

ATTENZIONE: Anche se ho impiegato cura ed impegno nella realizzazione di questo documento, consiglio vivamente chiunque voglia applicare i suggerimenti da me proposti, di eseguire tutte le opportune prove in un ambiente di test, prima di effettuare eventuali modifiche in produzione. Questa convinzione nasce dal fatto che nonostante mi sia documentato attraverso libri di testo, manuali e documenti in rete, esiste sempre la probabilità di eventuali errori nella stesura ed imprecisioni nella esposizione.

ATTENZIONE: Quanto scritto in questo documento può essere utilizzato in modo del tutto libero. Pregherei comunque tutti coloro che desiderano delle modificarlo, di comunicarlo al mio indirizzo di posta in modo tale da permettermi di aggiornare e/o correggere eventuali errori.

<http://www.oral.com> Giugno 2005

Di [Andrea Salzano](#)

Come Installare Oracle 9iR2 su RedHat AS3 (da [puschitz](#) e da [metalink](#))

1. Premessa

Il seguente documento prevede che sia stata eseguita l'installazione minima del Sistema Operativo. Pertanto dopo occorre scaricare alcuni pacchetti necessari ad Oracle (Capitolo 2). In fase di creazione dei file system inoltre, è stata costruita la partizione /sbrdbms, che sarà la radice per il software di Oracle stesso (Capitolo 3).

2. Installazione pacchetti aggiuntivi

Tre dei pacchetti elencati di seguito non sono necessari per l'installazione di Oracle in se, ma sono utili per altri motivi: xauth serve per l'autenticazione di X se ci si connette via ssh (in caso contrario l'interfaccia grafica non parte); strace per il debug; mentre le libaio per il supporto dell'I/O asincrono. Dopo aver scaricato tutti pacchetti in una directory, installarli come segue:

```
binutils-2.14.90.0.4-26.3 (già installato)
compat-db-4.0.14-5.i386.rpm (cd3)
compat-gcc-7.3-2.96.122.i386.rpm (cd2)
compat-gcc-c++-7.3-2.96.122.i386.rpm (cd2)
compat-libstdc++-7.3-2.96.123.i386.rpm (cd2)
compat-libstdc++-devel-7.3-2.96.122.i386.rpm (cd2)
glibc-devel-2.3.2-95.6.i386.rpm (cd2)
glibc-headers-2.3.2-95.6.i386.rpm (cd2)
glibc-kernheaders-2.4-8.34.i386.rpm (cd2)
libaio-0.3.96-3.i386.rpm (cd2)
libaio-devel-0.3.96-3.i386.rpm (cd2)
setarch-1.3-1.i386.rpm (già installato)
strace-4.5.1-1.EL.1.i386.rpm (cd2)
tcl-8.3.5-92.i386.rpm (cd1)
XFree86-xauth-4.3.0-55.EL.i386.rpm (cd1)
```

Dopo aver copiato tutti i pacchetti elencati in una directory d'appoggio (/tmp/appo per esempio), occorre installarli come root:

```

[root]# rpm -ivh *
Preparing... ##### [100%]
 1:compat-libstdc++ ##### [ 10%]
 2:compat-libstdc++-devel ##### [ 20%]
 3:tcl ##### [ 30%]
 4:glibc-kernheaders ##### [ 40%]
 5:glibc-headers ##### [ 50%]
 6:glibc-devel ##### [ 60%]
 7:compat-gcc ##### [ 70%]
 8:compat-db ##### [ 80%]
 9:compat-gcc-c++ ##### [ 90%]
11:Free86-xauth ##### [  %]
12:libaio ##### [  %]
13:libaio-devel ##### [  %]
10:strace ##### [100%]

```

Occorre poi eseguire la modifica di alcuni eseguibili come indicato dalla nota [252217.1](#)¹:

```

[root]# mv /usr/bin/gcc /usr/bin/gcc323
[root]# mv /usr/bin/g++ /usr/bin/g++323
[root]# ln -sf /usr/bin/gcc296 /usr/bin/gcc
[root]# ln -sf /usr/bin/g++296 /usr/bin/g++

```

3. Creazione delle directory di Oracle

In fase di installazione del sistema operativo, è stata creata una partizione /sbrdbms che conterrà il software di Oracle. Tale file system dovrà essere di proprietà dell'utente oracle. In particolare:

- /sbrdbms/oracle, è la home directory dell' utente oracle
- /sbrdbms/app/oracle è la ORACLE_BASE
- /sbrdbms/app/oracle/product/9.2.0 è la ORACLE_HOME

Creiamole:

```

[root]# mkdir -p /sbrdbms/app/oracle/product/9.2.0
[root]# mkdir -p /sbrdbms/oracle

```

4. Creazione dell'utente oracle

L'utente oracle avrà come home directory /sbrdbms/oracle e come shell la bash:

```

[root]# useradd -g 200 dba
[root]# groupadd -g 200 -u 200 -d /sbrdbms/oracle oracle

```

Verrà così creato l'utente oracle la shell /bin/bash (opzione di default). Poiché in genere, tutti gli utenti hanno come home directory /home/<utente>, per mantenere tale configurazione, creiamo un link /home/oracle verso /sbrdbms/oracle:

¹ In realtà, con l'installazione minima, sia gcc che g++ non vengono installati, per cui i due "move" indicati nella nota non sono necessari. Vengono tuttavia riportati qui solo per scopo di compatibilità con tale nota di metalink.

```
[root]# ln -s /sbrdbms/oracle /home/oracle
```

e cambiamo i permessi del file system /sbrdbms:

```
[root]# chown -R oracle:dba /sbrdbms
```

5. Installazione del software

Prima di procedere all'installazione occorre:

- scaricare la [patch](#) 3006854 da metalink, per correggere l'omonimo bug (non pubblico) 3006854. Riporto quanto indicato nel documento che accompagna tale patch:

```
[1] To apply the patch, unzip the PSE container file:  
% unzip p3006854_9204_LINUX.zip
```

```
[2] Login as the root user,
```

```
[3] Set your current directory to the directory where the patch  
is located, eg:  
# cd 3006854
```

```
[4] Run the script supplied by this patch  
# sh rhel3_pre_install.sh
```

- configure l'environment come:

```
export LD_ASSUME_KERNEL=2.4.19  
export ORACLE_BASE=/sbrdbms/app/oracle  
export ORACLE_HOME=${ORACLE_BASE}/product/9.2.0
```

6. Modifica dei parametri di kernel

Come indicato nel [manuale](#) di installazione di Oracle 9i per Linux, affinché la creazione del db vada a buon fine, è necessario modificare i parametri di kernel in modo permanente: in linea di massima infatti quelli impostati di default non sono adeguati. Il comando che permette di verificare i valori attuali² è sysctl:

```
[root]# sysctl -a  
.....cut.....  
kernel.msgmnb = 16384  
kernel.msgmni = 16  
kernel.msgmax = 8192  
kernel.shmmni = 4096  
.....cut.....
```

Per eseguire le modifiche permanentemente, basta aggiungere le seguenti righe nel file /etc/sysctl.conf:

```
# Modifica della shared memory e dei semafori  
kernel.shmmax=2147483648
```

² E' sempre possibile controllare i file di configurazione dei vari parametri sotto /proc/sys

```
kernel.shmall = 2097152
kernel.sem=250 32000 100 128
```

In ogni caso, per conoscere il nome del parametro da modificare, basta eseguire:

```
[root]# sysctl -a |grep <nome parametro>
```

prenderne il nome ed aggiungerlo in /etc/sysctl.conf, proprio come fatto per shmmax e gli altri.

7. Supporto per l'I/O asincrono

Abilitare o disabilitare il supporto del kernel di Oracle per l'I/O asincrono è piuttosto semplice. Per abilitarlo:

```
[oracle]#cd $ORACLE_HOME/rdbms/lib
[oracle]# make -f ins_rdbms.mk async_on
[oracle]#make -f ins_rdbms.mk ioracle
```

Per disabilitarlo:

```
[oracle]#cd $ORACLE_HOME/rdbms/lib
[oracle]#make -f ins_rdbms.mk async_off
[oracle]#make -f ins_rdbms.mk ioracle
```

8. Creazione del database

Questo capitolo non vuole sostituirsi al manuale di installazione fornito da Oracle. Piuttosto è pensato come integrazione. La creazione del database, procede seguendo l'interfaccia grafica del dbca:

```
[oracle]# $ORACLE_HOME/bin/dbca
```

Dopo aver navigato tra le varie finestre del tool, nella schermata finale scegliere l'opzione di salvare gli script di creazione del database e deselezionare contestualmente il flag di creazione automatica del db.

A questo punto se necessario si possono modificare i vari file sql prodotti. Sicuramente conviene modificare il file CREATEDB specificando esplicitamente il TIME_ZONE del database (vedi anche il [documento](#) TIMEZONE, da me pubblicato e presente nel sito).

Occorre poi aggiungere alcune righe nel profile dell'utente oracle (se non sono è già stato fatto):

```
export ORA_TSFILe=$ORACLE_HOME/oracore/zoneinfo/timezlrq.dat
export ORACLE_SID=<nome>
export PATH=$PATH:$HOME/bin:$ORACLE_HOME/bin
```

La variabile ORA_TSFILe serve per utilizzare il time zone esteso.

9. Appendice A: aggiunta di un file system

Supponiamo di avere sulla macchina su cui installiamo Oracle, due dischi da 35GB e che in fase di installazione del Sistema Operativo ne abbiamo utilizzato uno solo, mentre il secondo che vogliamo invece usare per il database non sia stato ancora partizionato. Di seguito è descritta la procedura su come procedere alla suo partizionamento ed alla relativa formattazione.

```
[root]# fdisk -l
```

```
Disk /dev/sda: 36.7 GB, 36703934464 bytes
255 heads, 63 sectors/track, 4462 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	127	1020096	83	Linux
/dev/sda2		128	383	2056320	83	Linux
/dev/sda3		384	511	1028160	82	Linux swap
/dev/sda4		512	4462	31736407+	5	Extended
/dev/sda5		512	575	514048+	83	Linux
/dev/sda6		576	639	514048+	83	Linux
/dev/sda7		640	4461	30700183+	83	Linux

```
Disk /dev/sdb: 36.7 GB, 36703934464 bytes
255 heads, 63 sectors/track, 4462 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
--------	------	-------	-----	--------	----	--------

```
[root]#
```

Come si può notare il disco /dev/sdb è presente ma non ancora formattato. Lanciamo allora fdisk su /dev/sdb:

```
[root@nispushtest root]# fdisk /dev/sdb
```

```
The number of cylinders for this disk is set to 4462.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)
```

```
Command (m for help):
```

Con "m" possiamo vedere il menù delle opzioni:

```
Command (m for help): m
Command action
a  toggle a bootable flag
b  edit bsd disklabel
c  toggle the dos compatibility flag
d  delete a partition
l  list known partition types
```

```
m  print this menu
n  add a new partition
o  create a new empty DOS partition table
p  print the partition table
q  quit without saving changes
s  create a new empty Sun disklabel
t  change a partition's system id
u  change display/entry units
v  verify the partition table
w  write table to disk and exit
x  extra functionality (experts only)
```

Command (m for help):

poiché vogliamo creare una nuova partizione, l'opzione da scegliere è ovviamente "n":

```
Command (m for help): n
Command action
  e  extended
  p  primary partition (1-4)
```

e quindi "p" per creare una partizione primaria (in questo caso):

```
p
Partition number (1-4):
```

Digitare "1" per creare la prima partizione:

```
Partition number (1-4): 1
First cylinder (1-4462, default 1):
```

Partiamo dal primo cilindro (opzione di default):

```
First cylinder (1-4462, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-4462, default 4462):
```

A questo punto, per decidere quanto spazio vogliamo utilizzare, possiamo specificare il numero di cilindri o la dimensione in byet, megabyte o kilobyte rispettivamente. Poiché dei 36GB, vogliamo usarne sono 30, specifichiamo la dimensione in megabyte:

```
Last cylinder or +size or +sizeM or +sizeK (1-4462, default 4462): +30720m
```

Command (m for help):

Il sistema restituisce il prompt "Command". Per verificare quanto da noi scelto, scegliamo l'opzione "p"

Command (m for help): p

Disk /dev/sdb: 36.7 GB, 36703934464 bytes
255 heads, 63 sectors/track, 4462 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	3736	30009388+	83	Linux

Command (m for help):

e per rendere definitive le nostre scelte (ricordo che da questa operazione abbiamo lasciato fuori circa 6GB di spazio), digitiamo "w":

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
[root]#

Il sistema ci ha restituito il prompt, uscendo automaticamente dal tool "fdisk". A questo punto non ci resta che formattare la partizione appena creata. Nella directory /sbin, sono presenti tutti i possibili tipi di formattazione (è possibile velocizzare la cosa editando mk e poi premendo 2 volte il tasto tab). Nel nostro caso, formattiamo il file system con mkfs.ext3 (creiamo cioè un file system di tipo ext3). Per essere certi di cosa formattare, eseguiamo nuovamente il comando fdisk:

[root]# fdisk -l

Disk /dev/sda: 36.7 GB, 36703934464 bytes
255 heads, 63 sectors/track, 4462 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	127	1020096	83	Linux
/dev/sda2		128	383	2056320	83	Linux
/dev/sda3		384	511	1028160	82	Linux swap
/dev/sda4		512	4462	31736407+	5	Extended
/dev/sda5		512	575	514048+	83	Linux
/dev/sda6		576	639	514048+	83	Linux
/dev/sda7		640	4461	30700183+	83	Linux

Disk /dev/sdb: 36.7 GB, 36703934464 bytes
255 heads, 63 sectors/track, 4462 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	3736	30009388+	83	Linux

Questa volta vediamo che sul disco /dev/sdb, esiste la partizione /dev/sdb1 che abbiamo appena creato e che dobbiamo formattare.

```
[root@nispushtest root]# mkfs.ext3 /dev/sdb1
mke2fs 1.32 (09-Nov-2002)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
3751936 inodes, 7502347 blocks
375117 blocks (5.00%) reserved for the super user
First data block=0
229 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000
```

```
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

This filesystem will be automatically checked every 31 mounts or 180 days, whichever comes first. Use `tune2fs -c` or `-i` to override.

Fatto! Dobbiamo solamente aggiornare `/etc/fstab` aggiungendo una riga in modo che la nuova partizione venga montata automaticamente al boot.

```
[root]# cat /etc/fstab
LABEL=/ / ext3 defaults 1 1
none /dev/pts devpts gid=5,mode=620 0 0
LABEL=/home /home ext3 defaults 1 2
none /proc proc defaults 0 0
none /dev/shm tmpfs defaults 0 0
LABEL=/sbrdbms /sbrdbms ext3 defaults 1 2
LABEL=/usr /usr ext3 defaults 1 2
LABEL=/var /var ext3 defaults 1 2
/dev/sda3 swap swap defaults 0 0
/dev/cdrom /mnt/cdrom udf,iso9660
noauto,owner,kudzu,ro 0 0
/dev/fd0 /mnt/floppy auto noauto,owner,kudzu 0 0
/dev/sdb1 /oradata ext3 defaults 1 1
```

L'ultima riga è quella che dobbiamo aggiungere. Nella seconda colonna è indicato il mountpoint dove verrà montata la partizione `/dev/sdb1`: in questo caso è stato scelto `/oradata`. Dobbiamo allora creare tale directory:

```
[root]# mkdir /oradata
```

A questo punto possiamo montare il file system:

```
[root@nispushtest /]# mount -a
```

Con `df -k` ci rendiamo conto della disponibilità della nuova partizione:

```
[root@nispushtest /]# df -k
Filesystem          1K-blocks      Used Available Use% Mounted on
/dev/sda1            1004024    789104    163916   83% /
/dev/sda5             497829      8239    463888    2% /home
none                 1030796         0    1030796    0% /dev/shm
/dev/sda7            30217828   5647392   23035428   20% /sbrdbms
/dev/sda2            2023952    504008    1417128   27% /usr
/dev/sda6             497829     22988    449139    5% /var
/dev/sdb1            29538432    32828   28005136    1% /oradata
```

10. Appendice B: personalizzazione dell'enviroment

L'esperienza insegna che in genere è utile personalizzare il proprio enviroment, soprattutto se si lavora in ambiente unix. Questa sezione fornisce suggerimenti ed esempi di codice, in modo da rendere più amichevole la shell che normalmente si utilizza.