LINUX COMMAND REFERENCE v0.6

Latest version located at http://www.rationallyparanoid.com
Document provided "as is". Author is not liable for any damages resulting from inaccuracies or mistakes within this guide.

File Commands	
cat `filename`	displays the contents of `filename`
cat `filename(s) > newfile`	copies one or more files to a newly created `newfile`
cat `filename >> destination`	appends the contents of `filename` to `destination`
chattr `<option> filename`	changes the file attributes of `filename` (see **lsattr**)
chmod `permission file_or_dir_name`	changes file permissions; (u) user's, (g) group's, (o) others' rights can be (+) added or (-) deleted in (r) read, (w) write or (x) execute mode. **chmod u+x file** adds execute rights to user.
chown `<owner>:<group> file`	changes ownership of `file` to `<owner>` and `<group>`
cp `oldfilename newfilename`	copies `oldfilename` to `newfilename`
file `filename`	displays what type of file `filename` is
find / -name "`filename`**"**	searches entire root directory for `filename`
grep `string filename`	searches for and displays the `string` within `filename`
gunzip `filename`	uncompresses `filename` (if compressed with **gzip**)
gzip `filename`	compresses `filename` (use **gunzip** to uncompress)
head `filename`	displays the first ten lines of `filename`
less `filename`	displays the contents of `filename` with the ability to scroll
ln –s `source target`	creates a soft link from `source` to `target` (i.e. a shortcut)
locate `filename`	searches for and locates `filename` (commands AND files) NOTE: need to update db with root command "**updatedb**"
lsof	lists currently open files
more `filename`	displays the contents of `filename` one screen at a time
mv `oldfilename newfilename`	renames `oldfilename` to `newfilename`
mv `filename directoryname`	moves `filename` to `directoryname`
rm `filename`	removes or deletes `filename`
rpm –Uvh `package-name`	installs `package-name` (upgrades if package already present). Add "--test" to test for dependencies before install.
sort `filename`	alphabetizes (sorts) `filename`
split `–b1m file files.`	splits `file` into multiple `files.[x]` files that are 1 MB each. use cat to re-assemble (ex: `cat game.zip.* > game.zip`)
stat `filename`	provides statistics on `filename` (like windows file properties)
tail `filename`	displays the last ten lines of `filename`. –f to show new output
tar `-cvf newfile.tar fileordirectory`	packs `fileordirectory` and call the package `newfile.tar`. (**tar -xvf newfile.tar** with will unpack the file)
touch `file`	changes file timestamp of `file` (can also create `file`)
tree	shows the directory tree of the current directory
vi	a (brutal) text editor. Use nano instead :)
wc –(`l` or `w` or `c`**)** `file`	counts total lines (`l`), words (`w`) and/or characters (`c`) in `file`
whereis `command`	locates the binary, source, manual of `command` (see **locate**)
which `command`	prints the location of `command` or the alias to which it points

Directory Commands

cd *dir*	changes working directory (leave blank for home directory)
ls	lists files and subdirectories (add **-a** for hidden files/directories, **-F** to put "/" after directories, **-l** for details)
mkdir *directory*	makes a new *directory*
pwd	displays the current working directory
rmdir *directory*	deletes *directory* (only if it is empty)
rm -r *directory*	deletes the *directory* including its subdirectories and files

Information Commands

apropos *subject*	shows commands related to *subject* (ex: **apropos floppy**) If it never finds anything, update db with "**mandb**"as root.
cal	displays a calendar (ex: "**cal 2004**" to see all of 2004)
date	prints or the system's date and time
df	lists the amount of diskspace used and available. Putting "." after the command lists the diskspace on the partition you are using right then. Use –h to convert bytes to KB, MB, GB.
dmesg	quick view of the boot log from the last system boot
du	gives the amount of diskspace taken by the directory. Try -h
free	gives the amount of memory available
history	displays the last commands that you issued at the prompt
hostname	prints the name of the machine used
id	prints information about the user/owner of shell
info *command*	similar to the **man** command but with hyperlinking available
last *username*	displays information about previous logins of *username*
ldd *program*	lists the shared libraries required by *program*
lsattr *file*	lists the file attributes of a given *file*
lsmod	lists all loaded kernel modules
lspci	lists all PCI devices
lsusb	lists all USB devices
man *-k* *command*	gives more information about different commands. **-k** gives the manuals that contain information about the keyword
ps	lists the active processes running in your account (try "ps -ef")
rpm –q *program*	tells you whether program is installed or not (and version)
sdiff *file1 file2*	compares *file1* to *file2* and prints the lines that differ
stty	shows and sets terminal characteristics
top	displays in real-time a list of running tasks
tty	prints the tty of a shell
uname	displays information about the operating system
uptime	displays the uptime of the machine and the load it carries
whatis *command*	summary of *command* (run *makewhatis* to build DB)
whoami (sometimes "**who am i**")	gives your username, tty and time the shell was started
xwininfo	print information on the window you click (X-Windows only)

Communication/Network Commands

chfn	change the info displayed about you when you are fingered
finger *userID*	displays login information about the *userID* indicated
ftp *host*	FTP client for connecting to *host*
ifconfig	display & configure network interface (like ipconfig)
host *ip/name*	performs DNS lookup on *IP* or *name*
ifup *devicename*	activates *devicename* (usually primary NIC card: eth0)
ifdown *devicename*	deactivates *devicename* (usually primary NIC card: eth0)
lynx	text based WWW browser. q exits the program
mesg *option*	sets your account to accept a talk requests (**y** or **n**). See **talk**
netstat	prints various network information (try netstat --inet -a)
nslookup *ip/name*	resolves the name (if *IP* is entered) or IP (if *name* is entered)
ping *host*	sends ping packet (ICMP echo request) to *host*
quota	displays disk usage and limits
route	show and manipulate IP routing tables. No switch shows info.
talk *userID*	initiates a talk request with *userID* (on the same system)
scp *user@host:/remote/file localdir*	downloads the file /remote/file from remote *host* through *user* and saves it locally to *localdir* (to upload a file: **scp localfile user@host:/remote/file**)
ssh *host*	ssh client for connecting to *host*
traceroute *host*	displays route taken by packets to reach to *host*
who	displays information on all users currently online
w	displays more information than "**who**" of online users
wget *URL*	downloads the file specified with *URL* (either http:// or ftp://). In order to mirror a *URL*: "**wget –m –t3 *URL* <-o logfile.log>**"
whois *domain*	gives domain registration information about *domain*

IO Commands

halt	shutdown the system
hdparm *<options>*	displays/sets hard drive parameters (try: **hdparm -i /dev/hda**)
lpq *-Pprinter*	lists the job line of a *Pprinter*
lpr *-Pprinter file.ps*	prints a ready postscript file. See enscript
lprm *-Pprinter job*	removes the printing *job* from *Pprinter* (**lpq** to get job #)
mke2fs */dev/fd0*	creates ext2/ext3 filesystem (format floppy: **mke2fs /dev/fd0**)
mount /dev/*<device>* *<mountpoint>*	mounts a *<device>* (ex: mount /dev/fd0 /mnt/floppy). No argument shows currently mounted devices.
reset	resets the terminal (use **clear** to simply do a clear screen)
shutdown	brings the system down (to reboot, try **shutdown -r now**)
umount *<mountpoint>*	unmounts a device (ex: umount /mnt/floppy)

Account Administration Commands

groupadd *group*	creates a group called *group*
groups *username*	list the group(s) that *username* belongs to
gpasswd –a –d *username group*	the **-a** switch adds *username* to *group*. The **–d** switch removes *username* from *group*
passwd	initiates the process for changing your password
su *<user>*	switch to a different *<user>*
sudo *command*	runs *command* with superuser access (to do this user first neesds to be listed in /etc/sudoers)
useradd	`useradd -c "normal user" -d /home/userid -g users\` `-G groupname1,groupname2 -s\ /bin/bash userid`
userdel –r *userid*	deletes *userid* (-r to delete home directory)

Other Commands

alias *aliasname='command'*	creates a temporary macro *aliasname* that executes *command*
chsh	changes the shell
clear	clears the screen
crontab	maintain crontab files for individual users
env *<VARIABLE=value>*	sets environmental *VARIABLE* to *value* (blank to show all)
kill *pid*	kills process *pid*. **kill –9** *pid* to force kill (use *ps* to list pid)
newgrp *group*	sets *group* as the primary group owner for new files created
nice *-n priority command*	executes the *command* with a specified *priority*
renice *priority pid*	changes a program's priority level afterwards (see **nice**, **ps**).
script *<scriptname>*	logs everything that you type to *<scriptname>*. "exit" to quit.
sleep *time*	makes the computer wait for *time*. Try "sleep 30; echo Go"
telinit *level*	specifies which runlevel to run at (0 = halt, 1 = single-user, 3 = full multi-user, 5 = X-windows, 6 = reboot)
umask *permission*	changes the default file permission for new files (ex: rw-r—r--). umask 077 makes all new files rw------. The permission that you set is subtracted from the default 666 permission (rw-rw-rw) so that umask 022 does 666 – 022 = 644 = rw-r--r--)

Shortcuts

[ctrl] + [alt] + [backspace]	restarts X-windows
[ctrl] + [alt] + F1 - F6	to switch to text terminal 1 - 6 (in X-windows)
[ctrl] + [alt] + F7	switch back to X-windows
[alt] + F1 - F7	switch to terminal 1-7 (when NOT logged in X-Windows)
[range] (format: [start-end])	access a range. For example "cp archive.zip.[0-9] <folder>" would copy archive.zip.0 thru archive.zip.9 to folder <folder>
./*command*	executes *command* (in current directory). Commands can be compounded by putting ";" in-between each command. They can also be set to the background by putting "&" at the end
[ctrl] + d	logout from the terminal
[ctrl] + z	stops (not cancel) program running in foreground. Enter "bg" <enter> to move program to the background, or "fg" for

	foreground
jobs	shows the status of your job(s)
%<#>	how to reference a job ("kill %2" kills job #2. "fb %3" moves job #3 back into the foreground)
~	your home folder (ex: "cd ~" sends you to /home/<username>)

Important Directories

/bin	holds the "essential" Linux commands and utilities
/boot	holds files required for boot process (kernel, vmlinuz, grub)
/dev	holds device files (hard drive, USB, CD-ROM, etc.)
/etc	holds system configuration files
/etc/init.d	holds scripts to start/stop network services
/etc/rc.d	holds system startup/shutdown scripts
/etc/X11	holds configuration files for X-windows
/home	holds user home directories (except for the root account)
/lib	holds system/shared library files
/lost+found	holds files restored after system crash
/mnt	used as temporary mount point for CD-ROM, floppy, etc.
/opt	typically where large software applications are installed
/proc	holds kernel and process information
/root	home directory for the root account (not /home/root)
/sbin	like /bin but holds commands used by root/admin users
/tmp	used as temporary file space
/usr	holds user-related programs and files
/usr/bin	holds commands/utilities that are outside of the "essential" list
/usr/lib	holds libraries for installed packages
/usr/local	holds files and data developed or customized on system
/usr/share/doc	holds Linux application documentation
/usr/share/man	holds Linux man pages
/var	holds files that "vary" in size (log files, spools)
/var/www	frequently used as the root directory for apache web servers
/var/log	holds many (but not all) log files

Important & Useful Files

/boot/grub/menu.lst	configuration file for GRUB
/etc/bashrc	global profile for all bash shell users on the system (ex: alias)
/etc/issue	name and version of Linux system shown at login prompt
/etc/fstab	filesystems that are automatically mounted on bootup
/etc/inittab	specifies runlevel and processes that are started at bootup
/etc/modules.conf	configuration of kernel modules that are loaded at bootup
/etc/nologin	if present it will refuse all non-root logins and display the contents of nologin file
/etc/profile	global profile for all users regardless of shell

/etc/sudoers	defines who has sudo access
/etc/syslog.conf	configuration for log files. Also see /etc/logrotate.conf
/etc/sysctl.conf	configuration file for system controls (ex: ignore ping)
/etc/X11/xorg.conf	configuration file for X-windows
/proc/cpuinfo	information about system CPU (try: **cat /proc/cpuinfo**)
/proc/meminfo	information about system memory (try: **cat /proc/meminfo**)