Install Fedora core 5 with kernel 2.6.15-1.2054_FC5

The installation options include the selection of packages. I installed the usual set of packages including Open Office. I also included anything that looked like software development including Eclipse, windows sharing libraries and legacy software development packages. However, I excluded server type packages such as web hosting, NIS, DHCP and a few others.

Following first boot, the first task is to install the ndiswrapper software for my wireless network card.

I then compiled and installed the gkrellm utility, but this is not necessary for the Oracle installation.

Now for the ORACLE install.
Following the Puschitz online guide fairly closely...
http://www.puschitz.com/InstallingOracle10g.shtml
carry out the following steps.

1. Install packages. This is the list that I needed but depending on your selections when installing Fedora 5 will you may need a few more.
   Having downloaded them perform the following:
   ```
   rpm -ivh \
   compat-db-4.2.52-4.i386.rpm \
   gnome-libs-1.4.1.2.90-49.i386.rpm \
   libaio-0.3.106-2.2.i386.rpm \
   libpng10-1.0.18-3.2.1.i386.rpm \
   ORBit-0.5.17-15.2.2.i386.rpm \
   sysstat-6.0.1-3.2.1.i386.rpm
   ```

2. Set the kernel parameters for the Linux kernel you are running. This is done by appending the following to the /etc/sysctl.conf file
   ```
   kernel.shmmax=2147483648
   kernel.sem=250 32000 100 128
   fs.file-max=65536
   net.ipv4.ip_local_port_range=1024 65000
   net.core.rmem_default=262144
   net.core.wmem_default=262144
   net.core.rmem_max=262144
   net.core.wmem_max=262144
   ```

3. The next step is to create the oracle user
   ```
   su - root
   groupadd dba          # group of users to be granted SYSDBA system privilege
   groupadd oinstall     # group owner of Oracle files
   useradd -c "Oracle software owner" -g dba -G oinstall oracle
   passwd oracle
   ```
4. Set the oracle user shell limits. Edit the file /etc/security/limits.conf appending the following entries:
   ```
   oracle           soft    nofile          4096
   oracle           hard    nofile          63536
   oracle           soft    nproc          2047
   oracle           hard    nproc          16384
   ```

5. Reboot the machine to makes sure all the settings have taken hold. Check through a sample of kernel parameters that have been set. Use sysctl -a | grep <parameter>

6. Login as the user oracle and set the shell environments by modifying ~oracle/.bash_profile to including the following
```
export ORACLE_BASE=/<your oracle base>
export ORACLE_SID=<db sid>
. oraenv
```
7. Ensure the user oracle has permissions to write on the $ORACLE_BASE directory and the location for storing database files.
8. Make a backup copy of the file /etc/redhat-release file and replace the contents of the original file with the following
```
Red Hat Enterprise Linux AS release 3 (Taroon)
```

9. As the user oracle run the runInstaler program to install oracle, however ensure you copy the modified version of the gennttab file (See Appendix) into $ORACLE_HOME/bin directory between 40% complete and 84% complete.

10. This completes successfully, now run DBCA to create the database.

Appendix

genttab file

```
:
#
#
# gennttab generates ntcontab.o
# usage: gennttab [ -n protocol-name  ... ]
#    -n: do not include protocol adapters with given names
#
# generates .s file for SunOS 5, AIXRIOS, SVR4_386, NCR
#           .c file for all others
#

if [ "$ORACLE_TRACE" ]; then
  set -x
fi

  UNAME=/bin/uname
  PLATFORM=`$UNAME`

if [ $PLATFORM = "SunOS" ] || [ $PLATFORM = "SOL" ]; then
 NM=/usr/ccs/bin/nm
 SED=/usr/bin/sed
 AWK=/usr/bin/awk
 GREP=/usr/bin/grep
 EXPR=/usr/bin/expr
 LN=/usr/bin/ln
 ECHO=/usr/bin/echo
elif [ $PLATFORM = "Linux" ]; then
 NM=/usr/bin/nm
 SED=/bin/sed
 AWK=/usr/bin/awk
 GREP=/bin/grep
 EXPR=/usr/bin/expr
 LN=/bin/ln
 ECHO=/bin/echo
else
 NM=/usr/bin/nm
 SED=/usr/bin/sed
 AWK=/usr/bin/awk
 GREP=/usr/bin/grep
 EXPR=/usr/bin/expr
 LN=/usr/bin/ln
 ECHO=/usr/bin/echo
fi

if [ -z "$TMPDIR" ] ;
then
  TMPDIR=/tmp

fi

VER=10

###################################
NMOPTS=

###################################

if [ "x$1" = "x-64" ] ; then
       LIB=lib64
else
       LIB=lib
fi
```

```
LIBPATH=${ORACLE_HOME}/${LIB}
LIBNET="libn${VER}"
LIBNETPATH=${LIBPATH}/${LIBNET}
TRANSPORTS="US TCP BEQ TCPS DCE NMP RAW SDP"
TtoINI="US:ntusini TCP:nttini BEQ:ntpini TCPS:ntzini DCE:ntodrini NMP:ntnini RAW:ntrini
SDP:nttsdpini"
TtoBAS="US:ntus TCP:ntt BEQ:ntp TCPS:ntz DCE:ntodr NMP:ntn RAW:ntr SDP:ntt"
TtoNAM="US:IPC TCP:TCP/IP BEQ:BEQueath TCPS:SSL DCE:DCE NMP:NamedPipes RAW:RAW SDP:SDP"
TtoLIB="US:us TCP:tcp BEQ:beq TCPS:tcps DCE:dce NMP:nmp RAW:raw SDP:tcp"

trap 'rm -f ${TMPDIR}/net$$; for T in $TRANSPORTS; do ; rm -f ${TMPDIR}/${T}$$; done ;
exit 0' 1 2 15

  $NM $NMOPTS ${LIBNETPATH}.* > $TMPDIR/net$$ 2> /dev/null

for T in $TRANSPORTS; do
# Default is to use all available drivers
  eval NTUSE$T="true"

  LIB=`$ECHO ${TtoLIB}  | $SED 's/ /\n/g' | $GREP "^${T}:" | $AWK -F: '{print $2}'`
  INI=`$ECHO ${TtoINI}  | $SED 's/ /\n/g' | $GREP "^${T}:" | $AWK -F: '{print $2}'`
  BAS=`$ECHO ${TtoBAS} | $SED 's/ /\n/g' | $GREP "^${T}:" | $AWK -F: '{print $2}'`

  if [ -z "${LIB}" ] ; then
    $LN -s $TMPDIR/net$$ $TMPDIR/${T}$$
  else

    $NM $NMOPTS ${LIBPATH}/libn${LIB}${VER}.* > ${TMPDIR}/${T}$$ 2> /dev/null

  fi

  if [ "`$GREP ${INI} ${TMPDIR}/${T}$$ 2> /dev/null`" = "" ]; then

    eval NTUSE${T}="false"
  fi
done

# clean up temp files
rm -f ${TMPDIR}/net$$
for T in $TRANSPORTS; do
  rm -f ${TMPDIR}/${T}$$
done
#
# Now begins the big divide between SunOS 5 and the rest of the world...
#
#
# Now begins the big divide between AIXRIOS and the rest of the world...
#
USE_AS=${USE_AS:=false}
if [ "$USE_AS" = "true" ] && [ "$SRCHOME" = "" ]; then
       $ECHO ""

else  # USE_AS

$ECHO '/*'
$ECHO '**'
$ECHO '** This file is GENERATED by gennttab'
$ECHO '**'
$ECHO '*/
'

$ECHO "#include <sys/types.h>
"

# the following is taken from ntdrv.h
$ECHO '/* NT driver descriptor */
struct ntdrv
{
    char   *ntdrvdes;                                /* NT driver designator */
    void  (*ntdrvini)(           /* NTCONNECT - NT driver init function */
             /*_ ptr_t gbl, struct ntpa *pactx _*/ );
    char   *ntdrvsini;                                 /* Name of INIT func */
    char   *ntdrvmod;                                   /* NT driver module */
};
```

```
typedef struct ntdrv ntdrv;
'

if [ "${NTUSEDCE}" = "true" ]
then
$ECHO "void ntodrini();"
fi

if [ "${NTUSETCP}" = "true" ]
then
$ECHO "void nttini();"
fi

if [ "${NTUSETCPS}" = "true" ]; then
$ECHO "void ntzini();"
fi

if [ "${NTUSESDP}" = "true" ]; then
$ECHO "void nttsdpini();"
fi

$ECHO "void ntrini();"

$ECHO "void ntpini();"

$ECHO "void ntusini();"

if [ "${NTUSENMP}" = "true" ]
then
$ECHO "void ntnini();"
fi

# the protocol table
$ECHO "ntdrv ntcontab[] = {"
#Add the DCE adapter
if [ "${NTUSEDCE}" = "true" ]
then
$ECHO '
 { "dce", ntodrini, "_NTODRINI", "ntodr" },
'
fi

# Add the TCP adapter
if [ "${NTUSETCP}" = "true" ]
then
$ECHO '
 { "tcp", nttini, "NTTINI", "ntt" },
'
fi

# Add the SSL adapter
if [ "${NTUSETCPS}" = "true" ]; then
$ECHO '
 { "tcps", ntzini, "NTZINI", "ntcps" },
'
fi

# Add the SDP adapter
if [ "${NTUSESDP}" = "true" ]; then
$ECHO '
 { "sdp", nttsdpini, "NTTSDPINI", "ntt" },
'
fi

# RAW Bequeath and IPC PAs always included
$ECHO '
 { "raw", ntrini, "_NTRINI", "ntr" },
'
$ECHO '
{ "beq", ntpini, "_NTPINI", "ntp" },
'
$ECHO '
{ "ipc", ntusini, "_NTUSINI", "ntus" },
'
```

```
# Add the Named pipes adapter
if [ "${NTUSENMP}" = "true" ]
then
$ECHO '
{ "nmp", ntnini, "_NTNINI", "ntn" },
'
fi
# Add the OSI adapter
if [ "${NTUSEOSI}" = "true" ]
then

$ECHO '
{  "osi4", snto4in, "_snto4in",  "nto4" },
'

fi
# Add the LU6.2 adapter
if [ "${NTUSELU62}" = "true" ]
then
$ECHO '
{  "lu62", ntlini, "_ntlini",  "ntl" },
'
fi
# Add the async adapter
if [ "${NTUSEASYNC}" = "true" ]
then
$ECHO '
{  "async", ntasgini, "_ntasgini",  "ntas" },
'
fi

$ECHO " { 0, 0, 0, 0}};"

$ECHO '/* no. of connect table entries */
'

$ECHO "const int ntconent = sizeof(ntcontab)/sizeof(ntcontab[0]) - 1;"

fi # USE_AS
```