netasst` command.

5.5 Questions and answers

I can't start 'netasst'

The problem is with a couple of zero−length files. Installing the patch should fix this problem.

6.Final Words

6.1 Useful Software

Now that you've managed to get Oracle installed, you'll want to try and use it. Although it's possible to do everything from your server PC, it's generally best to user the client−server facilities and use another machine to access your database.

Naturally Oracle have a large collection of, largely, pretty good client software, however there's not much for Linux at this time. Of the Oracle software, I recommend getting hold of the following:

- Oracle Enterprise Manager. It's much easier not to have to remember all those obscure ALTER USER and ALTER SYSTEM commands.
- Oracle WebDB. It can be quite difficult to install, but it can allow people to build impressive web−sites without knowing any HTML.

But most of the best software comes from other places...

- Tool for Oracle Application Development (T.O.A.D.). This used to be free but is now owned by

Quest Software. You can download a free version (if you're prepared to do it every couple of months) or you can pay for it. It's significantly more expensive than free but is not bad value.
- SQLNavigator. Also by Quest Software. I've not really used it but it's been highly recommended by all who have.
- Orac. Another that I've not used much, but has been described as a nice, configurable DBA−tool by a number of people.

6.2 Useful Books

I seem to get most of my Oracle information from colleagues and books. I'm not able to give away my colleagues, but the books I recommend are as follows:

- Oracle Performance Tuning, Mark Gurry and Peter Corrigan, O'Reilly and Associates, ISBN 1−56592−237−9.
- Oracle Design, Dave Ensor and Ian Stevenson, O'Reilly and Associates, ISBN 1−56592−268−9.
- PL/SQL Programming, Steven Feuerstein, O'Reilly and Associates, ISBN 1−56592−335−9.
- PL/SQL Built−in Packages, Steven Feuerstein, O'Reilly and Associates, ISBN 1−56592−375−8.

You'll note a bit of an O'Reilly theme there. I've not found a bad O'Reilly book yet. Similarly, I've never found a *good* Oracle Press book.

6.3 Useful Internet resources

There's a lot of useful stuff on the web.

- Oracle Technet. This is Oracle's public and free support website. Lot's of very useful information there.
- Oracle Metalink. Oracle's private (you need a support contract) support website. Only slightly more useful than Technet!
- Oracle Fans. Editorials and support forums. No official connection to Oracle (so the information there is less biased!).
- OraFaq. A site full of questions and answers regarding Oracle on all platforms.
- Oracle Linux mailing list (Send a mail to ListGuru@fatcity.com with the words 'SUBSCRIBE ORACLE−LINUX−L' in the body.
- And don't forget my website!
