cheat_sheet.org

Last updated 8/14/2012
Best viewed in emacs org-mode.

* Command Reference:
** Basics:
*** Getting help:

View the manual for target command
man [command]

Get help with a target command (probably the same as above, but not always):
[command] -h

In case you forget the name of a command, print possible commands relating to [guess]:
apropos [guess]

View index of help pages:
info

*** Command Line Utilities:
**** Basic File and Directory Operations:
# Print current working directory:
pwd

Show files in current directory:
ls

Show maximum information about all files, including hidden:
ls -a

Recurse into subdirectories and list those as well:
ls -r

Move/rename a file or directory (be careful that you don't move the source over a destination with the same name):
mv [source] [destination]

Delete target forever (be very careful), use -r recursive flag for directories:

rm [target]

Copy file or directory:
cp [source] [destination]

Mount filesytem:
mount /dev/[device name] /media/[device name]

Unmount:
umount /media/[device name]

Forensically clone filesystems and do other low-level operations on files. Very dangerous:
dd

Work with filesystems and partitions. (Easier, still quite dangerous):
fdisk

**** System Administration:

# Execute command as an administrator (dangerous, but necessary for system administration tasks):
sudo [command]

# Become system administrator:
sudo -s

Quit system administration:
exit

# Check distro repositories for software updates:
sudo apt-get update

# Download and install updates (update first):
sudo apt-get upgrade

Search for package in the repositories:
apt-cache search [keyword]

Get more detail on one specific package:
apt-cache show [package name]

# Download and install a package:
sudo apt-get install [package name]

View the output of a command in a more convenient format:
[command] | less

**** Working With Files:

Print a file in terminal:
cat [file]

Find files matching [filename]:
locate [filename]

Search through [filename] for matches to [phrase]:
grep [phrase] [filename]

Search through output of a command for [phrase]:
[command] | grep [phrase]

**** Working With Processes:

List all running processes:
ps -e

Standard system monitor showing a more extensive view of all processes and system resources:
top

Like top, but with a better, cleaner interface:
htop

Stop a process from using all system resources and lagging computer:
nice [process name]

Kill misbehaving process (use sparingly, last resort, try 'nice' command first):
pkill [process name]

**** Compression and Encryption:

Make a simple compressed backup of a file or directory:
tar -cvzf [backup output.tgz] [target file or directory]

Open a compressed .tgz or .tar.gz file:
tar -xvf [target.tgz]

Encrypt a file:
gpg -o [outputfilename.gpg] -c [target file]

Decrypt a file:
gpg -o [outputfilename] -d [target.gpg]

Zip and encrypt a directory simultaneously:
gpg-zip -o encrypted-filename.tgz.gpg -c -s file-to-be-encrypted

*** The Bash shell:
**** File Name expansions:
# Current user's home directory:
~/

Current directory:
./

Parent directory:
../

Or even (Two parent directories down):
../../

All files in target directory. (Be very careful.):
/*

**** Output Redirects:

Redirect output of one command into the input of another with a pipe:
[command 1] | [command 2]

Or even:

[command 1] | [command 2] | [command 3]

Redirect output to a file:
[command] > file

Or:

[file] > [file]

Or even, to redirect in a different direction:
[file] < [file]

Append output rather than writing over the target file:

[file/command] >> [file]

Works like |, but it writes output to both target and terminal:
tee [target]

**** Controlling Execution:
# Wait until [command 1] is finished to execute [command 2]
[command 1] ; [command 2]

Or even:
[command 1] ; [command 2] ; [command 3]

**** Wildcards:
# Zero or more characters:
*

Matches "phrase" and any number of trailing characters:
phrase*

Matches any incidences of "phrase" with any trailing or leading chars:
phrase

Matches any one char:
?

Matches any of the characters listed inside brackets:
[chars]

Matches a range of chars between a-z:
[a-z]

** Advanced:
*** Command Line Utilities, Continued:
**** Networking:

Configure network interfaces:
ifconfig

Configure wireless network interfaces:
iwconfig

Connect to a remote server.
ssh [username]@[ipaddress]

Forward x from target to current machine (Get a remote desktop. Very obscure and very useful):
ssh -x [username]@[ipaddress]

Copy files over the network from one machine to another:
scp [source filename]:[username]@[ipaddress] [target filename]:[target username]@[target ipaddress]

Copy only changes between files or directories (super efficient way to sync directories, works either locally or with remote servers using username@ipaddress:optionalport, just like ssh):
rsync [source] [target]

Check to see if target is online and responding
ping [ip address]

View network route to target:
traceroute6 [ip address]

Network Monitor
netstat

Manage standard linux firewall (advanced users only)
iptables

Scan this machine to check for open ports:
nmap 127.0.0.1

***** netcat:

Listen for input from network on [recieving port], dump it to a file (possibly insecure):
netcat -l [recieving port] > file_copied

Pipe the output of a command to a target ip and port over the network:
[command] | netcat -w [number of seconds before timeout] [target ip] [target port]

Use tar to compress and output a file as a stream, pipe it to a target ip and port over the network:

sudo tar -czf - [filename] | netcat -w [number of seconds before timeout] [target ip] [target port]

**** Users and Groups:
# Change owner of a file or directory:
chown

Change privileges over file or directory:
chmod

Create a new user:
adduser

Change user privileges (be very careful with this one):
usermod

Delete user"
deluser

Print groups:
groups

Create a new group:
groupadd

Change group privileges:
groupmod

Delete group:
delgroup

Temporarily become a different user:
su [username]

Print usernames of logged in users:
users

Write one line to another user from your terminal:
talk

Interactive talk program to talk to other users from terminal:
ytalk

**** Working With Files, Continued:
# View what processes are using what files:
lsof

View the differences between two files:
diff [file 1] [file 2]

Output the top -n lines of [file]:
head -n [number of lines] [file]

Like head, but it outputs the last -n lines:
tail

Checksum a file:
md5sum [file]

Checksum every file in a directory:
md5deep [directory]

Checksum a file (safer algorithm with no hash collisions):
sha1sum

Same operation as md5deep, but using sha1:
sha1deep

Call [command] every -n seconds, and display output:
watch -n [number of seconds] [command]

Execute [command], print how long it took:
time [command]

View files in home from largest to smallest:
du -a ~/ | sort -n -r | less

remove spaces from filenames in current directory
rename -n 's/[\s]/"/g' *

change capitals to lowercase in filenames in current directory
rename 'y/A-Z/a-z/' *

***** Environment and Hardware:
# Print full date and time:
date

Print the hostname of this machine:
echo $HOSTNAME

Print information about current linux distro:
lsb_release -a

Print linux kernel version:
uname -a

Print information about kernel modules:
lsmod

Configure kernel modules (never do this):
modprobe

View Installed packages:
dpkg --get-selections

Print environment variables:
printenv

List hardware connected via PCI ports:
lspci

List hardware connected via USB ports:
lsusb

# Print hardware info stored in BIOS:
sudo dmidecode

Dump captured data off of wireless card:
dumpcap

Dump info about keyboard drivers:
dumpkeys

***** System Administration (Continued):

Add a Personal Package Archive from Ubuntu Launchpad:

add-apt-repository

# Install a .deb file from command line:
sudo dpkg -i package.deb
**** Python:

update pip (Python package manager):
pip install -U pip

search pip repos
pip

create a virtual python environment
virtualenv [dirname] --no-site-packages

connect to a virtual python environment
source [dirname]/bin/activate

disconnect from a python environment:
deactivate

install package into virtual python environment from outsie:
pip install [packagename]==[version_number] -E [dirname]

export python virtual environment into a shareable format:
pip freeze -E [dirname] > requirements.txt

import python virtual environment from a requirements.txt file:
pip install -E [dirname] -r requirements.txt

**** git (all commands must be performed in the same directory as .git folder):

Start a new git project:
git init

Clone a git (target can be specified either locally or remotely, via any number of protocols):
git clone [target]

Commit changes to a git:
git commit -m "[message]"

Get info on current repository:
git status

Show change log for current repository:

git log

Update git directory from another repository:
git pull [target]

Push branch to other repository:
git push [target]

Create a new branch:
git branch [branchname]

Switch to target branch:
git checkout [branchname]

Delete a branch:
git branch -d [branchname]

Merge two branches:
git merge [branchname] [branchname]

*** Virtualization:

#clone a virtual machine (this works, it's been tested):
vboxmanage clonehd [virtual machine name].vdi --format VDI ~/[target virtual machine name].vdi

#mount a shared virtual folder:
#you need to make sure you have the right kernel modules. You can do this with modprobe, but this package works instead in a ubuntu-specific way.

sudo apt-get install virtualbox-ose-guest-utils

sudo mount -t vboxsf [name of Shared folder specified in Virtualbox] [path of mountpoint]

*** mysql:

Get help:
help

Show databases:
show databases;

Choose a database to use:

```
use [database name here];

# Show database schema:
show tables;

# Delete database:
DROP DATABASE [databasename];

# New database:
CREATE DATABASE [databasename];

# Create a new user:
CREATE USER [username@localhost] IDENTIFIED BY '[password]' ;

# Show users:
select * from mysql.user;

# Delete a user:
delete from mysql.user WHERE User='[user_name]';

# Give user access to all tables (make them root). the "%" means that they can sign in remotely,
from any machine, not just localhost.:
grant all privileges on *.* to someusr@"%" identified by '[password]';

# give certain privileges to a user on a certain database:
grant select,insert,update,delete,create,drop on [somedb].* to [someusr]@["%"] identified by
'[password]';

# Tell mysql to use new user priv policies:
flush privileges;

# change user password:
use mysql;

update user set password='[password]'('[newpassword]') where User='[user_name]' ;

# mysql command line args:

# export text file with commands to rebuild all mysql tables:

mysqldump [databasename] > [dumpfilename.txt]

# restore from a dump:
```

```
mysql -u [username] -p < [dumpfilename.txt]

# dump entire database:

mysqldump -u [username] -p --opt [databasename] > [dumpfile.sql]

# restore from entire database dump:

mysql -u [username] -p --database=[databasename] < [dumpfile.sql]
```