UNIX Commands for DBAs

This article contains a brief list of commands that most UNIX DBAs will need on a regular basis. Over time I've been adding more Linux-related entries.

Basic File Navigation

The "pwd" command displays the current directory.

```
root> pwd
/u01/app/oracle/product/9.2.0.1.0
```

The "ls" command lists all files and directories in the specified directory. If no location is defined it acts on the current directory.

```
root> ls
root> ls /u01
root> ls -al
```

The "-a" flag lists hidden "." files. The "-l" flag lists file details.

The "cd" command is used to change directories.

```
root> cd /u01/app/oracle
```

The "touch" command is used to create a new empty file with the default permissions.

```
root> touch my.log
```

The "rm" command is used to delete files and directories.

```
root> rm my.log
root> rm -R /archive
```

The "-R" flag tells the command to recurse through subdirectories.

The "mv" command is used to move or rename files and directories.

```
root> mv [from] [to]
root> mv my.log my1.log
root> mv * /archive
root> mv /archive/* .
```

The "." represents the current directory.

The "cp" command is used to copy files and directories.

```
root> cp [from] [to]

root> cp my.log my1.log

root> cp * /archive
root> cp /archive/* .
```

The "mkdir" command is used to create new directories.

```
root> mkdir archive
```

The "rmdir" command is used to delete directories.

```
 root> rmdir archive
```

The "find" command can be used to find the location of specific files.

```
root> find / -name dbmspool.sql
root> find / -print | grep -i dbmspool.sql
```

The "/" flag represents the staring directory for the search. Wildcards such as "dbms*" can be used for the filename.

The "which" command can be used to find the location of an executable you are using.

```
oracle> which sqlplus
```

The "which" command searches your PATH setting for occurrences of the specified executable.

File Permissions

See Linux Files, Directories and Permissions.

The "umask" command can be used to read or set default file permissions for the current user.

```
root> umask 022
```

The umask value is subtracted from the default permissions (666) to give the final permission.

```
666 : Default permission
022 : - umask value
644 : final permission
```

The "chmod" command is used to alter file permissions after the file has been created.

```
root> chmod 777 *.log

Owner       Group       World       Permission
=========   =========   =========   ======================
7 (u+rwx)   7 (g+rwx)   7 (o+rwx)   read + write + execute
6 (u+rw)    6 (g+rw)    6 (o+rw)    read + write
5 (u+rx)    5 (g+rx)    5 (o+rx)    read + execute
4 (u+r)     4 (g+r)     4 (o+r)     read only
2 (u+w)     2 (g+w)     2 (o+w)     write only
1 (u+x)     1 (g+x)     1 (o+x)     execute only
```

Character eqivalents can be used in the chmod command.

```
root> chmod o+rwx *.log
root> chmod g+r   *.log
root> chmod -Rx   *.log
```

The "chown" command is used to reset the ownership of files after creation.

```
root> chown -R oinstall.dba *
```

The "-R" flag causes the command ro recurse through any subdirectories.

OS Users Management

See Linux Groups and Users.

The "useradd" command is used to add OS users.

```
root> useradd -G oinstall -g dba -d /usr/users/my_user -m -s
/bin/ksh my_user
```

- The "-G" flag specifies the primary group.
- The "-g" flag specifies the secondary group.
- The "-d" flag specifies the default directory.
- The "-m" flag creates the default directory.

- The "-s" flag specifies the default shell.

The "usermod" command is used to modify the user settings after a user has been created.

```
root> usermod -s /bin/csh my_user
```

The "userdel" command is used to delete existing users.

```
root> userdel -r my_user
```

The "-r" flag removes the default directory.

The "passwd" command is used to set, or reset, the users login password.

```
root> passwd my_user
```

The "who" command can be used to list all users who have OS connections.

```
root> who
root> who | head -5
root> who | tail -5
root> who | grep -i ora

root> who | wc -l
```

- The "head -5" command restricts the output to the first 5 lines of the who command.
- The "tail -5" command restricts the output to the last 5 lines of the who command.
- The "grep -i ora" command restricts the output to lines containing "ora".
- The "wc -l" command returns the number of lines from "who", and hence the number of connected users.

Process Management

See Linux Process Management (ps, top, renice, kill).

The "ps" command lists current process information.

```
# ps
# ps -ef | grep -i ora
# ps -ef | grep -i ora | grep -v grep # ps -ef | grep -i [o]ra
```

Specific processes can be killed by specifying the process id in the kill command.

```
# kill 12345
# kill -9 12345
```

You can kill multiple processes using a single command by combining "kill" with the "ps" and "awk" commands.

```
# kill -9 `ps -ef | grep ora | awk '{print $2}'`
```

uname and hostname

The "uname" and "hostname" commands can be used to get information about the host.

```
root> uname -a
OSF1 oradb01.lynx.co.uk V5.1 2650 alpha

root> uname -a | awk '{ print $2 }' oradb01.lynx.co.uk
root> hostname oradb01.lynx.co.uk
```

Error Lines in Files

You can return the error lines in a file using.

```
root> cat alert_LIN1.log | grep -i ORA-
```

The "grep -i ORA-" command limits the output to lines containing "ORA-". The "-i" flag makes the comparison case insensitive. A count of the error lines can be returned using the "wc" command. This normally give a word count, but the "-l" flag alteres it to give a line count.

```
root> cat alert_LIN1.log | grep -i ORA- | wc -l
```

Remove Old Files

The `find` command can be used to supply a list of files to the `rm` command or the "-delete" command can be used directly.

```
find /backup/logs/ -name daily_backup* -mtime +21 -exec rm -f {}
;

find /backup/logs/daily_backup* -mtime +5 -exec rm -f {} \;
 find /backup/logs/daily_backup* -mtime +5 -delete;
```

File Exists Check

The Bash shell allows you to check for the presence of a file using the "[-e filepath]" comparison. In the following script a backup log is renamed if it is present and files older than 30 days are deleted are deleted.

```
#!/bin/bash
if [ -e /tmp/backup.log ]; then
  DATE_SUFFIX=`date +"%Y"-"%m"-"%d"`   mv /tmp/backup.log
/tmp/backup-$DATE_SUFFIX.log fi

# Delete old log files. find /tmp/backup*.log -mtime +30 -
delete;
```

This is one example of a log rotation, where the most current log doesn't include the date in it's name.

Rotate Log Files

See the previous section for another variant on log rotation.

The following script provides an example of how to manage a log rotation using the Bash shell. The log file includes the date in the file name. Files older than 30 days are deleted.

```
#!/bin/bash
DATE_SUFFIX=`date +"%Y"-"%m"-"%d"`
LOG_FILE=/tmp/backup-$DATE_SUFFIX.log
```

```
# Do something that needs logging.
echo "Send this to log" >> $LOG_FILE 2>&1

# Delete old log files. find /tmp/backup*.log -mtime +30 -
delete; Find Big Files
```

Find Big Files

Find the top 20 biggest files recursively from this directory.

```
$ find . -type f -print0 | xargs -0 du -h | sort -hr | head -20
```

Perform Action for Every File in a Directory

The following scripts shows two methods for performing an action for each file in a directory.

```
#!/bin/bash
for FILE in `ls /tmp/`;
do echo $FILE; done

# Do something with the file name.

# Or this.
for FILE in $( ls /tmp/ );
do   echo $FILE; done
```

Perform Action for Every Line in a File

The following scripts shows a method for performing an action for each line in a file.

```
#!/bin/bash

while read LINE; do
  # Do something with the line.

echo $LINE; done < /tmp/myfile.txt
```

alias

An alias is a named shortcut for a longer command using the following format.

```
alias name='command'
```

For example, if you require sudo access for a specific command, you might want to include this as an alias so you don't have to remember to type it.

```
alias myscript='sudo -u oracle /path/to/myscript'
```

Remove DOS CR/LFs (^M)

Remove DOS style CR/LF characters (^M) from UNIX files using.

```
sed -e 's/^M$//' filename > tempfile
```

The newly created tempfile should have the ^M character removed.

Where available, it is probably better to use the `dos2unix` and `unix2dos` commands.

Run Commands As Oracle User From Root

The following scripts shows how a number of commands can be run as the "oracle" user the "root" user.

```
#!/bin/ksh

su - oracle <<EOF
ORACLE_SID=LIN1; export ORACLE_SID
rman catalog=rman/rman@w2k1 target=/ cmdfile=my_cmdfile
log=my_logfile append  EOF
```

This is often necessary where CRON jobs are run from the root user rather than the oracle user.

Compress Files

See Linux Archive Tools (tar, star, gzip, bzip2, zip, cpio).

In order to save space on the filesystem you may wish to compress files such as archived redo logs. This can be using either the `gzip` or the `compress` commands. The `gzip` command results in a compressed copy of the original file with a ".gz" extension. The `gunzip` command reverses this process.

```
gzip myfile

gunzip myfile.gz
```

The `compress` command results in a compressed copy of the original file with a ".Z" extension. The `uncompress` command reverses this process.

```
compress myfile

uncompress myfile
```

General Performance vmstat

Reports virtual memory statistics.

```
# vmstat 5 3

procs -----------memory---------- ---swap-- -----io---- --
system-- -----cpu--
----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in
cs us sy id wa st
 0  0      0 1060608  24372 739080    0    0  1334    63 1018
1571 14 11 66 10  0
 0  0      0 995244  24392 799656    0    0  6302   160 1221
1962 10 10 62 18
0
 0  0      0 992376  24400 799784    0    0     1    28  992
1886  3  2 95  0
0
#
```

See the vmstat man page.

free

Reports the current memory usage. The "-/+ buffers/cache:" line represents the true used and free memory, ignoring the Linux file system cache.

```
# free

             total       used       free     shared    buffers
cached Mem:       8178884    4669760    3509124          0
324056    1717756
-/+ buffers/cache:    2627948    5550936
Swap:     10289148          0   10289148
```

iostat

Reports I/O statistics.

```
# iostat
Linux 3.2.10-3.fc16.x86_64 (maggie.localdomain)   03/19/2012
     _x86_64_(4 CPU)
 avg-cpu:  %user   %nice %system %iowait  %steal   %idle
2.02    0.23    0.51    0.78    0.00   96.46

Device:            tps    kB_read/s    kB_wrtn/s    kB_read
kB_wrtn sda                  9.23       100.55        62.99
1796672    1125538 dm-0             13.60       100.31
62.99    1792386    1125524 dm-1              0.02         0.08
0.00       1432           0

#
```

CPU Usage

See Linux Process Management (ps, top, renice, kill).

sar

On Linux systems sar (System Activity Reporter) is probably one of the simplest and most versatile tools for reporting system utilization including CPU, memory, disk and network activity. It automatically collects system activity statistics when installed using the following command.

```
# yum install sysstat
```

The `sar` command syntax takes the following form.

```
# sar [options] [interval [count]]
```

The "options" parameters determine what is reported, which will be discussed later. The "interval" parameter indicates the time interval in seconds between samples. The "count" parameter indicates the number of samples that will be taken before the command ends. If "count" is omitted, the sampling will continue indefinitely. If both "interval" and "count" are

omitted, the command will report the values from the 10 minute samples taken since the machine was last restarted.

As seen in the sar man page, there are lots of available options, but some starting points you may find interesting include:

```
CPU:
Basic CPU: sar [-u] [interval [count]]

Load Average: sar -q [interval [count]]
Memory:
Kernel Paging: sar -B [interval [count]]

Unused Memory: sar -r [interval [count]]

Swap Space: sar -S [interval [count]]

Disk:
Average Disk I/O: sar -b [interval [count]]

Disk I/O: sar -dp [interval [count]]

Network:
Network: sar -n DEV [interval [count]]

Network Errors: sar -n EDEV [interval [count]]
```

Here is an example of the output from a CPU report.

```
# sar -u 1 5
Linux 2.6.32-100.0.19.el5 (ol5-112.localdomain)   06/27/2011

03:10:07 PM     CPU     %user     %nice   %system   %iowait
%steal     %idle
03:10:08 PM     all      0.00      1.01     23.23     75.76
0.00
0.00
03:10:09 PM     all      0.00      1.02     35.71     63.27
0.00
0.00
03:10:10 PM     all      0.98      3.92     35.29     59.80
0.00      0.00
03:10:11 PM     all      0.00      1.03     29.90     69.07
0.00      0.00
03:10:12 PM     all      0.00      2.00     35.00     63.00
```

```
0.00      0.00
Average:        all      0.20      1.81     31.85     66.13
0.00
0.00
#
```

mpstat

Reports processor related statistics.

```
# mpstat 10 2
Linux 2.6.32-100.0.19.el5 (ol5-112.localdomain)   06/27/2011
01:59:57 PM  CPU   %user   %nice    %sys %iowait    %irq   %soft
%steal
%idle    intr/s
02:00:07 PM  all    1.21    0.00    0.90    0.20    0.00    0.00
0.00   97.69    980.50
02:00:17 PM  all    0.70    0.00    0.40    0.00    0.00    0.10
0.00   98.79    973.77
Average:     all    0.95    0.00    0.65    0.10    0.00    0.05
0.00
98.24    977.14 #
```

See the mpstat man page. **top**

Displays top tasks.

```
# top
top - 13:58:17 up 2 min,  1 user,  load average: 2.54, 1.11,
0.41 Tasks: 160 total,   6 running, 154 sleeping,   0 stopped,
0 zombie
Cpu(s): 77.1%us, 22.6%sy,  0.0%ni,  0.0%id,  0.0%wa,  0.3%hi,
0.0%si,  0.0%st
Mem:   2058872k total,   879072k used,  1179800k free,    23580k
buffers Swap:  4095992k total,        0k used,  4095992k free,
620116k cached
  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+
COMMAND
 2882 oracle    20   0  610m  64m  56m R 24.9  3.2   0:02.20
oracle  2927 root      20   0 90328 3832 2604 R 24.6  0.2
0:00.89 Xorg
 2931 oracle    20   0  605m  34m  31m R 11.5  1.7   0:00.35
```

```
oracle
 2933 oracle    20   0  605m  34m  30m S  9.8  1.7   0:00.30
oracle
 2888 oracle    20   0  614m  52m  40m S  6.9  2.6   0:00.78
oracle
 2935 oracle    20   0  604m  22m  20m S  6.2  1.1   0:00.19
oracle
 2937 oracle    20   0  604m  19m  17m R  4.6  1.0   0:00.14
oracle
 2688 oracle    -2   0  603m  15m  13m S  4.3  0.8   0:01.08
oracle
 2685 oracle    20   0  603m  15m  13m S  0.7  0.8   0:00.22
oracle
 2939 oracle    20   0  217m 4084 3504 R  0.7  0.2   0:00.02
oracle
 2698 oracle    20   0  604m  18m  16m S  0.3  0.9   0:00.17
oracle
 2702 oracle    20   0  609m  22m  14m S  0.3  1.1   0:00.17
oracle
 2704 oracle    20   0  618m  21m  19m S  0.3  1.1   0:00.21
oracle
 2714 oracle    20   0  603m  20m  18m S  0.3  1.0   0:00.18
oracle
root      20   0 10364  704  588 S  0.0  0.0   0:00.36 init
root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
root      RT   0     0    0    0 S  0.0  0.0   0:00.00
migration/0
root      20   0     0    0    0 S  0.0  0.0   0:00.00
ksoftirqd/0
root      RT   0     0    0    0 S  0.0  0.0   0:00.00
watchdog/0
root      20   0     0    0    0 S  0.0  0.0   0:00.03 events/0
root      20   0     0    0    0 S  0.0  0.0   0:00.00 cpuset
root      20   0     0    0    0 S  0.0  0.0   0:00.00 khelper
root      20   0     0    0    0 S  0.0  0.0   0:00.00 netns #
```

The **PID** column can then be matched with the **SPID** column on the V$PROCESS view to provide more information on the process.

```
SELECT a.username,
       a.osuser,
       a.program,         spid,         sid,
       a.serial# FROM   v$session a,        v$process b WHERE
```

```
a.paddr = b.addr
AND     spid = '&pid'; See the top man page.
```

the top man page.

Hide Passwords

You may be required to use passwords in scripts calling Oracle tools, like SQL*Plus, Export/Import and RMAN etc. One method to remove the credentials from the script itself is to create a credentials file to hold them. In this case I'm using "/home/oracle/.scottcred", which contains the following.

```
scott/tiger
```

Change the permissions to make sure the file is only visible to the owner.

```
$ chmod 600 /home/oracle/.scottcred
```

Now replace references to the credentials with the contents of the file.

```
$ expdp < /home/oracle/.scottcred

schemas=SCOTT directory=DATA_PUMP_DIR
dumpfile=SCOTT.dmp logfile=expdpSCOTT.log
```

Alternatively, consider using one of the following:

- Secure External Password Store
- OS Authentication

Automatic Startup Scripts on Linux

This text has been replaced by a separate article here.

CRON

See CRON : Scheduling Tasks on Linux.

There are two methods of editing the crontab file. First you can use the "crontab -l > filename" option to list the contents and pipe this to a file. Once you've editied the file you can then apply it using the "crontab filename".

```
Login as root
crontab -l > newcron
Edit newcron file.
crontab newcron
```

Alternatively you can use the "crontab -e" option to edit the crontab file directly.

The entries have the following elements.

```
field          allowed values -----          --------------
minute         0-59 hour           0-23 day of month   1-31
month          1-12
day of week    0-7 (both 0 and 7 are Sunday) user
Valid OS user command        Valid command or script.
```

The first 5 fields can be specified using the following rules.

```
*       - All available values or "first-last".
3-4     - A single range representing each possible from the
start to the end of the range inclusive.
1,2,5,6 - A specific list of values.
1-3,5-8 - A specific list of ranges.
0-23/2  - Every other value in the specified range.
```

The following entry runs a cleanup script a 01:00 each Sunday. Any output or errors from the script are piped to /dev/null to prevent a buildup of mails to root.

```
0 1 * * 0 /u01/app/oracle/dba/weekly_cleanup > /dev/null 2>&1
```

To prevent a new job starting if the last run is still running, consider using flock. The job will only run if a lock can be obtained on the specified lockfile.

```
From:
0 1 * * 0 /u01/app/oracle/dba/weekly_cleanup > /dev/null 2>&1
To:
0 1 * * 0 /usr/bin/flock -n /tmp/weekly_cleanup.lockfile
/u01/app/oracle/dba/weekly_cleanup > /dev/null 2>&1
```

Cluster Wide CRON Jobs On Tru64

On clustered systems cron is node-specific. If you need a job to fire once per cluster, rather than once per node you need an alternative approach to the standard cron job. One approach is put forward in the HP best practices document (Using cron in a TruCluster Server Cluster), but in my opinion a more elegant solution is proposed by Jason Orendorf of HP Tru64 Unix Enterprise Team (TruCluster Clustercron).

In his solution Jason creates a file called /bin/cronrun with the following contents.

```
#!/bin/ksh
set -- $(/usr/sbin/cfsmgr -F raw /) shift 12
[[ "$1" = "$(/bin/hostname -s)" ]] && exit 0 exit 1
```

This script returns TRUE (0) only on the node which is the CFS serving cluster_root.

All cluster wide jobs should have a crontab entry on each node of the cluster like.

```
5 * * * /bin/cronrun && /usr/local/bin/myjob
```

Although the cron jobs fire on all nodes, the "/bin/cronrun &&" part of the entry prevents the script from running on all nodes except the current CFS serving cluster_root.

NFS Mount (Sun)

The following deamons must be running for the share to be seen by a PC.

```
/usr/lib/nfs/nfsd -a
/usr/lib/nfs/mountd
/opt/SUNWpcnfs/sbin/rpc.pcnfsd
```

To see a list of the nfs mounted drives already present type.

```
exportfs
```

First the mount point must be shared so it can be seen by remote machines.

```
share -F nfs -o ro /cdrom
```

Next the share can be mounted on a remote machine by **root** using.

```
mkdir /cdrom#1 mount -o ro myhost:/cdrom /cdrom#1
```

NFS Mount (Tru64)

On the server machine, if NFS is not currently setup do the following.

- Application Manager -> System Admin -> Configuration -> NFS
- Select the "Configure system as an NFS server" option.
  Accept all defaults.

Create mount point directory.

```
mkdir /u04/backup
```

Append the following entry to the "/etc/exports" file.

```
/u04/backup
```

Make sure the correct permissions are granted on the directory.

```
chmod -R 777 /u04/backup
```

On the client machine, if NFS is not currently setup do the following.

```
Application Manager -> System Admin -> Configuration -> NFS
Select the "Configure system as an NFS client" option.
Accept all defaults.
```

Create mount point directory.

```
mkdir /backup
```

Append an following entry to the "/etc/fstab" file.

```
nfs-server-name:/u04/backup     /backup          nfs rw,bg,intr 0
0
```

Finally, mount the fileset.

```
mount /backup
```

At this point you can start to use the mount point from your client machine. Thanks to Bryan Mills for his help with Tru64.

Samba/CIFS Mount (Linux)

See Linux Samba Configuration.

Create a directory to use for the mount point.

```
# mkdir /host
```

Add the following line to the "/etc/fstab" file.

```
//192.168.0.4/public /host          cifs
     rw,credentials=/root/.smbcred,uid=500,guid=500 0 0
```

Create a file called "/root/.smbcred" with the following contents.

```
username=myuser password=mypassword
```

Change the permissions on the credentials file.

```
# chmod 600 /root/.smbcred
```

Mount the share.

```
# mount /host
```

PC XStation Configuration

Download the CygWin setup.exe from http://www.cygwin.com.

Install, making sure to select all the X11R6 (or XFree86 in older versions) optional packages.

If you need root access add the following entry into the /etc/securettys file on each server.

```
<<client-name>:0
```

From the command promot on the PC do the following.

```
set PATH=PATH;c:cygwinbin;c:cygwinusrX11R6bin XWin.exe :0 -query
<server-name>
```

The X environment should start in a new window.

Many Linux distributions do not start XDMCP by default. To allow XDMCP access from Cygwin edit the "/etc/X11/gdm/gdm.conf" file. Under the "[xdmcp]" section set "Enable=true".

If you are starting any X applications during the session you will need to set the DISPLAY environment variable. Remember, you are acting as an XStation, not the server itself, so this variable must be set as follows.

```
DISPLAY=<client-name>:0.0; export DISPLAY
```

xauth (Magic Cookie)

Access to X servers can get broken when using `su` and `sudo` commands. The `xauth` command provides a solution to this. The process involves the following stages:

- Check your current display number.
- Use `xauth list` to get a list of magic cookies. □ Switch to the new user.
- Use `xauth add` to set the magic cookie for your display number.

An example of this is shown below.

```
$ echo $DISPLAY localhost:12.0 $ xauth list
ol6.localdomain/unix:12  MIT-MAGIC-COOKIE-1
```

```
be64852468ca3c334720b10bb3c4d3cb
$ sudo su oracle
$ xauth add ol6.localdomain/unix:12  MIT-MAGIC-COOKIE-1
be64852468ca3c334720b10bb3c4d3cb
```

You will now be able to access the X server, just as you could before the user switch.

Useful Profile Settings

See Linux Groups and Users : Important Files.

The following ".profile" settings rely on the default shell for the user being set to the Korn shell (/bin/ksh).

The backspace key can be configured by adding the following entry.

```
stty erase "^H"
```

The command line history can be accessed using the [Esc][k] by adding the following entry.

```
set -o vi
```

Auto completion of paths using a double strike of the [Esc] key can be configured by adding the following entry.

```
set filec
```

Summary:

Navigation and File Management:

```
cd: Change directory. Example: cd /home/oracle/data
ls: List directory contents. Example: ls -l (long listing
format)
pwd: Print working directory. Example: pwd
mkdir: Create a new directory. Example: mkdir backup_dir
cp: Copy files and directories. Example: cp database.sql
backup_dir
mv: Move or rename files and directories. Example: mv
```

```
archive.log /archive_logs
rm: Remove files and directories. Example: rm -rf temp_files
(use with caution!)
```

Viewing and Editing Files:

```
cat: Display file contents. Example: cat table_schema.sql
more: View files page-by-page. Example: more log_file.txt
less: View files with more navigation options. Example: less
error_report
head: View the first few lines of a file. Example: head -n 10
config.ini
tail: View the last few lines of a file. Example: tail -f
transaction_log (follow updates)
nano: Simple text editor. Example: nano script.sh
vi: Powerful text editor (steeper learning curve). Example: vi
config.xml
```

Text Manipulation:

```
grep: Search for patterns in files. Example: grep "ERROR"
log_file.txt
awk: Extract specific data from files. Example: awk '{print $4}'
employee_data.csv
sed: Modify text in files. Example: sed
's/old_value/new_value/g' config.txt
```

Running Commands and Processes:

```
ps: List running processes. Example: ps aux | grep oracle
top: Monitor system resource usage. Example: top
kill: Terminate processes. Example: kill -9 12345 (process ID)
bg: Move a process to the background. Example: bg %1
fg: Bring a process to the foreground. Example: fg %2
```

DBA-Specific Commands:

```
sqlplus: Connect to an Oracle database. Example: sqlplus / as
sysdba
rman: Oracle Recovery Manager for backup and recovery. Example:
rman target /
lsnrctl: Manage Oracle listener processes. Example: lsnrctl
start
ps -ef | grep postgres: Check for running PostgreSQL processes.
mysql: Connect to a MySQL database. Example: mysql -u root -p
```

Useful Files

Here are some files that may be of use.

Path	Contents
`/etc/passwd`	User settings
`/etc/shadow`	Where encrypted user passwords are stored
`/etc/group`	Group settings for users.
`/etc/hosts`	Hostname lookup information.
`/etc/system`	Kernel parameters for Solaris.
`/etc/sysconfigtab`	Kernel parameters for Tru64.
`/etc/sysctl.conf`	Kernel parameters for Linux.

Thank you!