GNU/Linux Commands

ILG Insight GNU/Linux Group

Reinventing the way you,
Think,
Learn,
Work

www.gnugroup.org

info@gnugroup.org

Page Intentionally left blank

Chapter 1

Using Man pages

you want to find man pages for commands related to a certain word

$apropos crontab

apropos command - to search the man page database

/etc/anacrontab [anacrontab] (5) - configuration file for anacron
crontab (1) - maintain crontab files for individual
users (ISC Cron V4.1)
crontab (1p) - schedule periodic background work
crontab (5) - tables for driving cron (ISC Cron V4.1)
crontabs (rpm) - Root crontab files used to schedule the
execution of programs.

$whatis cat
cat (1) - concatenate files and print on the standard output
cat (1p) - concatenate and print files

$man find

FIND(1) FIND(1)
NAME
find - search for files in a directory hierarchy
SYNOPSIS
find [-H] [-L] [-P] [path...] [expression]

Section	Description
1	**General user commands**
2	**System calls**
3	**Programming routines / library functions**
4	**Special files**
5	**Configuration files and file formats**
6	**Games**
7	**Miscellaneous**
8	**Administrative commands and daemons**

$ **man mount -a** *Shows all man pages related to component*
$ **man 5 crontab** *Shows section 5 man page for component*
$ **man mount -P more** *Use more, not less to page through*
$ **man --path** *List locations of man directories*
/usr/kerberos/man:/usr/local/share/man:/usr/share/man/en:
/usr/share/man:/usr/X11R6/man:/usr/local/man
$ **man -f mount** *Same as the whatis command*
$ **man -k mount** *Same as the apropos command*

Converting output of man to html

$ **whereis -m cat**
cat: /usr/share/man/man1/cat.1.gz /usr/share/man/man1p/cat.1p.gz
$ **cd /tmp ; cp /usr/share/man/man1/cat.1.gz .**
$ **gunzip cat.1.gz**
$ **man2html cat.1 > cat.1.html**
$ **links cat.1.html**

To display information on a command

$ **info ls**

Keystroke	Movement
?	Display the basic commands to use in info windows.
Shift+l	Go back to the previous node you were viewing.
n, p, u	Go to the node that is next, previous, or up.
Tab	Go to the next hyperlink that is in this node.
Enter	Go to the hyperlink that is under the cursor.
Shift+r	Follow a cross-reference.
Shift+q	Quit and exit from info.

Chapter 2

Installing fedora and adding software

Choosing Where Fedora Software Is from the Boot Screen

boot: **linux askmethod**

Choose the methods which you would be using

Local CDROM — Continue installing from the local CD or DVD.

Hard drive — To use this method, you must copy the DVD or CD images to a local hard disk. When asked, identify the partition and directory holding the images.

NFS image — To use this method, you must copy the DVD or CD images to a directory on a computer on your LAN and share that directory using NFS. When asked, identify the NFS resource holding the images.

FTP — You can use this method to install from an existing Internet FTP mirror, or from your own in-house install point. When asked, identify the FTP site's URL and directory. To create your own FTP install point, you can, for example, copy the contents of the DVD or all CD images to a directory on your FTP server with a command such as cp –ar.

HTTP — Same as FTP, but using an HTTP web server (an existing Internet mirror or your own).

Choosing How Install Proceeds from the Boot Screen

To have the install proceed in different ways, you can add boot options. Here are examples of different install types you can request from the boot prompt:

boot: **linux text**
boot: **linux vnc**
boot: **linux vnc vncconnect=192.168.0.20 vncpassword=99pass07**
boot: **linux ks=floppy**
boot: **linux ks=hd:/dev/hda1/ks.cfg**
boot: **linux ks=http://example.com/ks.cfg**
boot:linux askmethod For askmethod remote installation

Use Linux text to run the install in text mode (if your graphical screens are garbled).

If you use Linux vnc, you can step through the graphical section of the install remotely by connecting a VNC client to the IP of the install machine. The installer will show the IP address and display to connect to after it starts the VNC server. You can also start a VNC client on your network in listening mode and point the installer to that client using vncconnect. In the second vnc example above, vncviewer -listen is running on the machine at 192.168.0.20 with a password of 99pass07.

The three ks examples tell the installer where to find a kickstart file to guide the install process.

first looks for a ks.cfg file on the local floppy disk.

second looks for ks.cfg on the first IDE hard disk partition, and the

Third looks for ks.cfg in the root of the web server at example.com.

A kickstart file contains information that lets the install process bypass some or all questions asked during installation. A sample kickstart
file can be found in /root/anaconda-ks.cfg after a Fedora install is completed. Using that file, you can repeat the install done on that machine on another computer.

boot: linux rescue
boot: linux local
boot: linux memtest86

The rescue option starts a mini–Linux system in rescue mode, so you can mount file systems and fix problems from the command line.

The local option bypasses the CD/DVD and tries to boot from hard disk.

The memtest86 option checks your computer’s memory.

Installing, managing, removing software with YUM

Finding Packages

yum for finding information about specific packages or searching yum repositories for specific packages or components. Use the list option to list packages meeting your criteria

# **yum list available**	List packages available to be installed
# **yum list installed**	List packages already installed
# **yum list extras**	List packages not installed from any repo
# **yum list *vorbis***	List packages with “vorbis” in title
# **yum list updates**	List packages that have updates available
# **yum grouplist**	List packages in group
# yum repolist all	List all the repos

Use the info option to see package descriptions from repos.

# **yum info zsh**	Description for zsh package
# **yum info word***	Descriptions for packages beginning with “word”

To search packages for a string that appears in the description, packager, package name, or summary of the package, use the search option as follows:

# **yum search vim**	Search for packages including the “mp3” string

To search packages for a file or other feature and list the packages found, use the *whatprovides* option.

 # yum whatprovides /bin/cat
 # yum whatprovides /etc/sysctl.conf

Installing packages

 # yum install zsh
 # yum install mysql mysql-devel

yum to install a package from a directory on the local computer,

 # yum localinstall heyu-2.0beta.3.1-1.i386.rpm

to install all packages in an installation group.

 # yum groupinstall "Virtualisation Tools" "Virtualization Platform"

Updating Packages

If updates are available, you can update a single package, group of packages, or all packages.

# **yum check-update**	Lists all packages with updates ready
# **yum list updates openoffice***	Find available openoffice* updates
# **yum update openoffice***	Update all openoffice packages
# **yum update**	Update all packages with updates ready
# **yum groupupdate XFCE**	Update all packages in XFCE group

Removing Packages

You can remove individual packages or groups of packages. An advantage to using yum to remove packages is that it can remove dependent packages, as well as the ones you selected.

# **yum remove beagle**	Removes the beagle package
# **yum remove xscreen***	Removes packages beginning with xscreen
# **yum groupremove XFCE**	Removes all packages in XFCE group

Cleaning Up Packages

Using the clean option to yum, you can clean up *packages*, *headers*, *metadata*, *cache*, and *dbcache* left around by the yum facility.

If *keepcache* is set to 1 in /etc/yum.conf, as the packages and headers you request are downloaded, they are saved in packages and headers subdirectories of */var/cache/yum/repo/*, respectively.

Metadata are stored in repomd.xml and comps.xml files in the same directory. Here are ways of cleaning out those items:

# **yum clean packages**	Cleans out packages left over in cache
# **yum clean metadata**	Cleans out metadata left over in cache
# **yum clean headers**	Cleans out headers left over in cache
# **yum clean all**	Cleans out metadata, headers, and packages

Enabling and disabling repositories

Enabling a Yum Repository

To enable a particular repository or repositories, type the following at a shell prompt as **root**:

yum-config-manager --enable **repository**...
where ***repository*** is the unique repository ID (use **yum repolist all** to list available repository IDs).

Yumdownloader

yumdownloader is useful for downloading packages from a yum repository to the local disk

yumdownloader cacti
yumdownloader zsh

yum-utils package (yum install yum-utils), you have access to a handful of useful commands that you can use for accessing and creating yum repositories.

repoquery is a program to query information from YUM repositories, in a similar way as the queries with rpm.

List the contents of a given package.
$ repoquery --list iftop

```
/usr/sbin/iftop
/usr/share/doc/iftop-0.17
/usr/share/doc/iftop-0.17/COPYING
/usr/share/doc/iftop-0.17/ChangeLog
```

```
/usr/share/doc/iftop-0.17/README
/usr/share/doc/iftop-0.17/TODO
/usr/share/man/man8/iftop.8.gz
```

$ repoquery -q --file /usr/bin/yum

```
yum-0:3.2.25-1.fc12.noarch
yum-0:3.2.27-3.fc12.noarch
```

Managing software with RPM's

Installing a Package

The following command installs a new package located in the current directory. Options in this command include i for install, v for verbose, and h for progress hash marks.

 # rpm -ivh rpmforge-release-0.2-2.2.fc5.rf.x86_64.rpm
Preparing... ### [100%]
1:rpmforge-release ### [100%]

The following example installs a new package located on the Internet. This approach works with http and ftp protocols:

 # rpm -ivh
http://ftp.belnet.be/packages/dries.ulyssis.org/fedora/fc5/x86_64/RPMS.dries/rpmforge-release-0.2-2.2.fc5.rf.x86_64.rpm

Retrieving http://ftp.belnet.be/packages/dries.ulyssis.org/fedora/
fc5/x86_64/RPMS.dries/rpmforge-release-0.2-2.2.fc5.rf.x86_64.rpm
Preparing... ### [100%]
1:rpmforge-release ### [100%]

Upgrading a Package

If an older version of the package is already installed, an error will occur when you go to install it. Use rpm -Uvh to upgrade an existing package to a newer version.

 # rpm -Uhv flash-plugin-9.0.31.0-release.i386.rpm

Removing a Package

 # rpm -e rpmforge-release

Sometimes, such as on 64-bit systems that have 32-bit packages installed for backwards compatibility, you may have two or more versions of a package installed. If you get an error when trying to remove one, you might be able to fix that using a full package name or by removing all matching packages:

 # rpm -e --nodeps avahi-0.6.11-3.fc5.i386

Querying Information about RPM Packages

This shows how to query installed packages for a package named rsync and display version information about that package (your version numbers may be different):

 # rpm -q rsync
rsync-2.6.9-1.FC5.1
U

Set the -qp option to get information about an RPM in the present directory:

 # rpm -qp rpmforge-release-0.2-2.2.fc5.rf.x86_64.rpm

list of all the packages installed on your system

 # rpm -qa | less
glibc-2.5.90-15
libICE-1.0.3-1.fc7

Check a file on your system to see what package the file belongs to, if any:
 # rpm -qf /etc/sysctl.conf
initscripts-8.31.6-1

You want to query for information on package.

 # rpm -qi rpmforge-release

Combine various query options to check an RPM file before it's installed:

 # rpm -qilp rpmforge-release-0.2-2.2.fc5.rf.x86_64.rpm | less

lists preinstall and postinstall scripts that come with an installed RPM package:

 # rpm -q --scripts kernel | less

Verifying Installed Packages

Letter indicating check failure	Description
S	File size differs
M	Mode differs: includes permissions and file type
5	MD5 checksum differs
U	User ownership differs
G	Group ownership differs
T	mTime (timestamp of last modification) differs

 # rpm -Va | grep bin
S.5....T /usr/bin/curl

# rpm -Vv coreutils	Verbose check files from coreutils
# rpm -V -f /usr/bin/pr	Verify package containing pr
# rpm -V -g Applications/Multimedia	Verify packages from selected group

Rebuilding Your RPM Database

If your RPM database becomes corrupted to the point where you can no longer install packages, you can rebuild the database from the installed package headers. First remove the old database files, and then rebuild the new ones as follows:

 # rm /var/lib/rpm/__db.00* **remove the db files**
 # rpm –rebuilddb **Rebuild the DB.**

Building RPMs from SRPMs

By rebuilding the source code that is used to build an RPM package, you can change it to better suit the way you use the software. To begin, you need to get the source RPMs (SRPMs) you want to modify and install the rpm-build package (yum install rpm-build). For example, you could download and install the rpmforge-release SRPM package in the current directory by typing the following command:

 # wget http://dag.wieers.com/rpm/packages/rpmforge-release/rpmforge-release-0.3.6-1.rf.src.rpm
 # mkdir -p /usr/src/redhat/SOURCES
 # rpm -ivh rpmforge-release-0.3.6-1.rf.src.rpm

When a source code package (src.rpm) is installed, rpm places the files it contains in the default build tree under the /usr/src/redhat directory. If you have software development tools packages and the rpm-build package installed, you can rebuild the binary RPM from this package. You can make changes to the spec file or the source code of that package, and then rebuild the package using the command shown in the following example:

 # rpmbuild -bb /usr/src/redhat/SPECS/rpmforge-release.spec

The result of this command is an RPM file that is output to a directory that is specific to your computer architecture: /usr/src/redhat/RPMS/arch, where arch is replaced by a name indicating the computer architecture (such as i386, i586, and so on). The resulting RPM file is ready to be installed.

Chapter 3

Using the shell

Shell environment is set up based on the user who started the shell. Bash shell settings –

for all users' shells are located in /etc/bashrc, /etc/profile, and /etc/profile.d/

User-specific shell settings are determined by commands executed from several dot files in the user's home directory : .bash_profile, .bash_login, and ,.profile.

When a shell is closed, any commands in the user's ~/.bash_logout file are executed.

Command Line Completion

Tab key to complete different types of information on the command line.

$ tracer<Tab> Command completion: Completes to traceroute command
$ cd /home/ch<Tab> File completion: Completes to /home/ilg directory
$ cd ~jo<Tab> User homedir completion: Completes to /home/john
$ echo $PA<Tab> Env variable completion: Completes to $PATH

$ ping <Alt+@><Tab> Host completion: Show hosts from /etc/hosts

Redirecting stdin and stdout

$ ls /tmp /tmmp > output.txt
ls: /tmpp: No such file or directory

stdout is redirected to the file output.txt, while stderr is still directed to the screen.

$ ls /tmp /tmmp 2> errors.txt

stderr (stream 2) is directed to errors.txt while stdout goes to the screen.

$ ls /tmp /tmmp 2> errors.txt > output.txt
the first two examples are combined.

$ ls /tmp /tmmp > everything.txt 2>&1

directs both streams to the everything.txt file.

To append to a file instead of overwriting it, use two greater-than signs:
$ ls /tmp >> output.txt

If you don't ever want to see an output stream, you can simply direct the output stream to a special bit bucket file (/dev/null)
$ ls /tmp 2> /dev/null

Direct standard input to a command.
$ mail ilg@gnugroup.org < /etc/hosts
Above e-mails the /etc/hosts file to the user named ilg on the local system

Redirect output from one process to another process

$ ls /tmp | sort

$ ls /tmp/ /tmmp 2> /dev/null | sort

Few more examples

$ rpm -qa | grep -i sql | wc -l
$ ps auwx | grep firefox
$ ps auwx | less
$ whereis -m yum | awk '{print $2}'

Using backticks, you can execute one section of a command line first and feed the output of that
command to the rest of the command line

$ rpm -qf `which ps`
$ ls -l `which traceroute`

Using alias

aliases are already set in your system's /etc/bashrc or /etc/profile.d/* files or the user's ~/.bashrc file.

```
$ alias
alias cp='cp -i'
alias l.='ls -d .* --color=tty'
alias ll='ls -l --color=tty'
alias ls='ls --color=tty'
alias mv='mv -i'
alias rm='rm -i'
```

Setting Alias

```
$ alias la='ls -la'
```

unalias

$ unalias la	Unalias the previously aliased la command
$ unalias -a	Unalias all aliased commands

Watching

keep an eye on a command whose output is changing

```
$ watch 'cat /proc/loadavg'
```

Use Ctrl+c to quit

```
$ watch -n 10 'ls -l'
$ watch -d 'ls -l'
$ watch 'ls –l mydownload.iso'
```

Watch the logs in real time

```
# tail -f /var/log/messages
```

Super User Power

su Command

```
$ su -
Password:*****
```

Difference between running $ su and $ su - ?

PATH and $PATH ?

Delegating Power with sudo

sudo command is configured in /etc/sudoers. Edit this file with visudo command

Every use of sudo gets logged in /var/log/secure:

```
$ sudo -u ilg  /bin/ls /home/ilg
```

Display all of the environment variables.

```
$ set | less
BASH=/bin/bash
COLORS=/etc/DIR_COLORS.xterm
COLUMNS=118
```

DISPLAY=:0.0
HOME=/home/ilg
HOSTNAME=ilg

make variables part of the environment and inheritable by children processes by exporting them:

$ export ABC=123
$ bash
$ echo $ABC
123

concatenate a string to an existing variable:

 # export PATH=$PATH:/home/ilg

list your bash's environment variables use:
 # env

Editing and Running a Script

$ chmod u+x myscript.sh

Start of shell script

#!/bin/bash **Sharp bang / Hash Bang / Shebang**

Executing shell script

$./myscript.sh **. / Is used because current dir is not in the PATH.**

##

Chapter 4

Working with Files

In UNIX/Linux everything is a file, if its not a file, it's a process.

Types of Files

Device files
Character device files
Block device files
Hard links and soft links
Named pie and sockets

Files that provide access to the hardware components on your computer are referred to as device files.

There are character and block devices.

There are hard links and soft links you can use to make the same file accessible from different locations.
Less often used directly by regular users are named pipes and sockets, which provide access points for processes to communicate with each other.

Regular files consist of data files (documents, music, images, archives, and so on) and commands (binaries and scripts).

$ file article* bash*
article.ms: ASCII troff or preprocessor input text
article.ps: PostScript document text conforming at level 3.0

create some blank files

$ touch /tmp/newfile.txt	Create a blank file
$ > /tmp/newfile2.txt	Create a blank file, but dangerous, as it will blank the file if file already exists

long list on a file is another way to determine its file type.

$ ls -l /tmp/newfile2.txt

Listing Files and directories

$ ls -l
-rwxr-xr-x 1 root root 1786 Feb 13 1006 /usr/bin/apropos

directory is a container for files and subdirectories. Directories are set up in a hierarchy from the root (/) down to multiple subdirectories, each separated by a slash (/). Directories are called folders when you access them from graphical file managers.

create new directories

$ mkdir /tmp/new	Create “new” directory in /tmp
$ mkdir -p /tmp/a/b/c/new	Create parent directories as needed for “new”
$ mkdir -m 700 /tmp/new2	Create new2 with drwx------ permissions

Symbolic Links

1) Symbolic link which points to a file or change to one that points to a directory,
the command you run acts on the file or directory that is the target of that link.

2) The target has its own set of permissions and ownership that you cannot see from the symbolic link.

3) The symbolic link can exist on a different disk partition than the target. In fact, the symbolic link can exist, even if the target doesn’t.

Hard links

1) Hard link can only be used on files (not directories) and is basically a way of giving multiple names to the same physical file.

2) Every physical file has at least one hard link, which is commonly thought of as the file itself.

3) Any additional names (hard links) that point to that single physical file must be on the same partition as the original target file (in fact, one way to tell that files are hard links is that they all have the same inode number).

4) Changing permissions, ownership, date/time stamps, or content of any hard link to a file results in all others being changed as well.

5) Deleting one link will not remove the file; it will continue to exist until the last link to the file is deleted.

Create hard and symbolic links:

```
$ touch myfile
$ ln myfile myfile-hardlink
$ ln -s myfile myfile-symlink
$ ls -li myfile*
292007  -rw-rw-r--  3  ilg     ilg  0   Mar 25 00:07       myfile
292007 -rw-rw-r--   3  ilg     ilg      0 Mar 25 00:07     myfile-hardlink
292008 lrwxrwxrwx 2 ilg     ilg    6 Mar 25 00:09       myfile-symlink
```

Device Files

When applications need to communicate with your computer's hardware, they direct data to device files.

Device files are stored in the /dev directory.

Devices are generally divided into block devices (such as storage media) and character devices (such as serial ports and terminal devices).

Each device file is associated with a major number (indicating the type of device) and minor number (indicating the instance number of the device)

Terminal (tty) devices are represented by major character device 4, while SCSI hard disks are represented by major block device number 8.

```
$ ls -l /dev/tty0 /dev/sda1                    List character and block special devices
brw-r----- 1 root disk 8, 1 2007-09-05 08:34 /dev/sda1
crw-rw---- 1 root root 4, 0 2007-09-05 08:34 /dev/tty0
```

create your own device file

```
# mknod  /dev/ttyS4 c 4 68                     Add device for fifth serial port

$ ls -l /dev/ttyS4                                                  List new device file
crw-r--r--      1       root        root        4, 68     Sep 6 00:35 /dev/ttyS4
```

Named Pipes and Sockets

To provide a presence in the file system from which a process can communicate with other processes, you can create named pipes or sockets.

Named pipes are typically used for interprocess communication on the local system,

Sockets can be used for processes to communicate over a network.

create your own named pipe, use the mkfifo command

```
$ mkfifo mypipe
$ ls -l mypipe
prw-rw-r-- 1 ilg ilg 0 Sep 26 00:57 mypipe
```

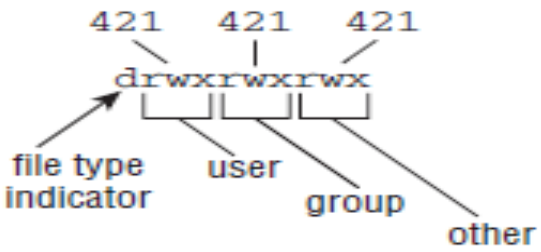
To create your own socket, use the mksock command as follows:

```
$ /usr/sbin/mksock mysock
$ ls -l mysock
srwxrwxr-x 1 ilg ilg 0 Sep 26 00:57 mysock
```

Its usually done when you developing applications,

File Permissions Octal and Alphabetically.

Setting File/Directory Permissions



Changing Permissions with chmod

chmod command (octal or letters)	Original Permission	New Permission	Description
chmod 0700	any	drwx------	The directory's owner can read or write files in that directory as well as change to it. All other users (except root) have no access.
chmod 0711	any	drwx--x--x	Same as for owner. All others can change to the directory, but not view or change files in the directory. This can be useful for server hardening, where you prevent someone from listing directory contents, but allow access to a file in the directory if someone already knows it's there.
chmod go+r	drwx------	drwxr--r--	Adding read permission to a directory may not give desired results. Without execute on, others can't view the contents of any files in that directory.
chmod 0777 chmod a=rwx	any	drwxrwxrwx	All permissions are wide open.
chmod 0000 chmod a-rwx	any	d---------	All permissions are closed. Good to protect a directory from errant changes. However, backup programs that run as non-root may fail to back up the directory's contents.
chmod 666	any	-rw-rw-rw-	Open read/write permissions completely on a file.
chmod go-rw	-rw-rw-rw-	-rw-------	Don't let anyone except owner view, change, or delete the file.
chmod 644	any	-rw-r--r--	Only the owner can change or delete the file, but all can view it.

1st digit 0 in 0700 is an octal digit that can be used on commands (executables) to indicate that the command can run as a

Set-UID program (4)

Run as a **set-GID program (2).**

Sticky program (1).

With set-UID and set-GID, the command runs with the assigned user or group permissions (instead of running with permission of the user or group that launched the command)

sticky bit on for a directory keeps users from removing or renaming files from that directory that they don't own. **Alpha = t**

Check the permissions on the **/tmp** directory.

Recusrsive chmod

# chmod -R 700 /tmp/test	Open permission only to owner below /tmp/test
# chmod -R 000 /tmp/test	Close all permissions below /tmp/test
# chmod -R a+rwx /tmp/test	Open all permissions to all below /tmp/test

The umask command, you can set the permissions given to files and directories when

$ umask 0066	Make directories drwx--x--x and files -rw-------
$ umask 0077	Make directories drwx------ and files -rw-------
$ umask 0022	Make directories drwxr-xr-x and files -rw-r--r--
$ umask 0777	Make directories d--------- and files ----------

Umask for files is calculated based on 666, as on directories its 777.

Changing Ownership

# chown ilg test/	Change owner to ilg
# chown ilg:ilg test/	Change owner to ilg and group to market
# chgrp ilg test/	Change group to market
# chown -R ilg test/	Change all files below test/ to owner ilg

Traversing the File System

cd

$ cd	Change to your home directory
$ cd $HOME	Change to your home directory
$ cd ~	Change to your home directory
$ cd ~ilg	Change to ilg' home directory
$ cd -	Change to previous working directory
$ cd $OLDPWD	Change to previous working directory
$ cd ~/public_html	Change to public_html in your home directory
$ cd ..	Change to parent of current directory
$ cd /usr/bin	Change to usr/bin from root directory
$ cd usr/bin	Change to usr/bin beneath current directory

what your current directory

$ pwd	Print working directory

Copying Files

```
# cd ; touch index.html
# cp -i index.html /var/www/html/
```

```
# cp -il index.html /var/www/html
# cp -a /var/www/html /mnt/sda1/var/www/
# cp -R /var/www/html /mnt/sda1/var/www/
```

Changing File Attributes

```
# lsattr /etc/host*
------------- /etc/host.conf
------------- /etc/hosts
------------- /etc/host.allow
------------- /etc/host.deny
$ lsattr -aR /tmp/ | less          Recursively list all /tmp attributes
```

Attributes are the following:

```
a (append only),
c (compressed),
d (no dump),
I (immutable),
j (data journaling),
s (secure deletion),
t (no tail-merging),
u (undeletable),
A (no atime updates),
D (synchronous directory updates),
S (synchronous updates), and
T (top of directory hierarchy)

# chattr +i /boot/grub/grub.conf
$ chattr +A -R /home/ilg/images/*
$ chattr +d FC6-livecd.iso
$ lsattr /boot/grub/grub.conf /home/ilg/images/* FC6-livecd.iso
----i-------- /boot/grub/grub.conf
-------A----- /home/ilg/images/einstein.jpg
-------A----- /home/ilg/images/goth.jpg
------d------ FC6-livecd.iso
```

remove an attribute

```
# chattr -i /boot/grub/grub.conf
```

Searching for Files

```
$ locate e100
```

locate is case sensitive unless you use the –i option

```
$ locate -r /ls$                   Locate files ending in /ls$
/bin/ls
/usr/share/locale/l10n/ls
```

$ locate -r mkfs*3 Locate files with mkfs and 3 in the name
/sbin/mkfs.ext3
/usr/share/man/man8/mkfs.ext3.8.gz

$ locate -r ^/boot/grub/me Locate files beginning with /boot/grub/me
/boot/grub/menu.lst

update the locate database immediately

updatedb

Locating Files with find

$ find / -name “e100*” -print

$ find / -name e100 -print 2>&1 | grep -v “Permission denied”

Or

$ find / -name e100 -print 2> /dev/null

$ find / -name ‘e100*’ -print

$ find /usr/bin/ -amin -2 –print Files accessd in past 2 mins

$ find /home/ilg/ -atime +60 Files not access beyond 60 days

$ find /etc -type d -print 2> /dev/null

$ find /sbin/ -perm 750 -print

$ find /var -user ilg -exec ls -l {} \;

$ find /var -user ilg -print | xargs ls -l

$ find / ! -group root -type f -print 2> /dev/null | xargs ls –l Negating the search

$ find / -xdev -size +10M -print | xargs ls -lS > /tmp/bigfiles.txt

$ whereis man Finds man pages and configuration files associated with a command

$ which ls Where the ls executable is (/bin/ls).

$ rpm –qif `which ps` Looking for the actual location of an executable file in your PATH

Listing Files

$ ls -l	Files and directories in current directory
$ ls -la	Includes files/directories beginning with dot (.**)**
$ ls -lt	Orders files by time recently changed
$ ls –lu	Orders files by time recently accessed
$ ls -lS	Orders files by size
$ ls -li	Lists the inode associated with each file
$ ls -ln	List numeric user/group IDs, instead of names
$ ls -lh	List file sizes in human-readable form (K, M, etc.)

$ ls -lR	List files recursively, from current directory and subdirectories

Chapter 5

Listing, Sorting, and Changing Text

$ cat myfile.txt	Send entire file to the screen
$ cat myfile.txt > copy.txt	Direct file contents to another file
$ cat myfile.txt >> myotherfile.txt	Append file contents to another file
$ cat -s myfile.txt	Display consecutive blank lines as one
$ cat -n myfile.txt	Show line numbers with output
$ cat -b myfile.txt	Show line numbers only on non-blank lines

Cat with head

$ head myfile.txt
$ cat myfile.txt | head

Process head

$ head -n 50 myfile.txt	Show the first 50 lines of a file
**$ ps auwx	head -n 15**	Show the first 15 lines of ps output

Process tail

$ tail -n 15 myfile.txt	Display the last 15 lines in a file
$ tail -15 myfile.txt	Display the last 15 lines in a file
**$ ps auwx	tail -n 15**	Display the last 15 lines of ps output

Log tail

# tail -f /var/log/messages	Watch system messages live
# tail -f /var/log/maillog	Watch mail server messages live
# tail -f /var/log/httpd/access_log	Watch web server messages live

Paging Through Text with more and less

**$ ps auwx	more**	Page through the output of ps (press spacebar)
$ more myfile.txt	Page through the contents of a file
**$ ps auwx	less**	Page through the output of ps
**$ cat myfile.txt	less**	Page through the contents of a file
$ less myfile.txt	Page through a text file

Searching for Text with grep

$ grep ilg myfile.txt	Show lines containing ilg
# grep 404 /var/log/httpd/access_log	Show lines containing 404
$ ps auwx | grep init	Show init lines from ps output
$ ps auwx | grep "\[*\]"	how bracketed commands
$ dmesg | grep "[ ]ata|^ata"	Show ata kernel device information

Recursive search

$ grep -R VirtualHost /etc/httpd/conf*

$ grep -Rn VirtualHost /etc/httpd/conf* With Line nos.

 # grep --color -Rn VirtualHost /etc/httpd/conf*

 # grep -v " 200 " /var/log/httpd/access_log* Display all lines that do not match the string.

Word Counts with wc

$ grep 192.198.1.1 /var/log/httpd/access-log | wc -l

$ wc /etc/httpd/conf.d/README List counts for a single file
9 58 392 /etc/httpd/conf.d/README

$ wc /etc/httpd/conf.d/* List single/totals for many files
20 83 566 /etc/httpd/conf.d/proxy_ajp.conf
9 58 392 /etc/httpd/conf.d/README
11 45 299 /etc/httpd/conf.d/welcome.conf
40 186 1257 total

Output with sort

$ rpm -qa | grep kernel | sort	Sort in alphanumeric order
$ rpm -qa | grep kernel | sort -r	Sort in reverse alphanumeric order

Text in Binaries

$ strings /bin/ls | grep -i libc	Find occurrences of libc in ls
$ cat /bin/ls | strings	List all ASCII text in ls
$ strings /bin/ls	List all ASCII text in ls

Replacing Text with sed

$ cat myfile.txt | sed s/jagjit/ilg/ Replace only the first occurrence per line

$ sed s/jagjit/ilg/g < myfile.txt > mynewfile.txt Change every occurrence and put it into mynewfile.txt

change the delimiter to any other character of your choice

$ sed 's-/var/www/-/home/www/-' < /etc/httpd/conf/httpd.conf
$ sed 'sD/var/www/D/home/www/D' < /etc/httpd/conf/httpd.conf

Run multiple substitutions at once

$ sed -e s/ilg/ILG/g -e s/test/TEST/g < myfile.txt

Translating or Removing Characters with tr

$ ls | tr '\n' ' '	Replace newline characters with spaces
$ tr f F < myfile.txt	Replace every f in the file with F
$ ls | tr -d '\n'	Delete new lines (resulting in one line)
$ tr -d f < myfile.txt	Delete every letter f from the file
$ echo ilg | tr a-z A-Z	Translate ilg into ILG
$ echo ilg | tr '[:lower:]' '[:upper:]'	Translate ilg into ILG

Differences Between Two Files with diff

$ diff /etc/named.conf.rpmnew /etc/named.conf

Exercise :

$ seq 1 7 > f1.txt Send a sequence of numbers to f1.txt
$ cat f1.txt Display contents of f1.txt
1
2
3
4
5
6
7

$ sed s/4/FOUR/ < f1.txt > f2.txt Change 4 to FOUR and send to f2.txt

$ diff f1.txt f2.txt
4c4 Shows line 4 was changed in file
< 4
— -
> FOUR

$ diff -u f1.txt f2.txt Display unified output of diff
— - f1.txt 2007-09-07 18:26:06.000000000 -0500

```
+++ f2.txt 2007-09-07 18:26:39.000000000 -0500
@@ -1,7 +1,7 @@
1
2
3
-4
+FOUR
5
6
7
```

sdiff : ***Merge the output of two files***

```
$ sdiff f1.txt  f2.txt
1                       1
2                       2
3                       3
4                     | FOUR
5                       5
6                       6
7                       7
```

Patch file

```
$ diff -u f1.txt f2.txt > patchfile.txt

$ patch f1.txt < patchfile.txt
patching file f1.txt

$ cat f1.txt
1
2
3
FOUR
5
6
7
```

awk and cut to Process Columns

```
$ ps auwx | awk '{print $1,$11}'          Show columns 1, 11 of ps
$ ps auwx | awk '/ilg/ {print $11}'       Show ilg' processes
$ ps auwx | grep ilg | awk '{print $11}'  Same as above

$ awk -F: '{print $1,$5}' /etc/passwd     Use colon delimiter to print cols

$ cut -d: -f1,5 /etc/passwd               Use colon delimiter to print cols

$ cut -d: -f1-5 /etc/passwd               Show columns 1 through 5, range

$ cut -d: -f5- /etc/passwd                Show columns 5 and later
```

Converting Text Files to Different Formats

Text files in the Unix world use a different end-of-line character (\n) than those used in the DOS/Windows world (\r\n)

$ od –c –t x1 myfile.txt

$ unix2dos < myunixfile.txt > mydosfile.txt
$ cat mydosfile.txt | dos2unix > myunixfile.txt

Page Left blank intentionally

Chapter 6

Multimedia

Playing Music

$ play inconceivable.wav	Play WAV file (may be ripped from CD)
$ play *.wav	Play all WAV files in directory (up to 32)
$ play hi.au vol .6	AU file, lower volume (can lower distortion)
$ play -r 14000 short.aiff	AIFF, sampling rate of 14000 hertz

Ogg format

$ ogg123 mysong.ogg	Play ogg file
$ ogg123 http://vorbis.com/music/Lumme-Badloop.ogg	Play web address
$ ogg123 -z *.ogg	Play files in pseudo-random order
$ ogg123 -Z *.ogg	Same as -z, but repeat forever
$ ogg123 /var/music/	Play songs in /var/music and sub dirs
$ ogg123 -@ myplaylist	Play songs from playlist

Need to install mpg321 package.

$ mpg321 yoursong.mp3	Play MP3 file
$ mpg321 -@ mp3list	Play songs from playlist of MP3s
$ cat mp3list | mpg321 -@ -	Pipe playlist to mpg321
$ mpg321 -z *.mp3	Play files in pseudo-random order
$ mpg321 -Z *.mp3	Same as -z, but repeat forever

Adjusting Audio Levels

use alsamixer to adjust sound when ALSA is used and aumix with OSS.

$ alsamixer	Show alsamixer screen with playback view
$ alsamixer -V playback	Show only playback channels (default)
$ alsamixer -V all	Show with playback and capture views
$ **alsamixer -c 1**	Use alsamixer on second (1) sound card
$ aumix -q	Show left/right volume and type for all channels
$ aumix -l q -m q	List current settings for line and mic only
$ aumix -v 80 -m 0	Set volume to 70% and microphone to 0
$ aumix -m 80 -m R -m q	Set mic to 80%, set it to record, list mic
$ aumix	With no options, aumix runs screen-oriented

Encoding Music

oggenc, you can start with audio files or streams in WAV, AIFF, FLAC, or raw format and convert them to Ogg Vorbis format.

$ oggenc mysong.wav	Encodes WAV to Ogg (mysong.ogg)
$ **oggenc ab.flac -o new.ogg**	Encodes FLAC to Ogg (new.ogg)
$ oggenc ab.wav -q 9	Raises encoding quality to 9

set the quality

**$ oggenc NewSong.wav -o NewSong.ogg **
**-a Bernstein -G Classical **
**-d 06/15/1972 -t “Simple Song” **
**-l “Bernsteins Mass” **
-c info=”From Kennedy Center”

Get information on Newsong.ogg

$ ogginfo NewSong.ogg

Another tool to encode

$ flac now.wav	Encodes WAV to FLAC (now.flac)
$ sox now.wav now.aiff	Encodes WAV to AIFF (now.aiff)
$ flac now.aiff -o now2.flac	Encodes AIFF to FLAC (now.flac)
$ flac -8 top.wav -o top.flac	Raises compression level to 8

convert files to MP3 format using the lame

$ lame in.wav	Encodes WAV to MP3 (in.wav.mp3)
$ lame in.wav –-preset standard	Encodes to MP3 with std presets
$ lame tune.aiff -o tune.mp3	Encodes AIFF to MP3 (tune.mp3)
$ lame -h -b 64 -m m in.wav out.mp3	High quality, 64-bit, mono mode
$ lame -q 0 in.wav -o abcHQ.mp3	Encodes with quality set to 0

Streaming Music

If your music is on one machine, but you’re working from another machine, setting up a streaming music server is a quick way to broadcast your music so it can be picked up from one or more computers on your network. The icecast streaming media server and ices audio source client can be installed in Fedora by typing:

 # yum install icecast ices

Here’s a quick and dirty procedure for setting up icecast and ices to stream your music. Perform this task on the computer that contains the music you want to serve:

1. Edit the /etc/icecast.xml file to change all passwords listed. Search for hackme to find the current passwords. You probably want different user and administrative passwords, especially if you allow others to stream music to the server. Remember the passwords you set for later. You may want to change other settings in this file as well, such as hostname.

2. If you have a firewall, check that TCP port 8000 is accessible.

3. Start the icecast server as root user by typing the following (the server will actually run as the icecast user):

 # service icecast start

4. Use the ices user account to create your playlist. The ices user account is created

when you install the ices package, However, you need to modify the account to be able to log in as the ices user and save files to that user's home directory. As root user, type the following:

usermod -m -d /home/ices -s /bin/bash
passwd ices
Changing password for user ices.
New UNIX password: ********

5. Log in as the ices user.

6. Create a playlist using any text edit or by directing a listing of your music to a file. For example, if all your Ogg music files are in /var/music subdirectories, type the following:

$ **find /var/music -name *.ogg > /home/ices/playlist.txt**

With the playlist file created, use any text editor to remove or add files or directories to make your playlist as you would like it. (If you want some files to try out for your playlist, download some from http://vorbis.com/music.)

7. As root user, edit the /etc/ices.conf file so it will play from your playlist and feed that music to your running icecast server. In particular, you want to modify the metadata, input, and instance modules. (Be sure to change /home/foo/ playlist.txt to the path where you put your playlist.txt file.)

<metadata>
<name>**My Music Server**</name>
<genre>**Different music styles**</genre>
<description>**Mix of my personal music**</description>
</metadata>
<input>
<module>**playlist**</module>
<param name="type">**basic**</param>
<param name="file">**/home/ices/playlist.txt**</param>
<! — random play — >
<param name="random">**1**</param>
...
</input>
<instance>
<hostname>**localhost**</hostname>
<port>**8000**</port>
<password>**MIcePw**</password>
<mount>/**mymusic.ogg**</mount>
...
</instance>
Of the values just shown (in bold), the most critical are the location of your playlist and the information about the instance of your icecast server. The password must match the source password you added to your /etc/icecast.xml file.

8. Launch the ices audio feed by typing the following:
service ices start

9. Test that you can play music from the local computer as follows:
$ **ogg123 http://localhost:8000/mymusic.ogg**

10. If that test works, try playing the icecast stream from another computer on your network by replacing localhost with the server's IP address or host name.

11. If there are problems, check /var/log/icecast and /var/log/ices log files. Recheck your passwords and locations of configuration files.

12. When you are done, just kill the ices and icecast services:

 # service ices stop
 # service icecast stop

When the icecast and ices servers are running, you should have access to that streaming music from any computer that can access your server computer. Use any music player that can play from an HTTP address (ogg123, Rhythmbox, XMMS, and so on). Windows music players that can support the type of content you are serving should work as well.
NOTE *If you want to skip a song, type this from the server:* killall -HUP ices.

Converting Audio Files

concatenates two WAV files to a single output file:

$ sox head.wav tail.wav output.wav

Mixing two WAV files:

$ soxmix sound1.wav sound2.wav output.wav

display information about a file,

$ sox sound1.wav -e stat
Samples read: 208512
Length (seconds): 9.456327
Scaled by: 2147483647.0
Maximum amplitude: 0.200592
Minimum amplitude: -0.224701
Midline amplitude: -0.012054
Mean norm: 0.030373
Mean amplitude: 0.000054
RMS amplitude: 0.040391
Maximum delta: 0.060852
Minimum delta: 0.000000
Mean delta: 0.006643
RMS delta: 0.009028
Rough frequency: 784
Volume adjustment: 4.450

Delete seconds of sound

$ sox sound1.wav output.wav trim 4 Trim 4 seconds from start
$ sox sound1.wav output.wav trim 2 6 Keep from 2-6 seconds of file

Transforming Images

Information about Images

$ identify p2090142.jpg

p2090142.jpg JPEG 2048x1536+0+0 DirectClass 8-bit 402.037kb

$ identify -verbose p2090142.jpg | less
Standard deviation: 61.1665 (0.239869)
Colors: 205713
Rendering intent: Undefined
Resolution: 72x72
Units: PixelsPerInch
Filesize: 402.037kb
Interlace: None
Background color: white
Border color: rgb(223,223,223)
Matte color: grey74
Transparent color: black
Page geometry: 2048x1536+0+0
Compression: JPEG
Quality: 44

Converting Images

$ convert tree.jpg tree.png Convert a JPEG to a PNG file
$ convert icon.gif icon.bmp Convert a GIF to a BMP file
$ convert photo.tiff photo.pcx Convert a TIFF to a PCX file

$ convert -resize 1024x768 hat.jpg hat-sm.jpg
$ convert -sample 50%x50% dog.jpg dog-half.jpg

$ convert -rotate 270 sky.jpg sky-final.jpg Rotate image 270 degrees
$ convert -rotate 90 house.jpg house-final.jpg Rotate image 90 degrees

Add text to an image

**$ convert -fill black -pointsize 60 -font helvetica **
**-draw 'text 10,80 "Copyright NegusNet Inc."' **
p10.jpg p10-cp.jpg

create thumbnails

$ convert -thumbnail 120x120 a.jpg a-a.png
$ convert -thumbnail 120x120 -polaroid 8 a.jpg a-b.png
$ convert -thumbnail 120x120 -polaroid 8 -rotate 8 a.jpg a-c.png

making your images fun and even weird.

$ convert -sepia-tone 75% house.jpg oldhouse.png
$ convert -charcoal 5 house.jpg char-house.png
$ convert -colorize 175 house.jpg color-house.png

Funny...weird

$ convert -swirl 300 photo.pcx weird.pcx

$$$

Chapter 7

Administering File Systems

Creating and Managing File Systems

List information about the partitions on your hard disk,

 # fdisk -l — List disk partitions for every disk

```
Disk /dev/sda: 82.3 GB, 82348277760 bytes
255 heads, 63 sectors/track, 10011 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Device     Boot   Start     End      Blocks      Id     System
/dev/sda1     *       1        13      104391      83     Linux
/dev/sda2             14      9881    79264710    83     Linux
/dev/sda3 9882 10011 1044225 82 Linux swap
```

 # fdisk /dev/sda — Start interactive fdisk session with disk 1

Command (m for help): m — Type m to list help text as shown

```
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):
```

Command (m for help): d — Ask to delete a partition
Partition number (1-4): 4 — Type partition number to delete

Command (m for help): n — Create a new disk partition
First cylinder (1-4983, default 1): 1 — Select start (or Enter)
Last cylinder ... (default 4983): 4983 — Select end (or Enter)

Command (m for help): a — Make a partition bootable
Partition number (1-3): 1 — Type bootable partition number

Command (m for help): t — Select a file system type
Partition number (1-3): 3 — Select partition to change
Hex code (type L to list codes): 82 — Assign partition as swap

Copying Partition Tables with sfdisk

Back up or replicate a disk's partition table

 # sfdisk –d /dev/sda > sda-table Back up partition table to file
 # sfdisk /dev/sda < sda-table Restore partition table from file
 # sfdisk –d /dev/sda | sfdisk /dev/sdb Copy partition table from disk to disk

Changing Disk Partitions with parted

List partitions with parted

```
# parted -l
Model: ATA FUJITSU MPG3409A (scsi)
Disk /dev/sda: 41.0GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Number Start End Size Type File system Flags
1 32.3kB 206MB 206MB primary ext3 boot
2 206MB 39.5GB 39.3GB primary ext3
3 39.5GB 41.0GB 1536MB primary linux-swap
```

Run parted interactively

```
# parted
GNU Parted 1.8.6
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

WARNING! Unlike fdisk, parted immediately incorporates changes you make to your partitions, without explicitly writing the changes to disk. So don't just assume you can back out of any changes by simply quitting parted.

#(parted) mkpart Create a new partition

```
Partition type? [logical]? primary
File system type? [ext2]? ext3
Start? 17GB
End? 24GB
```

#(parted) resize 2 Resize a partition

```
Start? [1.2GB] 1.2GB
End? [24GB] 10GB
```

File System Labels

See a partition's label

 # e2label /dev/sda2

 # e2label /dev/sda2 mypartition set the label on a partition

# findfs LABEL=mypartition	Find a partition when you know only the label

Exercise : Please find some information on uuidgen command for generating uniquer uuid for partitions

Formatting a File System

mkfs command to create file systems

# mkfs -t ext3 /dev/sdb1	Create ext3 file system on sba1
# mkfs -t ext3 -v -c /dev/sdb1	More verbose and scan for bad blocks
# mkfs.ext4 /dev/sdaX	Formatting with mkfs.ext4 command
# mkfs.ext3 -c /dev/sdb1	Same result as previous command

To add a partition label to the new partition, use the -L option

# mkfs.ext3 -c -L mypartition /dev/sdb1	Add mypartition label
# mkfs.ext4 -U <UUID number> /dev/sdXY	Add uuid for partition

Creating a Virtual File System

$ dd if=/dev/zero of=mydisk count=2048000	Create zero-filled 1GB file
$ du -sh mydisk 1001M mydisk	Check virtual file system size
$ mkfs -t ext3 mydisk mydisk is not a block special device Continue (y/n): y	Create files system on mydisk
$ mkdir /mnt/image	Create a mount point
# mount -o loop mydisk /mnt/image	Mount mydisk on /mnt/image

Using Virtual File system

# cd /mnt/image	Change to the mount point
# mkdir test	Create a directory on the file system
# cp /etc/hosts .	Copy a file to the file system
# cd	Leave the file system
# umount /mnt/image	Unmount the file system

Viewing and Changing File System Attributes

# tune2fs -l /dev/sda1	View tunable file system attributes
# dumpe2fs -h /dev/sda1	Same as tune2fs output

change settings on an existing ext2 or ext3 file system,

# tune2fs -c 31 /dev/sda1	Sets # of mounts before check is forced rather than number of mounts

Forced file system checks based on time interval

 # tune2fs -c -1 /dev/sda1

-i option to enable time-dependent checking.

# tune2fs -i 10 /dev/sda1	Check after 10 days
# tune2fs -i 1d /dev/sda1	Check after 1 day
# tune2fs -i 3w /dev/sda1	Check after 3 weeks
# tune2fs -i 6m /dev/sda1	Check after 6 months
# tune2fs -i 0 /dev/sda1	Disable time-dependent checking

-j option to turn an ext2 file system into ext3

# tune2fs -j /dev/sda1	Add journaling to change ext2 to

Creating and Using Swap Partitions

# mkswap /dev/sda1	Format sda1 as a swap partition

check your swap area for bad blocks, use the -c option to mkswap

 # mkswap -c /dev/sda1

create a swap area within a file

 # dd if=/dev/zero of=/tmp/swapfile count=65536
65536+0 records in
65536+0 records out
33554432 bytes (34 MB) copied, 1.56578 s, 21.4 MB/s

 # chmod 600 /tmp/swapfile

 # mkswap /tmp/swapfile
Setting up swapspace version 1, size = 67104 kB

Turn on the swap partition after you have created it.

# swapon /dev/sda1	Turn swap on for /dev/sda1 partition
# swapon -v /dev/sda1	Increase verbosity as swap is turned on
swapon on /dev/sda1	
# swapon -v /tmp/swapfile	Turn swap on for the /tmp/swapfile file
swapon on /tmp/swapfile	

see a list of your swap files and partitions

# swapon -s	View all swap files and partitions that are on
Filename Type Size Used Priority	

/dev/sda5 partition 1020088 142764 -1
/tmp/swapfile file 65528 0 -6

Turn off a swap area

 # swapoff -v /tmp/swapfile
swapoff on /tmp/swapfile

Specify the priority of your swap area

 # swapon -v -p 1 /dev/sda1 Assign top swap priority to sda1

Mounting File Systems

$ mount List mounted remote and local file systems

/dev/sda7 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sda6 on /mnt/debian type ext3 (rw)
/dev/sda3 on /mnt/slackware type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)

$ mount -t ext3 List mounted ext3 file systems
/dev/sda7 on / type ext3 (rw)
/dev/sda6 on /mnt/debian type ext3 (rw)
/dev/sda3 on /mnt/slackware type ext3 (rw)

display partition labels with mount information

$ mount -t ext3 -l List mounted ext3 file systems and labels

mount command to mount the /dev/sda1 device on an existing directory named /mnt/mymount

 # mount /dev/sda1 /mnt/mymount/ Mount a local file system

 # mount -v /dev/sda1 /mnt/mymount/ Mount file system, more verbose

 # mount -v -t ext3 /dev/sda1 /mnt/mymount/ Mount an ext3 file system

 # mount -vl -t ext3 /dev/sda1 /mnt/mymount/ Mount file system/show label

 # mount -v -t ext3 -o rw /dev/sda1 /mnt/mymount/ Mount read/write

 # mount -v -t ext3 -o ro /dev/sda1 /mnt/mymount/ Mount read-only

Remount

 # mount -v -o remount,ro /dev/sda1

Bind

 # mount --bind -v /mnt/mymount/ /tmp/mydir/

Move

 # mount -v --move /mnt/mymount/ /tmp/mydir/

Unmounting File Systems with umount

#umount -v /dev/sda1 — Unmount by device name
/dev/sda1 umounted

 # umount -v /mnt/mymount/ — Unmount by mount point
/tmp/diskboot.img umounted

Checking File Systems

 # badblocks /dev/sda1 — Physically scan hard disk for bad blocks
 # badblocks -v /dev/sda1 — Add verbosity to hard disk scan

 # badblocks -vsn /dev/sda1 — Check bad blocks, non-destructive

 # badblocks -vsw /dev/sda1 — Check bad blocks, destructive

Fsck

 # fsck /dev/sda1

 # fsck -TV /dev/sda1 — Check file system (verbose and no version)

 # fsck -TV /dev/sda1 — Prompting to correct problems encountered

 # fsck -TVy /dev/sda1 — Consider yes

Checking RAID Disks

 # mdadm -Q /dev/md1

 # mdadm -Q --detail /dev/md1

File System Use

$ df –h Display space on file systems in human-readable form

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda	**2 7.6G**	**3.4G**	**3.9G**	**47%**	**/**
/dev/sda1	**99M**	**14M**	**80M**	**15%**	**/boot**
tmpfs	**501M**	**0**	**501M**	**0%**	**/dev/shm**
/dev/sda5	**352G**	**197G**	**137G**	**59%**	**/home**

Check inode utilization

$ df -hi

Filesystem	Inodes	IUsed	IFree	IUse%	Mounted on
/dev/sda2	2.0M		108K	1.9M 6%	/

$ df –hl Display disk space only for local file systems

$ df -hT Add file system type information to listing

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
/dev/sda7	ext3	8.8G	5.5G	2.9G	66%	/

check for disk space usage for particular files or directories in a file system

$ du -h /home/ Show disk space usage for /home directory

```
du: `/home/ilg': Permission denied
4.0K /home/ilg/Mail
52K /home/ilg
64K /home/
```

 # du -sh /home You can display summary disk use as root user

```
1.6G /home
```

 # du -sch /home /var Show directory and total summaries

```
1.6G /home
111M /var
1.7G total
```

 # du -sh --exclude='*.iso' /home/ilg Exclude ISO images from totals

 # du -h --max-depth=1 /home Provide disk space use, to one level deep

 # du -h --max-depth=2 /home Dig two-levels deep for disk space use

Chapter 8

Backups and Removable Media

Create the archive and compress the output

$ tar zcf myfiles.tar.gz *.txt Create gzipped tar file of .txt files
$ tar zcvf myfiles.tar.gz *.txt Be more verbose creating archive

To extract

$ tar xzvf myfiles.tar.gz

The tar command with bzip2 compression,

$ tar cjvf myfiles.tar.bz2 *.txt Create archive, compress with bzip2

$ tar xjvf myfiles.tar.bz2 Extract files, uncompress bzip2 compression

Compressing with gzip

$ gzip myfile gzips myfile and renames it myfile.gz

$ gzip -v myfile gzips myfile with verbose output
myfile: 86.0% -- replaced with myfile.gz

$ gzip -tv myfile.gz Tests integrity of gzip file
myfile.gz: OK

$ gzip -lv myfile.gz Get detailed info about gzip file
method crc date time compressed uncompressed ratio uncompressed_name
defla 0f27d9e4 Jul 10 04:48 46785 334045 86.0% myfile

Compress all files in a directory

$ gzip -rv mydir Compress all files in a directory
mydir/file1: 39.1% -- replaced with mydir/file1.gz
mydir/file2: 39.5% -- replaced with mydir/file2.gz

$ gzip -1 myfile Fastest compression time, least compression
$ gzip -9 myfile Slowest compression time, most c ompression

Uncompress a gzipped file,

$ gunzip -v myfile.gz Unzips myfile.gz and renames it myfile
myfile.gz: 86.0% -- replaced with myfile

$ gzip -dv myfile.gz Same as previous command line

Compressing with bzip2

$ bzip2 myfile Compresses file and renames it myfile.bz2
$ bzip2 -v myfile Same as previous command, but more verbose
myfile: 9.529:1, 0.840 bits/byte, 89.51% saved, 334045 in, 35056 out.
$ bunzip2 myfile.bz2 Uncompresses's file and renames it myfile
$ bzip2 -d myfile.bz2 Same as previous command
$ bunzip2 -v myfile.bz2 Same as previous command, but more verbose myfile.bz2: done

Listing, Joining, and Adding Files to tar Archives

List an archive's contents

$ **tar tvf myfiles.tar** List files from uncompressed archive
-rw-r--r-- root/root 9584 2007-07-05 11:20:33 textfile1.txt
-rw-r--r-- root/root 9584 2007-07-09 10:23:44 textfile2.txt

$ tar tzvf myfiles.tgz List files from gzip compressed archive

Concatenate one tar file to another

$ tar -Af archive1.tar archive2.tar

To add one or more files to an existing archive.

$ tar rvf archive.tar myfile Add a file to a tar archive

To match multiple files to add

$ tar rvf archive.tar *.txt Add multiple files to a tar archive

Deleting Files from tar Archives

Deleting files from a tar archive

$ tar --delete file1.txt -f myfile.tar Delete file1.txt from myfile.tar

Backing Up tar Archives Over ssh

Gather backups from multiple client machines

$ mkdir mybackup ; cd mybackup

$ ssh ilg@server1 'tar cf - myfile*' | tar xvf -
ilg@server1's password: ******
myfile1
myfile2

To reverse the process and copy files from the local system to the remote system,

$ tar cf - myfile* | ssh ilg@server1 'cd /home/ilg/myfolder; tar xvf -'
igl@server1's password: ******
myfile1
myfile2

Exercise : What the below commands does ?.

$ ssh ilg@server1 'tar czf - myfile*' | cat > myfiles.tgz
$ tar cvzf - myfile* | ssh ilg@server1 'cat > myfiles.tgz'

Backing Up Files with rsync

$ rsync -avz --delete ilg@server1:/home/ilg/pics/ ilgpics/

to mirror the remote directory structure (/home/ilg/pics/) on the local system. The -a says to run in archive mode (recursively copying all files from the remote directory), the -z option compresses the files, and –v makes the output verbose. The --delete tells rsync to delete any files on the local system that no longer exist on the remote system.

Ongoing backups, you can have rsync do seven-day incremental backups , with cron scheduling.

 # mkdir /var/backups

**# rsync --delete --backup **
**--backup-dir=/var/backups/backup-`date +%A` **
**-avz ilg@server1:/home/ilg/Personal/ **
/var/backups/current-backup/

Backing Up to Removable Media

To create an ISO image from all files and directories under the /home directory

 # cd /tmp
 # mkisofs -o home.iso /home — Create basic ISO9660 image
 # mkisofs -o home2.iso -J -R /home — Add Joliet Rock Ridge extensions
 # mkisofs -o home3.iso -J -R -hfs /home — Also add HFS extensions

Multiple sources added to the image.

 # mkisofs -o home.iso -R -J music/ docs/test.pdf /var/spool/mail
Multiple Directories/Files
 # mkisofs -o home.iso -J -R -graft-points Pictures/=/var/pics/home/ilg
Graft files on to the image

Adding information into the heade of the ISO image

 # mkisofs -o /tmp/home.iso -R -J -p www.handsonhistory.com -publisher "Swan Bay Folk Art Center" -V "WebBackup" -A "mkisofs" -volset "1 of 4 backups, July 30, 2007" /home/ilg

Check the image

 # volname home.iso — Display volume name

WebBackup

# isoinfo -d -i home.iso	Display header information

```
CD-ROM is in ISO 9660 format
System id: LINUX
Volume id: WebBackup
Volume set id: All Website material on November 2, 2007
Publisher id: Swan Bay Folk Art Center
Data preparer id: www.handsonhistory.com
Application id: mkisofs
Copyright File id:
Abstract File id:
Bibliographic File id:
Volume set size is: 1
Volume set sequence number is: 1
Logical block size is: 2048
Volume size is: 23805
Joliet with UCS level 3 found
Rock Ridge signatures version 1 found
```

Accessing files on the ISO image by mounting

# mkdir /mnt/myimage	Create a mount point
# mount -o loop home.iso /mnt/myimage	Mount the ISO in loopback
# ls -l /mnt/myimage	Check the ISO contents
# umount /mnt/myimage	Unmount the image when done

Backup Images with cdrecord

check that your drive supports CD/DVD burning and determine the address of the drive.

# cdrecord --scanbus	Shows a drive that can burn CDs or DVDs

```
scsibus0:
0,0,0 0) 'LITE-ON ' 'DVDRW SOHW-1633S' 'BS0C' Removable CD-ROM
0,0,0 1) *
0,0,0 2) *
```

Burning a CD or DVD images:

# cdrecord -dummy home.iso	Test burn without actually burning
# cdrecord -v home.iso	Burn CD (default settings) in verbose
# cdrecord -v speed=24 home.iso	Set specific speed
# cdrecord -pad home.iso	Can't read track so add 15 zeroed sectors
# cdrecord -eject home.iso	Eject CD/DVD when burn is done
# cdrecord /dev/cdrw home.iso	dentify drive by device name (may differ)
# **cdrecord dev=0,2,0 home.iso**	Identify drive by SCSI name

Burn multi-session CDs/DVDs.

# cdrecord -multi home.iso	Start a multi-burn session
# cdrecord -msinfo	Check the session offset for next burn Using /dev/cdrom of unknown capabilities 0,93041
# mkisofs -J -R -o new.iso -C 0,93041 /home/ilg/more	Create a second ISO to burn

# cdrecord new.iso	Burn new data to existing CD

Making and Burning DVDs with 'growisofs'

Combine the two steps of gathering files into an ISO image (mkisofs) and burning that image to DVD (cdrecord). Besides saving a step, the growisofs command also offers the advantage of keeping a session open by default until you close it, so you don't need to do anything special for multi-burn sessions.

# growisofs -Z /dev/dvd -R -J /home/ilg	Master and burn to DVD
# growisofs -Z /dev/dvd -R -J /home/ilg	Add to burn
# growisofs -M /dev/dvd=/dev/zero	Close burn

# growisofs -dvd-compat -Z /dev/dvd=image.iso	Burn an ISO image to DVD

The -dvd-compat option can improve compatibility with different DVD drives over some multi-session DVD burning procedures

Chapter 9

Checking and Managing Running Processes

Active Processes

$ ps	List processes of current user at current shell
$ ps -u ilg	Show all ilg's' running processes (simple output)
$ ps -u ilg u	Show all ilg's' running processes (with CPU/MEM)
$ ps -fu ilg	Show all ilg running processes (with PPID)
$ ps -Fu ilg	Show all Ilg running processes (with SZ and PSR)
$ ps -e	Show every running process
$ ps -el	Show every running process, long listing
$ ps -ef	**Show every running process, full-format listing**
$ ps -eF	Show every running process, extra full-format listing
$ ps ax	Show every running process, short BSD style
$ ps aux	Show every running process, long BSD style
$ ps auwx	Show every running process, long BSD style, wide format
$ ps auwwx	Show every running process, long BSD style, unlimited width
$ ps -ejH	Show process hierarchy with process/session IDs
$ ps axjf	Show process hierarchy in BSD-style output
$ ps -ef --forest	Show process hierarchy in forest format
$ pstree	Show processes alphabetically in tree format

Custom views of running processes

$ ps -eo ppid,user,%mem,size,vsize,comm --sort=-size Sort by mem use

```
PPID    USER   %MEM SZ      VSZ     COMMAND
1       root    27.0      68176   84264   yum-updatesd
```

$ ps -eo ppid,user,bsdstart,bsdtime,%cpu,args --sort=-%cpu Sort by CPU use

```
PPID    USER   START TIME    %CPU  COMMAND
1       root Jul 30       44:20      27.1          /usr/bin/python /usr/sbin/yum-updatesd
```

$ ps -eo ppid,user,nice,cputime,args --sort=-nice Sort by low priority

```
PPID    USER   NI        TIME              COMMAND
1       root    19       00:44:26          /usr/bin/python /usr/sbin/yum-updatesd
```

$ ps -eo ppid,user,stat,tname,sess,cputime,args --sort=user Sort by user

```
PPID    USER    STAT    TTY     SESS              TIME            COMMAND
1       avahi   Ss ?            2221            00:00:07        avahi-daemon: running
[example.net]
```

$ ps -C httpd Display running httpd processes

```
PID TTY TIME CMD
1493 ? 00:00:00 httpd
1495 ? 00:00:00 httpd
```

$ ps -p 5413 -o pid,ppid,bsdtime,args Display info for PID 5413

```
PID     PPID            TIME            COMMAND
5413    1               0:08            gpm -m /dev/input/mice -t exps2
```

$ ps -U ilg, jagjit -o pid,ruser,tty,stat,args See info for 2 users

```
PID             RUSER           TT      STAT            COMMAND
1010            ilg             pts/0   Ss              -bash
5951            jagjit          pts/1   Ss+             /bin/bash
```

Watching Active Processes with top

$ top

```
top - 10:21:08 up 7 days, 49 min,  3 users,  load average: 0.55, 0.49, 0.40
Tasks: 231 total,   1 running, 230 sleeping,   0 stopped,   0 zombie
%Cpu(s): 20.7 us,  4.8 sy,  0.0 ni, 73.7 id,  0.7 wa,  0.0 hi,  0.1 si,  0.0 st
KiB Mem:   6013196 total,  3646852 used,  2366344 free,   143772 buffers
KiB Swap:  4878900 total,    17400 used,  4861500 free,  1616212 cached

  PID USER      PR  NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
 1440 root      20   0  583m 195m 164m S  37.5  3.3 172:07.55 Xorg
18126 root      20   0  167m  23m  15m S  31.3  0.4  39:33.83 gca
 8877 ilg       20   0 1498m 662m  49m S   6.3 11.3   2:07.50 firefox
 9911 ilg       20   0 20644 1400 1020 R   6.3  0.0   0:00.02 top
18120 root      20   0  640m  25m  15m S   6.3  0.4  36:01.71 gca
    1 root      20   0 27096 2656 1400 S   0.0  0.0   0:04.42 init
    2 root      20   0     0    0    0 S   0.0  0.0   0:00.06 kthreadd
    3 root      20   0     0    0    0 S   0.0  0.0   0:32.67 ksoftirqd/0
    5 root       0 -20     0    0    0 S   0.0  0.0   0:00.00 kworker/0:0H
```

$ top -u ilg Only see processes of effective user name ilg
$ top -p 190,2690 Only display processes 190 and 2690
$ top -b Run in non-interative non-screen-oriented mode

Finding and Controlling Processes

$ pgrep init Show PID for any process including ‘init’ string
1
2689

$ pgrep -l init Show PID and name for any process including ‘init’ string
1 init
2689 xinit

$ pgrep -lu ilg List all processes owned by user ilg
2551 sshd
2552 bash
2803 vim

$ ps -p `pgrep firefox` Search for metacity and run ps (short)
PID TTY TIME CMD
2778 ? 00:05:00 metacity

$ ps -fp $(pgrep xinit) Search for xinit and run ps (full)

UID	PID	PPID	C	STIME	TTY	TIME	CMD
ilg	2689	26	0	Aug14	tty1	00:00:00	xinit /etc/X11/xinit/xinitrc

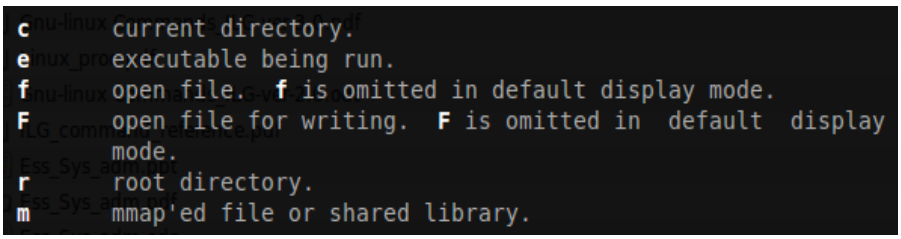
 # renice -5 $(pgrep firefox) Search for firefox, improve its priority
20522: old priority 0, new priority -5
20557: old priority 0, new priority -5

fuser to Find Processes

listing processes that have files open on a selected file system

 # fuser -mauv /boot Verbose output of processes with /boot open

	USER	PID	ACCESS	COMMAND
/boot/grub/:	root	3853	..c..	(root)bash
	root	19760	..c..	(root)bash
	root	28171	F.c..	(root)vi
	root	29252	..c..	(root)man
	root	29255	..c..	(root)sh
	root	29396	F.c..	(root)vi



show processes with files open:

 # fuser /boot Show parent PIDs for processes opening /boot

 # fuser -m /boot Show all PIDs for processes opening /boot

 # fuser -u /boot Show PIDs/user for this shell open in /boot

kill or send other signals to all processes with files open to a file system:

 # fuser -k /boot Kill all processes with /boot files open (SIGKILL)
fuser -l List supported signals
HUP INT QUIT ILL TRAP ABRT IOT BUS FPE KILL USR1 SEGV USR2 PIPE ALRM TERM STKFLT CHLD CONT STOP TSTP TTIN TTOU URG XCPU XFSZ VTALRM PROF WINCH IO PWR SYS UNUSED

 # fuser -k -HUP /boot Send HUP signal to all processes with /boot open

Changing Running Processes

Adjusting Processor Priority with nice

see your current nice value,
$ nice Run nice to determine current niceness
0

Priority number can range from
–20 (most favorable scheduling priority)
+19 (least favorable scheduling priority).

Change a command's nice value
 # nice -n -10 xclock & Launch xclock in the background with high priority

 # nice -n -10 xlogo Launch xlogo at higher priority

Change the process's nice value using the renice command, when a process is already executing.
$ renice +2 -u ilg Renice ilg' processes +2

$ renice +5 4737 Renice PID 4737 by +5

Running Processes in the Background and Foreground

Stop and start the process and move it between foreground and background
$ xlogo Run xlogo in the foreground
<Ctrl+z> Stop process and place in background
[1]+ Stopped xlogo

$ bg %1 Start process running again in background
$ fg %1 Continue running process in foreground
xlogo
<ctrl+c> **"ctrl+c"** will Kill / terminate process

$ jobs Display background jobs for current shell
[1] Running gimp &
[2] Running xmms &
[3]- Running gedit &
[4]+ Stopped gtali use top -p <pid> to view process in new terminal

$ jobs -l Display PID with each job's information
[1] 31676 Running gimp &
[2] 31677 Running xmms &
[3]- 31683 Running gedit &
[4]+ 31688 Stopped gtali
$ jobs -l %2 Display information only for job %2
[2] 31677 Running xmms &

You can disconnect the process from the current shell using the disown command

$ disown %3	Disconnect job %3 from current shell
$ disown -a	Disconnect all jobs from current shell
$ disown -h	Protect all jobs from HUP sent to current shell

After you have disowned a process, you can close the shell without also killing the process.

NOTE With fg, bg, or disown, if you don’t indicate which process to act on, the current job is used.

The current job has a plus sign (+) next to it.

Killing and Signaling Processes

Signal Number	Signal Name	Description
1	SIGHUP	Hang up from terminal or controlling process died
2	SIGINT	Keyboard interrupt
3	SIGQUIT	Keyboard quit
4	SIGILL	Illegal instruction
6	SIGABRT	Abort sent from abort function
8	SIGFPE	Floating point exception
9	SIGKILL	Kill signal
11	SIGSEGV	Invalid memory reference
13	SIGPIPE	Pipe broken (no process reading from pipe)
14	SIGALRM	Timer signal from alarm system call
15	SIGTERM	Termination signal
30,10,16	SIGUSR1	User-defined signal 1
31,12,17	SIGUSR2	User-defined signal 2
20,17,18	SIGCHLD	Child terminated or stopped
19,18,25	SIGCONT	Continue if process is stopped
17,19,23	SIGSTOP	Stop the process
18,20.24	SIGTSTP	Stop typed at terminal
21,21,26	SIGTTIN	Terminal input for background process
22,22,27	SIGTTOU	Terminal output for background process

$ kill 28665	Send SIGTERM to process with PID 28665
$ kill -9 4895	Send SIGKILL to process with PID 4895
$ kill -SIGCONT 5254	Continue a stopped process (pid 5254)
$ kill %3	Kill the process represented by job %3
$ killall spamd	Kill all spamd daemons currently running
$ killall -SIGHUP sendmail	Have sendmail processes reread config files

Run a process in a way that it is impervious to a hang-up signal i.e process to continue to run, even if you disconnect from the current shell

$ nohup updatedb & Run updatedb with no ability to interrupt
 # nohup nice -9 gcc hello.c & Run gcc uninterrupted and higher priority

Scheduling Processes to Run

'at' command runs a command at the time you set

$ at now +1 min Start command running in one minute
at> updatedb
at> <Ctrl+d> <EOT>
job 5 at Mon Aug 20 20:37:00 2007
$ at teatime Start command at 4pm today
$ at now +5 days Start a command in five days
$ at 06/25/08 Start a command at current time on June 25, 2008

Set a command to start as soon as the processor is ready

$ batch Start command running immediately
at> **find /mnt/isos | grep jpg$ > /tmp/mypics**
at> <Ctrl+d> <EOT>

Check the queue of at jobs that are set

$ atq
11 Wed Sep 5 21:10:00 2007 a ilg
10 Fri Aug 24 21:10:00 2007 a ilg
8 Thu Aug 23 20:53:00 2007 a ilg

Delete an at job from the queue

$ atrm 11 Delete at job number 11

Crontab

Create a personal crontab file,

$ crontab -e Create a personal crontab file

15 8 * * Mon,Tue,Wed,Thu,Fri mail ilg < /var/project/stats.txt
* * 1 1,4,7,10 * find / | grep .doc$ > /var/sales/documents.txt

Fields :
minute (0 to 59)
hour (0 to 23)
day of the month (0 to 31)
month 0 to 12 or Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, or Dec
Day of the week 0 to 7 or Sun, Mon, Tue, Wed, Thu, Fri, or Sat
An asterisk (*) in a field means to match any value for that field.

 # crontab -eu ilg Edit another user's crontab (root only)

$ crontab -l List contents of your crontab file

15 8 * * Mon,Tue,Wed,Thu,Fri mail ilg < /var/project/stats.txt
* * 1 1,4,7,10 * find / | grep .doc$ > /var/sales/documents.txt

$ crontab -r Delete your crontab file

$$

Chapter 10

Managing the System

Monitoring Memory Use

see how much memory is being used

$ free **List memory usage in kilobytes (-k default)**

```
              total       used       free     shared    buffers     cached
Mem:         742476     725108      17368          0     153388     342544
-/+ buffers/cache:      229176     513300
Swap:       1020116         72    1020044
```

$ free –m **List memory usage in megabytes**

```
             Total  used  free shared buffers cached
Mem:           725   706    18     0      48    333
-/+ buffers/cache:   223   501
Swap:          996     0   996
```

$ free -b **List memory usage in blocks**

```
            total       used      free   shared    buffers     cached
Mem: 760295424  742510592 1778483     0     157114368   350765056
-/+ buffers/cache: 234631168 525664256
Swap: 1044598784   73728  1044525056
```

$ free -mt **List memory usage with totals displayed (Swap + Mem)**

```
             total   used   free   shared  buffers  cached
Mem:           725    708     16        0      149     334
-/+ buffers/cache:    223    501
Swap:          996      0    996
Total:        1721    708   1013
```

$ free –g **List memory usage in gigabytes**

$ free -s 5 Continuously display memory usage every 5 seconds

View memory use over a given time period,

```
$ vmstat 3
procs -----------memory---------- --swap-- ----io---- --system-- -----cpu-----
 r  b   swpd   free  buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  0  97740  32488  3196 148360    0    0     0     1   26 3876 85 15  0  0  0
 1  1  98388   7428  3204 151472    0  216     0   333   30 3200 82 18  0  0  0
```

$ vmstat -S M -s | less Display statistics in megabytes
725 M total memory
717 M used memory
486 M active memory

175 M inactive memory
7 M free memory
1 M buffer memory
120 M swap cache
996 M total swap
802 M used swap
193 M free swap

Monitoring CPU Usage

Display a CPU utilization report:

$ iostat -c 3	CPU stats every 3 seconds (starting apps)
$ iostat -c -t	Print time stamp with CPU report
$ iostat -c -t 2 10	Repeat every 2 seconds for 10 times
$ dstat -t -c 3	View CPU usage continuously with time stamps.

Information about the processor itself

$ cat /proc/cpuinfo	View CPU information from /proc

Details about how your storage devices are performing

$ iostat 3	Check disk reads and writes per disk

List information about disk reads and writes

$ vmstat -d	Display disk read, write, and input/output statistics
$ vmstat -p sda1	Display read/write stats for a disk partition

What files and directories are currently open on your storage devices,

# **lsof | less**	List processes holding files and directories open
# **lsof -c bash**	List files open by bash shells
# **lsof -d cwd**	List directories open as current working directory
# **lsof -u ilg**	List files and directories open by user ilg
# **lsof /mnt/sda1**	List anything open on /mnt/sda1 file system
# **lsof +d /mnt/sda1/dx**	List anything open under /mnt/sda1/dx directory

Time/Date

During Fedora startup, the /etc/sysconfig/clock file is read to set your time zone and whether your system is using UTC time.

Your Linux system's time zone is set based on the contents of the /etc/localtime file

You can set a new time zone immediately by copying the file representing your time zone from a subdirectory of /usr/share/zoneinfo.

To change the time zone permanently, set the ZONE value in /etc/sysconfig/clock

Setting Your System Clock

$ date
Sun Aug 12 01:26:50 CDT 2007
Display current date, time and time zone

$ **date '+%A %B %d %G'**
Sunday August 12 2007
Display day, month, day of month, year

$ date '+The date today is %F.'
The date today is 2007-08-12
Add words to the date output

$ date --date='4 weeks'
Sun Sep 9 10:51:18 CDT 2007
Display date four weeks from today

$ date --date='8 months 3 days'
Tue Apr 15 10:59:44 CDT 2008
Display date 8 months 3 days from today

$ date --date='4 Jul' +%A
Wednesday
Display day on which July 4 falls

Display dates by month.

$ cal	Show current month calendar (today is highlighted)
$ cal 2009	Show whole year's calendar
$ cal -j	Show Julian calendar (numbered from January 1)
# **date 081215212009**	Set date/time to Aug. 12, 2:21PM, 2008
# **date --set='+7 minutes'**	Set time to 7 minutes later
# **date --set='-1 month'**	Set date/time to one month earlier

Setting Your Hardware Clock

# **hwclock -r**	Display current hardware clock settings
# **hwclock --hctosys**	Reset system clock from hardware clock
# **hwclock --systohc**	Reset system clock from hardware clock
# **hwclock --adjust**	Adjust hardware clock time for drift

# **hwclock --set --date="3/18/08 18:22:00"**	Set clock to new date/time

Network Time Protocol

# **yum install ntpd**	Install ntpd package if necessary.
# **service ntpd**	start Start NTP service immediately
# **chkconfig ntpd on**	Set NTP service to start at each reboot

Set SYNC_HWCLOCK to yes (SYNC_HWCLOCK=yes) in the /etc/sysconfig/ntpd to update the hardware clock.

# **ntpdate** pool.ntp.org	Configured in daily cron job

UDP port 123 must be open on firewall for allowing the communication between servers to update time.

ntpd –qg

–q option tells ntpd to exit after setting the clock (as opposed to keep running as a daemon). The –g option prevents ntpd from panicking if the system clock is off by more than 1000 seconds.

Managing the Boot Process

GRUB Boot Loader

The settings for your boot loader are stored in **the /boot/grub/grub.conf** file.

```
default=0
timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title Fedora (2.6.21-1.3194.fc7)
root (hd0,0)
kernel /boot/vmlinuz-2.6.21-1.3194.fc7 ro root=LABEL=/1 rhgb quiet
initrd /boot/initrd-2.6.21-1.3194.fc7.img
```

Note: hiddenmenu line indicates that you won't see the list of bootable titles when the boot splash screen appears (press a key during the timeout period to see the menu).

Reinstall GRUB

To reinstall GRUB on your hard disk's MBR, boot a Fedora live CD or install CD in rescue mode and follow the instructions to change root (chroot) to the hard disk partition containing your Fedora system. Then, assuming that you're booting from your computer's first SATA hard disk, type the following to reload the boot loader to the MBR:

grub-install /dev/sda

Repairing the initial ramdisk (initrd)

Initrd file is located in /boot with a name like initrd-2.6.20-1.2316.fc5.img.

If your initrd becomes corrupted, or if you need to add new block device drivers to it, run the mkinitrd command.

First, make sure you make a copy of your existing initrd file. Then run the following command:

 # mkinitrd –v –f /boot/initrd-`uname –r`.img `uname –r`

Controlling Startup and Run Levels

see the current run level

# **runlevel**	Display current and previous run levels
N 3	
# **who -r**	Same as above.

change the current run level

# init 5	Change the current run level to 5 (X Desktop), if you want togo into GUI mode.

Reexamine the /etc/inittab file and start or stop processes based on changes made to that file

# init q	Start or stop changed processes in inittab file

Manage services, you can use the chkconfig and service commands.

# service smb start	Start Samba service immediately
# service smb restart	Restart Samba service (first off, then on)
Shutting down SMB services: [OK]	
Shutting down NMB services: [OK]	
Starting SMB services: [OK]	
Starting NMB services: [OK]	
# service smb condrestart	Restart Samba service (if already running)
Shutting down SMB services: [OK]	
Shutting down NMB services: [OK]	
Starting SMB services: [OK]	
Starting NMB services: [OK]	
# service smb reload	Reload settings in smb.conf file
Reloading smb.conf file: [OK]	
# service smb status Check if the Samba service is running (smbd)	
smbd (pid 25917 25915) is running...	
# service smb stop	Stop Samba service
Shutting down SMB services: [OK]	
Shutting down NMB services: [OK]	

List services, turn them on, or turn them off on a per–run level basis.

# chkconfig smb on	Turn on the Samba service
# chkconfig --list smb	List runlevels service is on or off
smb 0:off 1:off 2:on 3:on 4:on 5:on 6:off	
# chkconfig –list	List all services, indicating on or off
# chkconfig --level 2 smb off	Turn off Samba service for run level 2
# chkconfig -–add mydaemon	Add /etc/init.d/mydaemon to chkconfig

The init command to change to any run level, including init 0 (shut down) and init 6 (reboot), there are also specific commands for stopping Linux. The advantages of commands such as halt, reboot, poweroff, and shutdown are that they include options to let you stop some features before shutdown occurs.

# reboot	Reboot the computer
# halt -n	Don't run sync to sync hard drives before shutdown
# halt -h	Put hard drives in standby mode before halting
# shutdown 10	Shutdown in ten minutes after warning the users
# shutdown –r 10	Reboot in ten minutes after warning the users
# shutdown 10 'Bye!'	Send custom message to users before shutdown

What kernel is currently running on your system

$ uname -r Display name of kernel release
2.6.21-1.3194.fc7
$ uname -a Display all available kernel info
Linux server.domain.com 2.6.20-1.2320.fc5 #1 SMP Tue Jun 12 18:50:49 EDT 2007 x86_64 x86_64 x86_64 GNU/Linux

Display the contents of the kernel ring buffer

$ dmesg |less

Information of interest about kernel processing

 # cat /var/log/messages* | less

View the names of the loaded modules,

 # lsmod

Fnd out more information about a particular module,

 # modinfo snd_ens1371

List all available modules and remove modules.

# modprobe -l | grep c-qcam	List all modules, then look for c-qcam
# modprobe c-qcam	Load module for Color QuickCam
# modprobe -r c-qcam	Remove module for Color QuickCam

Control kernel parameters with the system running

# sysctl -a | less	List all kernel parameters
# sysctl kernel.hostname	List value of particular parameter
# sysctl –p	Load parms from /etc/sysctl.conf

sysctl -w kernel.hostname=ilg Set value of kernel.hostname

Poking at the Hardware

lspci List PCI hardware items
lspci –v List PCI hardware items with more details
lspci -vv List PCI hardware items with even more details
lsusb List USB Devices

Display information about your computer's hardware components

dmidecode | less List hardware components

Exercise : Check for 'biosdecode' command.
View and change information relating to your hard disk.

hdparm /dev/sda Display hard disk settings (SATA or SCSI drive)

/dev/sda:
IO_support = 0 (default 16-bit)
readonly = 0 (off)
readahead = 256 (on)
geometry = 30401/255/63, sectors = 488395055, start = 0

hdparm /dev/hda Display hard disk settings (IDE drive)

hdparm –I /dev/sda Display detailed drive information
/dev/sda:
ATA device, with non-removable media
Model Number: FUJITSU MPG3409AT E
Serial Number: VH06T190RV9W
Firmware Revision: 82C5
…..

Chapter 11

Managing Network Connections

Managing Network Interface Cards

man ethtool View options to the ethtool command

display settings for a specific Ethernet card,

ethtool eth0 See settings for NIC at eth0
Settings for eth0:
Supported ports: [TP MII]
Supported link modes: 10baseT/Half 10baseT/Full
100baseT/Half 100baseT/Full
Supports auto-negotiation: Yes
Advertised link modes: 10baseT/Half 10baseT/Full
100baseT/Half 100baseT/Full
Advertised auto-negotiation: Yes
Speed: 100Mb/s
Duplex: Full
Port: MII

PHYAD: 1
Transceiver: internal
Auto-negotiation: on
Supports Wake-on: g
Wake-on: g
Current message level: 0x00000007 (7)
Link detected: yes

Find out about the driver being used for a particular network card,

ethtool -i eth0 Display driver information for NIC
driver: e1000
version: 7.3.15-k2-NAPI
firmware-version: 0.5-7
bus-info: 0000:04:00.0

ethtool -S eth0 Show statistics for NIC at eth0
NIC statistics:
rx_packets: 1326384
tx_packets: 773046
rx_bytes: 1109944723
tx_bytes: 432773480
rx_errors: 5
tx_errors: 2
rx_dropped: 0
tx_dropped: 0
multicast: 0
collisions: 0
rx_length_errors: 0
rx_over_errors: 0
rx_crc_errors: 5
rx_frame_errors: 0
rx_fifo_errors: 0
rx_missed_errors: 0
tx_aborted_errors: 0
tx_carrier_errors: 2

ethtool -s eth0 speed 100 duplex full autoneg off Change NIC settings

make these settings stick at the next reboot or network restart, add the options you want to the ETHTOOL_OPTS line in the /etc/ sysconfig/network-scripts/ifcfg-eth0

ETHTOOL_OPTS=”speed 10 duplex half autoneg off”

if you have an older NIC, try using mii-tool

mii-tool Show negotiated speed and link status of old NIC
eth0: negotiated 100baseTx-FD flow-control, link ok

Display mii-tool output with more verbosity

#mii-tool -v Show verbose output of settings for old NIC
eth0: negotiated 100baseTx-FD flow-control, link ok
product info: vendor 00:50:43, model 12 rev 2
basic mode: autonegotiation enabled
basic status: autonegotiation complete, link ok
capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

advertising: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control
link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control

Disable auto-negotiation and force a particular setting

 # mii-tool -F 10baseT-FD eth0 Force speed/duplex to 10baseT-FD

re-enable auto-negotiation,

 # mii-tool -r eth0 Re-enable auto-negotiation for an old NIC

Get network interface statistics

$ netstat -i Get network interface statistics for eth0

Refresh network interface statistics every second

$ netstat -ic Refresh network statistics every second

Get cleaner (screen-oriented) refreshed output from netstat

$ watch netstat -i Refresh network statistics (screen oriented)

Managing Network Connections

For Fedora, RHEL, and CentOS, control scripts and configuration files are located in the /**etc/sysconfig/network-scripts/** directory. NICs are configured by editing **/etc/sysconfig/network-scripts/ifcfg-interface**, where interface is your NIC's network interface.

To take all NICs offline then bring them back online,

 # service network restart Shutdown and bring up network interfaces
Shutting down interface eth0: [OK]
Shutting down loopback interface: [OK]
Bringing up loopback interface: [OK]
Bringing up interface eth0: [OK]

 # service network stop Shutdown network interfaces
 # service network start Bring up network interfaces
 # service network status Check network interface status
Configured devices:
lo eth0
Currently active devices:
lo eth0

To configure your network connections to start when Linux boots

```
# chkconfig network on          Turn on network service to start at boot
# chkconfig --list network     View runlevels where network is off or on
network      0:off   1:off   2:on   3:on   4:on   5:on   6:off
```

Multiple network interfaces, you may want to just bring one interface up or down.

# ifdown eth0	Take the eth0 network interface offline
# ifup eth0	Bring the eth0 network interface online

Viewing Ethernet Connection Information

shows the address information and status of your eth0 Ethernet interface

 # ifconfig eth0
eth0 Link encap:Ethernet HWaddr 00:D0:B7:79:A5:35
inet addr:10.0.0.155 Bcast:10.0.0.255 Mask:255.255.255.0
inet6 addr: fe80::2d0:b7ff:fe79:a535/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:1413382 errors:6 dropped:0 overruns:0 frame:6
TX packets:834839 errors:4 dropped:0 overruns:0 carrier:4
collisions:0 txqueuelen:1000
RX bytes:1141608691 (1.0 GiB) TX bytes:470961026 (449.1 MiB)

Get information on both active and inactive NICs

 # ifconfig -a

show information about the eth0 interface:

 # ip addr show eth0
1: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
link/ether 00:d0:b7:79:a5:35 brd ff:ff:ff:ff:ff:ff
inet 10.0.0.155/24 brd 10.0.0.255 scope global eth0
inet6 fe80::2d0:b7ff:fe79:a535/64 scope link
valid_lft forever preferred_lft forever

To see ip address

#ip addr show

#ip a

 # ip link set eth1 up
Calculate a host computer's netmask from its CIDR IP address:

$ ipcalc -bmn 192.168.1.100/27
NETMASK=255.255.255.224
BROADCAST=192.168.1.127
NETWORK=192.168.1.96

Using Wireless Connections

Determining exactly what wireless card you have,

# lspci | grep -i wireless	Search for wireless PCI cards

scans your network interfaces for supported wireless cards

 # iwconfig
eth0 no wireless extensions.
eth1 IEEE 802.11-DS ESSID:"" Nickname:"HERMES I"
Mode:Managed Frequency:2.457 GHz Access Point: Not-Associated

Bit Rate:11 Mb/s Tx-Power=15 dBm Sensitivity:1/3
Retry limit:4 RTS thr:off Fragment thr:off
Encryption key:off
Power Management:off

 # iwconfig eth1
eth1 IEEE 802.11-DS ESSID:"" Nickname:"HERMES I"
Mode:Managed Frequency:2.457 GHz Access Point: None
Bit Rate:11 Mb/s Tx-Power=15 dBm Sensitivity:1/3
Retry limit:4 RTS thr:off Fragment thr:off
Encryption key:off
Power Management:off
Link Quality=0/92 Signal level=134/153 Noise level=134/153
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0

Use iwconfig to modify your wireless interface settings.

# iwconfig wlan0 essid "MyWireless"	Set essid to MyWireless
# iwconfig wlan0 channel 3	Set the channel to 3
# iwconfig wlan0 mode Ad-Hoc	Change from Managed to Ad-Hoc mode
# iwconfig wlan0 ap any	Use any access point available
# iwconfig wlan0 sens -50	Set sensitivity to –50
# iwconfig wlan0 retry 20	Set MAC retransmissions to 20
# iwconfig wlan0 key 1234-5555-66	Set encryption key to 1234-5555-66

Checking Name Resolution

When you first installed Fedora, you either identified Domain Name System (DNS) servers to do name resolution or had them assigned automatically from a DHCP server. That information is then stored in the **/etc/resolv.conf** file

nameserver 11.22.33.44
nameserver 22.33.44.55

search your DNS servers for a particular host name

$ dig www.turbosphere.com Search DNS servers set in /etc/resolv.conf

Query a specific name server.

$ dig www.turbosphere.com @4.2.2.1

query for a specific record type:

$ dig turbosphere.com mx	Queries for the mail exchanger
$ dig turbosphere.com ns	Queries for the authoritative name servers

$ dig +trace www.turbosphere.com Recursively trace DNS servers

$ dig +short www.turbosphere.com Display only name/IP address pair

Do a reverse lookup to find DNS information based on an IP address:

$ dig -x 66.113.99.70 Get DNS information based on IP address

$ host 66.113.99.70

$ hostname	View the local computer's full DNS host name
$ hostname -s	View the local computer's short host name
$ hostname -d	View the local computer's domain name
$ dnsdomainname	Another way to view the local domain name
# **hostname server1.example.com**	Set local hostname

Note: set the local hostname so it is set each time the system starts up in /etc/sysconfig/network

HOSTNAME=server1.example.com

Troubleshooting network problems.

Checking Connectivity to a Host

You should have configured the default gateway (gw) either in the /etc/sysconfig/network file or in the individual network card's /etc/sysconfig/ network-script/ifcfg-eth? script. To check your default gateway in the actual routing table, use the ip command as follows

 # ip route
10.0.0.0/24 dev eth0 proto kernel scope link src 10.0.0.155
169.254.0.0/16 dev eth0 scope link
default via 10.0.0.1 dev eth0

$ ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.382 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.313 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.360 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=1.43 ms

….
Ctrl +c

$ ping -a 10.0.0.1	Add an audible ping as ping progresses
$ ping -c 4 10.0.0.1	Ping 4 times and exit (default in Windows)
$ ping -q -c 5 10.0.0.1	Show summary of pings (works best with -c)
# **ping -f 10.0.0.1**	Send a flood of pings (must be root)

$ ping -i 3 10.0.0.1	Send packets in 3-second intervals
# **ping -I eth0 10.0.0.1**	Set source to eth0 (use if multiple NICs)

PING 10.0.0.1 (10.0.0.1) from 10.0.0.155 eth0: 56(84) bytes of data.

# **ping -I 10.0.0.155 10.0.0.1**	Set source to 10.0.0.155

PING 10.0.0.1 (10.0.0.1) from 10.0.0.155 : 56(84) bytes of data.

$ ping -s 1500 10.0.0.1 Set packet size to 1500 bytes
PING 10.0.0.1 (10.0.0.1) 1500(1528) bytes of data.

Checking Address Resolution Protocol (ARP)

Address Resolution Protocol (ARP) can be used to find information at the MAC layer

 # arp -v List ARP cache entries by name

```
Address          HWtype HWaddress        Flags Mask      Iface
ritchie   ether        00:10:5A:AB:F6:A7 C                     eth0
einstein        ether 00:0B:6A:02:EC:98      C              eth0
Entries: 1 Skipped: 0 Found: 1
```

 # arp –vn List ARP cache entries by IP address

```
Address          HWtype          HWaddress      Flags Mask       Iface
10.0.0.1        ether 00:10:5A:AB:F6:A7     C              eth0
10.0.0.50       ether 00:0B:6A:02:EC:98     C              eth0
Entries: 1 Skipped: 0 Found: 1
```

Delete an entry from the ARP cache

 # arp -d 10.0.0.50 Delete address 10.0.0.50 from ARP cache

Add static ARP entries to the cache

 # arp -s 10.0.0.51 00:0B:6A:02:EC:95 Add IP and MAC addresses to ARP

Arp with IP command

 # ip neighbor

```
10.0.0.1 dev eth0 lladdr 00:10:5a:ab:f6:a7 DELAY
10.0.0.50 dev eth0 lladdr 00:0b:6a:02:ec:98 REACHABLE
```

 # ip nei del 10.0.0.50 dev eth0
 # ip n add 10.0.0.51 lladdr 00:0B:6A:02:EC:95 dev eth0

Query a subnet to see if an IP is already in use

 # arping 10.0.0.50 Query subnet to see if 10.0.0.50 is in use

```
ARPING 10.0.0.50 from 10.0.0.195 eth0
Unicast reply from 10.0.0.50 [00:0B:6A:02:EC:98] 0.694ms
Unicast reply from 10.0.0.50 [00:0B:6A:02:EC:98] 0.683ms
```

 # arping -I eth0 10.0.0.50 Specify interface to query from

 # arping -f 10.0.0.50 Query 10.0.0.50 and stop at the first reply
 # arping -c 2 10.0.0.51 Query 10.0.0.50 and stop after 2 counts

Tracing Routes to Hosts

Use traceroute to find the bottleneck or point of failure:

$ traceroute boost.turbosphere.com Follow the route taken to a host

Note:Lines of asterisks (*) at the end of the trace can be caused by firewalls that block traffic to the target.

Note:Traceroute uses UDP packets, which provide a more realistic performance picture than ICMP

Trace using ICMP packets

 # traceroute -I boost.turbosphere.com Use ICMP packets to trace a route
 # tracert boost.turbosphere.com Use ICMP packets to trace a route

Trace a route to a remote host using TCP packets

 # traceroute -T boost.turbosphere.com Use TCP packets to trace a route

 # traceroute -T -p 25 boost.turbosphere.com Connect to port 25 in trace

Note: By default, traceroute connects to port 80.

$ traceroute -n boost.turbosphere.com Disable name resolution in trace

$ tracepath boost.turbosphere.com Use UDP to trace the route

Display your local routing table.

 # route Display local routing table information

Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
10.0.0.0 * 255.255.255.0 U 0 0 0 eth0
default ritchie 0.0.0.0 UG 0 0 0 eth0

 # route -n Display routing table without DNS lookup
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
10.0.0.0 * 255.255.255.0 U 0 0 0 eth0
0.0.0.0 10.0.0.1 0.0.0.0 UG 0 0 0 eth0

 # route add default gw 10.0.0.2 Add 10.0.0.2 as default gateway

 # route add -net 192.168.0.0 netmask 255.255.255.0 eth0
or
 # route add -net 192.168.0.0 netmask 255.255.255.0 gw 10.0.0.100

Delete a route

 # route del -net 192.168.0.0 netmask 255.255.255.0 Delete a route

New ip commands

 # ip route show Display basic routing information
10.0.0.0/24 dev eth0 proto kernel scope link src 10.0.0.195
169.254.0.0/16 dev eth0 scope link

default via 10.0.0.1 dev eth0

 # ip route Display basic routing (example #2)
 # ip r Display basic routing (example #3)

Adding and deleting routes with ip:
 # ip r add 192.168.0.0/24 via 10.0.0.100 dev eth0 Add route to interface
 # ip r add 192.168.0.0/24 via 10.0.0.100 Add route no interface
 # ip r del 192.168.0.0/24 Delete route

Make a new route permanent, create a **/etc/sysconfig/network-scripts/** file named **route-ethX** (for example, route-eth0) and place the information about the new route in that file e.g **to add the route added with the ip command above, add the following lines to /etc/sysconfig/network-scripts/route-eth0**

ADDRESS=192.168.0.0
NETMASK=255.255.255.0
GATEWAY=10.0.0.100

Netstat Connections and Statistics

$ netstat -s | less Show summary of TCP, ICMP, UDP activities

List of all TCP connections, including which process is handling the connection:

netstat –tanp View active TCP connections
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
tcp 0 0 127.0.0.1:631 0.0.0.0:* LISTEN 2039/cupsd
tcp 0 0 127.0.0.1:25 0.0.0.0:* LISTEN 2088/sendmail
...

View active UDP connections as follows
 # netstat -uanp View active UDP connections
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
udp 0 0 0.0.0.0:631 0.0.0.0:* 2039/cupsd
udp 0 0 192.168.122.1:123 0.0.0.0:* 2067/ntpd
...

To narrow your output from netstat to daemons bound to a TCP port, look for the word listen

 # netstat -tanp | grep -i listen View daemons listening to a port

Explore networks and remote machines and see what services they offer

 # nmap 10.0.0.1 Scan ports on computer at 10.0.0.1
To get maximum verbosity from nmap, use the -vv option:

 # nmap -vv 10.0.0.1 Show maximum verbosity from nmap output

To use nmap to scan an entire network, use the network address as an argument.

We add the –sP option to tell nmap to perform a simple ping sweep

 # nmap -vv –sP 10.0.0.0/24 Scan hosts on an entire network

The -P0 option tells nmap not to use ping (this is good for scanning machines that don't respond to ping). The -O option displays OS fingerprinting for the machine you are scanning. The -p 100-200 option tells nmap to scan only ports 100 through 200:

 # nmap -vv -P0 -O -p 100-200 10.0.0.1 No ping, OS fingerprint, ports 100-200

To start stop network service

#service network start To start network service
#service network stop To stop the network service
#service network restart To restart network service

Or

 # /etc/init.d/network start
 # /etc/init.d/network stop
 # /etc/init.d/network restart

$$

Chapter 12

Accessing Network Resources

Downloading Files with wget

$ wget http://dag.wieers.com/rpm/packages/acroread/acroread-5.0.10-1.el5.rf.i386.rpm

FTP with username password on command line

$ wget ftp://user:password@ftp.example.com/path/to/file
$ wget --user=user --password=password ftp://ftp.example.com/path/to/file

Download webpages will be linked locally, so you won't have broken links after the pages from website have been downloaded.

$ wget -pk http://www.wiley.com Download pages and use local file names

Using –c – continue feature of the wget

$ wget -c http://example.com/DVD.iso Resume download where stopped

The download will be continued, in case you got disconnected from process inbetween.

Transfering Files with FTP Commands

Connect to an FTP server with lftp

$ lftp mirrors.kernel.org Anonymous connection
lftp mirrors.kernel.org:~>

$ lftp ilg@example.com Authenticated connection
lftp example.com:~>
$ lftp -u ilg example.com Authenticated connection
Password: ******
lftp example.com:~>
$ lftp -u ilg,Mypwd example.com Authentication with password
lftp example.com:~>
$ lftp Start lftp with no connection
lftp **:~> open mirrors.kernel.org** Start connection in lftp session
lftp mirrors.kernel.org:~>

When a connection is established to an FTP server, you can use a set of commands during the FTP session. FTP commands are similar to shell commands. Just like in a bash shell, you can press Tab to autocomplete file names. In a session, lftp also supports sending multiple jobs to the background (Ctrl+z) and returning them to foreground (wait or fg). These are useful if you want to continue traversing the FTP site while files are downloading or uploading. Background jobs run in parallel. Type jobs to see a list of running background jobs. Type help to see a list of lftp commands.

Usage

$ lftp mirrors.kernel.org
lftp mirrors.kernel.org:~> **pwd** Check current directory
ftp://mirrors.kernel.org
lftp mirrors.kernel.org:~> **ls** List current directory
drwxr-sr-x 8 400 400 4096 Jul 02 20:19 debian/
drwxr-xr-x 7 537 537 77 May 21 21:37 fedora/
...
lftp mirrors.kernel.org:~> **cd fedora/releases/7/Live/i386** Change directory
lftp mirrors.kernel.org:...> **get Fedora-7-Live-i686.iso** Download a file
Fedora-7-Live-i686.iso at 776398 (1%) 467.2K/s eta:26m {Receiving data]
lftp mirrors.kernel.org:...> **<Ctrl+z>** Send download to background
lftp mirrors.kernel.org:...> **mget /gnu/ed/*** Get all in /gnu/ed
lftp mirrors.kernel.org:...> **!ls** Run local ls
lftp mirrors.kernel.org:...> **bookmark add Live** Bookmark location
lftp mirrors.kernel.org:...> **quit** Close lftp

SSH Tools to Transfer Files

Copying Remote Files with scp

$ scp myfile ilg@server1:/tmp/ Copy myfile to server1
Password: ******
$ scp server1:/tmp/myfile . Copy remote myfile to local working dir
Password: ******

$ scp -p myfile server1:/tmp/ preserve permissions and timestamps

SSH service is configured to listen on a port other than the default port 22

$ scp -P 12345 myfile server1:/tmp/ Connect to a particular port

Recursive copies

$ scp -r mydir ilg@server1:/tmp/ Copies all mydir to remote /tmp

Copying Remote Files in sftp and lftp Sessions

$ sftp ilg@server1
ilg@server1's password: *****
sftp>

Usage with lftp

$ lftp sftp://ilg@server1
Password: ********
lftp ilg@server1:~>

Chapter 13

Locking Down Security

Working with Users and Groups

Add new users

 # useradd ilg Add user into system

set of configuration files (each beginning with a ".") copied to the home directory from /etc/skel.

Check the default values for the system.

# useradd -D	Show useradd default values
GROUP=100	Set group ID to 100 (users)
HOME=/home	Set base home directory to /home
INACTIVE=-1	Password expiration is disabled (-1)
EXPIRE=	Don't set date to disable user account
SHELL=/bin/bash	Set the default shell to /bin/bash
SKEL=/etc/skel	Copy default config files from /etc/skel to $HOME
CREATE_MAIL_SPOOL=yes	Create a mail spool directory

Add a password

 # passwd ilg
Changing password for user ilg.
New UNIX password: ********
Retype new UNIX password: ********
passwd: all authentication tokens updated successfully.

Override the defaults when you create a user.

# useradd -u 1001 -g 300 sherlock	Use specific UID and GID for user
# useradd -d /home/jj jags	Create /home/jj home directory
# useradd -G *support,sales* timd	Add user to support and sales groups
# useradd -c "Sherlock holmes" sherlock	Add user's full name to comment field
# useradd -s /bin/tcsh joe	Assign a new default shell (tcsh)
# useradd -e 2008-04-01 joe	Add account to expire April 01, 2008
# useradd -f 0 test	*Create a disabled account*
# useradd -s /sbin/nologin fake	Keep user from shelling in

# useradd -M fakedir	Prevent creation of home directory

list the group(s) that a user belongs to

$ groups ilg	List the groups that a user belongs to

Changing useradd Defaults

# useradd -D	List default settings for useradd
# useradd -D -b /home2 -s /bin/csh	Set default base dir and shell
# useradd -D -e 2009-01-01	Set all new users to expire in 2009

Modifying User Accounts

After account has been created we can change values for that account with the usermod command

# usermod -c “Sherlock Holmes” sherlock	Change user’s name in comment field
# usermod -s /bin/sh john	Change default shell to sh
# usermod -L swan	Lock the user account named swanson
# usermod -U stallman	Unlock user account named stallman

Changing personal account information , as useradd and usermod cannot be used by normal user.

$ chsh -s /bin/zsh	Change urrent user’s shell to /bin/zsh
# chsh -s /bin/zsh ilg	Change a user’s shell to /bin/zsh

$ finger ilg
Login: ilg Name: Ilg group
Directory: /home/ilg Shell: /bin/bash
Office: B-205, 212-555-1212 Home Phone: 212-555-1957
On since Sat Aug 4 13:39 (CDT) on tty1 4 seconds idle
No mail.
No Plan.

Deleting User Accounts

Remove user accounts from the system

# userdel jimbo	Delete user, not user’s home directory
# userdel -r test	Delete user, home directory, and mail spool

Managing Passwords

Change only their own passwords,

$ passwd	Change a regular user’s own password

Changing password for user chris.
Changing password for chris.
(current) UNIX password: ********
New UNIX password: *
BAD PASSWORD: it’s WAY too short
New UNIX password: *********
Retype new UNIX password: *********
passwd: all authentication tokens updated successfully

 # passwd yahoo Root can change any user's password
Changing password for user yahoo.
New UNIX password: *
BAD PASSWORD: it's WAY too short
Retype new UNIX password: *
passwd: all authentication tokens updated successfully.

lock and unlock user accounts.

 # passwd -l carl Lock the user account (carl)
Locking password for user carl.
passwd: Success

 # passwd -u carl Unlock a locked user account (carl)
Unlocking password for user carl.
passwd: Success

 # passwd -u jordan Fails to unlock account with blank password
Unlocking password for user jordan.
passwd: Warning: unlocked password would be empty.
passwd: Unsafe operation (use -f to force)

 # passwd -u -f jordan Able to unlock user with blank password
Unlocking password for user jordan.
passwd: Success

Require users to change passwords regularly, as well as warn users when passwords are about to expire

 # passwd -n 2 vern Set minimum password life to 2 days
 # passwd -x 300 vern Set maximum password life to 300 days
 # passwd -w 10 vern Warn of password expiration 10 days in advance
 # passwd -i 14 vern Days after expiration account is disabled

View password expiration

 # chage -l vern View password expiration information
Last password change : Aug 04, 2007
Password expires : May 31, 2009
Password inactive : Jun 14, 2009
Account expires : never
Minimum number of days between password change : 2
Maximum number of days between password change : 300
Number of days of warning before password expires : 10

chage can also set the day when a user must set a new password or a particular date the account becomes inactive:

 # chage -I 40 yahoo Make account inactive in 40 days

 # chage -d 5 perry Force user's password to expire in 5 days

Note: set that option to 0 and cause the user to have to set a new password the next time he or she logs in.

Adding Groups

 # groupadd marketing Create new group with next available GID
 # groupadd -g 701 sales Create new group with GID of 701
 # groupadd -r myadmin Create group with admin GID (under 499 or 1000)
 # groupadd -o -g 74 mysshd Create group with existing GID

Change the name or group ID

 # groupmod -g 491 myadmin Modify myadmin to use GID 491
 # groupmod -n myad myadmin Change name of myadmin group to myad

Remove an existing group

 # groupdel myad Remove existing myad group

Checking on Users

Getting information about people logging into your system:

$ last List the most recent successful logins
greek tty3 Sun Aug 5 18:05 still logged in
chris tty1 Sun Aug 4 13:39 still logged in
root pts/4 ilg Sun Aug 5 14:02 still logged in

$ last -a Makes it easier to read the remote client hostname

 # lastb List the most recent unsuccessful logins
julian ssh:notty ritchie Mon Aug 6 12:28 - 12:28 (00:00)
morris ssh:notty thompson Tue Jul 31 13:08 - 13:08 (00:00)

$ who –u List who is currently logged in (long form)
greek tty3 2007-08-05 18:05 17:24 18121
jim pts/0 2007-08-06 12:29 . 20959 (server1.example.com)

$ users List who is currently logged in (short form)
chris francois greek jim root

Finding out more about individual users on your system:

$ id Your identity (UID, GID and group for current shell)
uid=501(chris) gid=501(chris) groups=501(chris)

$ whoami Your identity (user, tty, login date, location)
chris pts/0 Aug 3 2140 (:0.0)

$ finger -s chris User information (short)
Login Name Tty Idle Login Time Office Office Phone
chris Chris Negus tty1 1d Aug 4 13:39 A-111 555-1212

$ finger -l chris User information (long)
Login: chris Name: Chris Negus
Directory: /home/chris Shell: /bin/bash
Office: A-111, 555-1212 Home Phone: 555-2323
On since Sat Aug 4 13:39 (CDT) on tty1 2 days idle
New mail received Mon Aug 6 13:46 2007 (CDT)
Unread since Sat Aug 4 09:32 2007 (CDT)
No Plan.

Working with System Logs

Fedora, RHEL, and CentOS use the **syslogd (system log daemon)** and **klogd (kernel log daemon)** from the sysklogd package to manage system logging. Those daemons are started automatically from the syslog init script (/etc/init.d/syslog). Information about system activities is then directed to files in the /var/log directory such as messages, secure, cron, and boot.log, based on settings in the **/etc/syslog.conf** file.

Automatic log rotation is handled by logrotate, based on settings in the /etc/logrotate.conf file and /etc/logrotate.d directory. The /etc/cron.daily/logrotate cronjob causes this daily log rotating to take place.

Send your own messages to the syslogd logging facility

# **logger Added new video card**	Message added to messages file
# **logger -p info -t CARD -f /tmp/my.txt**	Priority, tag, message file

Vim Editor

$ vi +25 /tmp/inittab	Begin on line 25
$ vi + /tmp/inittab	Begin editing file on the last line
$ vi +/tty /tmp/inittab	Begin on first line with word “tty”
$ vi -r /tmp/inittab	Recover file from crashed edit session
$ view /tmp/inittab	Edit file in read-only mode

:w	**Save the file before you are ready to quit**
ZZ	**To quit and save changes**

Or

:wq

:q	**To quit without saving changes**

Or

:q!

:w /tmp/myfile.txt	**To save a file in different name**

Line up several files at a time to edit.

$ vi a.txt b.txt c.txt

Moving to next file by pressing **:n**

Split your screen multiple times either horizontally or vertically

:split /etc/httpd/conf/httpd.conf

:vsplit /etc/init.d/httpd

Navigate between split windows ,Pres **ctrl+w**, followed by **w** key

Editor keys

Key	Result	Key	Result
PageDown or Ctrl+f	Move down one page	PageUp or Ctrl+b	Move up one page
Ctrl+d	Move down half page	Ctrl+u	Move up half page
Shift+g	Go to last line of file	:1	Go to first line of file (use any number to go to that line)
Shift+h	Move cursor to screen top	Shift+l	Move cursor to screen bottom
Shift+m	Move cursor to middle of screen	Ctrl+l	Redraw screen (if garbled)
Enter	Move cursor to beginning of the next line	-	Move cursor to beginning of the previous line
Home or $	Move cursor to end of line	End or ^ or 0	Move cursor to line beginning
(	Move cursor to beginning of previous sentence	)	Move cursor to beginning of next sentence
{	Move cursor to beginning of previous paragraph	}	Move cursor to beginning of next paragraph
w	Move cursor to next word (space, new line, or punctuation)	Shift+w	Move cursor to next word (space or new line)

b	Move cursor to previous word (space, new line, or punctuation)	Shift+b	Move cursor to previous word (space or new line)
e	Move cursor to end of next word (space, new line, or punctuation)	Shift+e	Move cursor to end of next word (space or new line)
Left arrow or Backspace	Move cursor left one letter	Right arrow or l	Move cursor right one letter
k or up arrow	Move cursor up one line	j or down arrow	Move cursor down one line
/string	Find next occurrence of *string*	?*string*	Find previous occurrence of *string*
n	Find same string again (forward)	Shift+n	Find same string again (backwards)

i	Typed text appears before current character	Shift+i	Typed text appears at the beginning of current line
a	Typed text appears after current character	Shift+a	Typed text appears at the end of current line

o	Open a new line below current line to begin typing	Shift+o	Open a new line above current line to begin typing
s	Erase current character and replace with new text	Shift+s	Erase current line and enter new text
c?	Replace ? with l, w, $, or c to change the current letter, word, end of line, or line	Shift+c	Erase from cursor to end of line and enter new text
r	Replace current character with the next one you type	Shift+r	Overwrite as you type from current character going forward

x	Delete text under cursor	Shift+x	Delete text to left of cursor
d?	Replace ? with l, w, $, or d to cut the current letter, word, end of line from cursor, or entire line	Shift+d	Cut from cursor to end of line
y?	Replace ? with l, w, or $ to copy (yank) the current letter, word, or end of line from cursor	Shift+y	Yank current line
p	Pastes cut or yanked text after cursor	Shift+p	Pastes cut or yanked text before cursor

u	Type **u** to undo the previous change. Multiple u commands will step back to undo multiple changes.
.	Typing a period (.) will repeat the previous command. So, if you deleted a line, replaced a word, changed four letters, and so on, the same command will be done wherever the cursor is currently located. (Entering input mode again resets it.)
Shift+j	Join the current line with the next line.
Esc	If you didn't catch this earlier, the Esc key returns you from an input mode back to command mode. This is one of the keys you will use most often.

`7cw`	Erase the next seven words and replace them with text you type
`5, Shift+d`	Cut the next five lines (including the current line)
`3p`	Paste the previously deleted text three times after the current cursor
`9db`	Cut the nine words before the current cursor
`10j`	Move the cursor down ten lines
`y2)`	Copy (yank) text from cursor to end of next two sentences
`5, Ctrl+f`	Move forward five pages
`6, Shift+j`	Join the next six lines

Set parameters

`:set all`	List all settings.
`:set`	List only those settings that have changed from the default.
`:set number`	Have line numbers appear left of each line. (Use `set nonu` to unset.)
`:set ai`	Sets autoindent, so opening a new line follows the previous indent.
`:set ic`	Sets ignore case, so text searches will match regardless of case.
`:set list`	Show `$` for end of lines and `^I` for tabs.
`:set wm`	Causes vi to add line breaks between words near the end of a line.

Working in Visual Mode

The Vim editor provides a more intuitive means of selecting text called visual mode. To begin visual mode, move the cursor to the first character of the text you want to select and press the v key. You will see that you are in visual mode because the following text appears at the bottom of the screen:

-- VISUAL --

At this point, you can use any of your cursor movement keys (arrow keys, Page Down, End, and so on) to move the cursor to the end of the text you want to select. As the page and cursor move, you will see text being highlighted. When all the text you want to select is highlighted, you can press keys to act on that text. For example, **d deletes the text**, **c lets you change the selected text**, **:w /tmp/test.txt** saves selected text to a file.

Getting Information from /proc

$ cat /proc/cmdline — Shows options passed to the boot prompt
ro root=LABEL=/123 rhgb quiet

$ cat /proc/cpuinfo — Shows information about your processor
Processor : 0
vendor_id : GenuineIntel
cpu family : 6
model : 8
….

$ cat /proc/devices — Shows existing character and block devices
Character devices:
1 mem
4 /dev/vc/0

$ cat /proc/diskstats	Display disks, partitions, and statistics

1 0 ram0 0 0 0 0 0 0 0 0 0 0 0
1 1 ram1 0 0 0 0 0 0 0 0 0 0 0
...
8 0 sda 2228445 1032474 68692149 21672710 1098740 4003143

$ cat /proc/filesystems	List filesystem types supported by current kernel
$ cat /proc/interrupts	View IRQ channel assignments
$ cat /proc/iomem	Show physical memory addresses
$ cat /proc/ioports	Show virtual memory addresses
$ cat /proc/keys	Displays a list of keys being kept by kernel
$ cat /proc/loadavg	Shows 1, 5, and 15 minute load averages,
$ cat /proc/meminfo	Shows available RAM and swap
$ cat /proc/misc	Shows name/minor number of devices
$ **cat /proc/modules**	Shows loaded modules, memory size,
$ cat /proc/mounts	Show mounted local/remote file system info
$ cat /proc/partitions	Show mounted local disk partitions
$ cat /proc/mdstat	If using software RAID, show RAID status
$ cat /proc/stat	Shows kernel stats since system boot
$ cat /proc/swaps	List information about swap space
$ cat /proc/uptime	Seconds since system booted/total seconds idle
$ cat /proc/version	List kernel version and related compiler

Changing /proc information

To allow forwarding of IPv4 packets, such as to allow a system to do Network Address Translation (NAT) or IP Masquerading

 # echo 1 > /proc/sys/net/ipv4/ip_forward

# sysctl -A | less	Display all kernel runtime parameters
# sysctl -w net.ipv4.ip_forward=1	Turn on IPV4 packet forwarding

It will be updated.....more...more.....practice....