Debian Reference Card

Software Maintenance, Part 1: apt

APT = Advanced Package Tool; can only be used by user „root“, the administrator. Apt itself uses the program „*dpkg*“, explained later on in 'Part 2' and 'Part 3'.
Apt uses online-repositories to retrieve programs from.

Commands:

apt-get	Operations involving downloads:	*install, delete, purge, update, upgrade, dist-upgrade*
apt-cache	Operations on apt's database:	*search*

root@debian:/home/user# apt-get install <program>
Install <program>

root@debian:/home/user# apt-get install -f
Try to fix (-f) a current problem by installing additional software that might resolve the issue

root@debian:/home/user# apt-get update
Update the apt-database

root@debian:/home/user# apt-get upgrade
Upgrade programs, installs maintenance and security fixes according to the current apt-database informations

root@debian:/home/user# apt-get dist-upgrade
Upgrades the core system components to the highest available versions according to the current apt-database informations

root@debian:/home/user# apt-get remove <program>
Deletes a program from the system, but leaves it's configuration files for future installs in place

root@debian:/home/user# apt-get remove -f
Try to fix (-f) a current problem by removing offending software, which might resolve the issue

root@debian:/home/user# apt-get purge <program>
Deletes a program and it's configuration files from the system

root@debian:/home/user# apt-cache search <string>
Searches the apt-database for the given string. The string can be part of a programs name or part of the description for a program

root@debian:/home/user# rm /var/lib/dpkg/lock
When an installation-process was interrupted, there might be a so-called „lock-file“ left. DPKG, the tool doing the actual installing in the background, creates a file in it's folder to tell itself that it is already busy and running. In case of a crash this file might not always be removed. Apt will complain about this file if it already exists. When you are sure there is no other apt or dpkg process running, you can remove / delete the file.

Alternative:
root@debian:/home/user# aptitude install <program>
There is an alternative tool to apt, named aptitude. It is used in the same way as apt, but is smarter solving problems. When encountering a problem in apt that it cannot solve, the first step is to let aptitude do the same:

For installation of whole program „blocks“ the standard tool is „tasksel“ which is also used during the system-installation:

Name	**Description / Function**	**Syntax on the Commandline**
tasksel	Select „tasks“ for installation	tasksel
	List available tasks	tasksel --list-tasks
	Install a specific task	tasksel install <task>
	Example:	tasksel install ssh-server

Software Maintenance, Part 2: manual install & repositories

Software not found in Apt's online-repositories can be installed in other ways:

a) Downloading the Debian-specific version and install it manually using *dpkg:*

root@debian:/home/user# dpkg -i <debian-package>

Skype is a good example. It can be downloaded for Debian explicitely.
The downloaded file ends in '.deb'. Debian distinguishes between 32bit (i386) and 64bit (amd64) versions of programs.
32bit programs can be installed on 64bit systems after the system was prepared accordingly for that. Look for that in 'Part 3'.
64bit software cannot be installed on a 32bit system.

b) Adding a repository to the list of repositories in */etc/apt/sources.list:*

root@debian:/home/user# nano /etc/apt/sources.list

In nano, behind every line ending with “main” we add “contrib non-free”, so that the resulting lines look similar to the following example-line:

deb http://ftp.us.debian.org/debian/ jessie main contrib non-free

This allows Debian to also use online-repositories which contain software which is under copyright, such as the Adobe Flashplayer.
There is an additional repository with special multimedia-software available.
You can add that at the end of the sources.list file:

deb http://www.deb-multimedia.org jessie main non-free

Add an empty line after this line.
Now we need to inform our Linux of the changes by telling the software manager to fetch / update the list of available software:

root@debian:/home/user# apt-get update

We'll get an error message, as the new repository we added in the last line uses an unknown security-key. So, we add that key by installing the repositories „keyring“:

root@debian:/home/user# apt-get install deb-multimedia-keyring

With the security-key now installed, we can re-do the updating of the repositories index:

root@debian:/home/user# apt-get update

This should finish without any errors now.

c) Using a general Linux (installer) version of the program, if available.

Many variations exist – most programs come with an explanation.
It boils down to these common variants:

root@debian:/home/user# bash<program.run or program.sh>

Bash executes programs. Files ending in '.run' are self-extracting archives, files ending in .sh are shell scripts. Shell scripts are sets of instructions for Bash that use system-commands to download and then install the actual program.

root@debian:/home/user# chmod +x <program.pl>
root@debian:/home/user# ./<program.pl>

Some installers come as a 'Perl-Script' file. Often, these perl-script files need to be made executable (chmod +x <program.pl>) and then started directly (./<program.pl>) as bash would realize it's a script, but doesn't know about perl-commands and thus fails executing the script.

Software Maintenance, Part 3: dpkg

DPKG = Debian PacKaGe ; dpkg does the actual work of unpacking, verifying , installing or removing programs. Debian packages end in '.deb', and might carry an identifier in their name when they are for a specific architecture such as 64bit PC-systems (amd64) or 32bit PC-systems (i386).
Example names for debian packages would be:

skype-debian_4.3.0.37-1_i386.deb (32Bit-PC Debian Package)

or

systemd_215-10_amd64.deb (64Bit-PC Debian Package)

or

perl-modules_5.20.1-5_all.deb (Architecture-agnostic / 'Universal' Debian Package)

We need to use dpkg only for four tasks (unless the crap really hit the fan so hard that the wings broke off):

a) Installing manually downloaded packages (removal is handled via *apt-get remove*)

root@debian:/home/user# dpkg -i <debian-package>

This installs a debian-package. The installed program can now be found with:

root@debian:/home/user# apt-cache search <first 3 to 6 letters of the package name>

With the 'program name' which apt-cache gives us – which is what apt uses, whilst dpkg uses filenames for installing – we now can remove or purge the installed program:

root@debian:/home/user# apt-get remove <program>

or:

root@debian:/home/user# apt-get purge <program>

b) Forcefully removing an installed program that causes problems for apt

root@debian:/home/user# dpkg -P --force-all <program>

-P = Purge ; --force-all applies 'all dirty tricks' toenforce the removal of the program.
This procedure might break things. Ideally one would do an '*apt-get install -f*' afterwards, to do a clean fresh install of the assumedly broken file, or an '*apt-get remove -f*' to remove programs that relied upon the defective program.

c) Configuring a 64bit system to allow installation of 32bit software

root@debian:/home/user/Downloads# dpkg --add-architecture i386

This tells Debian to activate 32Bit compatibility

root@debian:/home/user/Downloads# apt-get update

This tells Debian to update the list of available Software, now including the 32Bit software

root@debian:/home/user/Downloads# apt-get install -f

This installs any needed 32Bit Software. At the first time, that's usually no program at all.

d) Fixing a badly interrupted installation-process

Most of the times apt-get is able to handle an interrupted installation process. Sometimes, however, it is not, in which case it normally tells you on it's own that you should start dpkg:

root@debian:/home/user/Downloads# dpkg --configure -a

This will try to finish an interrupted installation-process. Apt will usually inform you when this step is neccessary.

Filesystem Commands:

„Invisible“ files are files which's name begins with a leading period.

Linux is case sensitive, cookie.txt is a completely different file than Cookie.TXT !
* = „any number of any symbol“ ; a „joker“ or „wildcard“ symbol
? = „one unknown symbol“ ; a „joker“
\ = „Escape Character“; means that a following special symbol (empty space, *) is to be treated as a character without any special meaning, or a normal character gets special meaning (\n = „new line“ symbol).
./ = this directory ../ = next higher directory ~/ = The current users home-directory (/home/<user>)

Name	Description / Function	Syntax on the Commandline
cd	Change into directory	cd <directory>
	Change into a higher directory	cd ..
pwd	Print Working Directory (where am I?)	pwd
whoami	Which user is logged in here ?	whoami
rm	remove / delete a file	rm <file>
	removing a directory (with contents)	rm -r <directory>
	removing all files in the current directory	rm *
	remove also all „invisible“ files	rm .*
mv	move a file	mv <path/to/source> <path/to/target>
	rename a file	mv <oldfile> <newfile>
cp	copy a file from A to B	cp <path/to/source> <path/to/target>
	copy a directory and all included files from A to B	cp -r <path/to/source> <path/to/target>
	Example: Copy all files and files inside folders from the Desktop to the current directory	cp -r ~/Desktop/* ./
ls	list directory contents	ls = ls </path/to/folder>
	list also „invisible“ files	ls -a
	list more informations	ls -l
mkdir	make directory	mkdir <newdirectory>
rmdir	remove an empty directory	rmdir <emptydirectory>
cfdisk	Configure Partitions & Format Disks	cfdisk <disk>
	Work on the standard primary HDD	cfdisk /dev/sda
	Work on a SD-Card in a Laptops cardreader	cfdisk /dev/mmcblk
chown	Change Owner and Group of a file	chown <user>:<group> <file or directory>
	Change Owner of all files in a directory	chown -R <user> <directory>
chmod	Change File Read/Write/Execute permissions	chmod <UGO> <file>
	U= User ; G= Group ; O = Other; octal values (from 0 to 7). The permissions for each „entity“ are calculated by adding : 4 (for reading (r)), 2 (for writing(w)) and 1 (for executing / access (x))	
	Examples: read, write, execute for all	chmod 777 <file>
	rwx for owner, rx for group and world	chmod 755 <file>
	For more options, see *man chmod*	
ln	Create a (symbolic is standard, needs '-s' option) link	ln -s <file> <link>

Process-Management Commands

Name	Description / Function	Syntax on the Commandline
ps	Lists processes in the order of their process-id (PID)	
	List processes running in this terminal	ps
	List all running processes	ps -A
top	Lists active processes, sorted by activity per default can also sort after used memory, runtime, … Kill a process pressing „k“, then entering the PID, and optionally the kill-signal (usually 9 or 15) Terminates upon pressing „q“ ; processes at >100% cpu-load are somewhat suspicious	top
kill	Kill a process based on it's PID	kill <PID>
	Kill a process with a special kill-signal	kill -<kill-signal, 9 or 15> <PID>
killall	Kill all process matching <name>	killall <name>
CTRL+C	Cancel/Kill the current command / process	
CTRL+Z	Send the current command / process to sleep	
bg	Send a sleeping process to run in the BackGround	bg
fg	Pull a sleeping process to run in the ForeGround	fg
&	Run a program in the background	<program> &
&&	Execute left command first, then right command...	<command1> && <command2>

Networking Tools

Name	Description / Function	Syntax on the Commandline
scp	Secure Copy; Copies local / remote files Copy a file locally as normal user to a system-folder	scp <file> root@localhost:/<path/to/target>
	Copy a file from a local file to a remote machine	scp <file> <remote-user>@<remote-pc-ip-or-name>:/<path/to/target>
	Example:	scp readme.txt smurf@192.168.0.2:/home/smurf/documentation/
	Copy a file from remote PC1 to the current folder	scp <remote-user>@<remote-pc-ip-or-name>:/<path/to/file>
ping	Measure the response time of another system	ping <ip-address>
	Send 4 pings to ip-address 192.168.0.5	ping -c4 192.168.0.5
whois	Request informations about a Server / Domain	whois <domain>
	Request data about the Domain google.de	whois google.de
traceroute	Trace the path of datapackets to a target	traceroute <ip or name>

Helpful Shell Commands

Name	Description / Function	Syntax on the Commandline
lsusb	List connected USB components	lsusb
lspci	List connected PCI / PCIe components	lspci
lscpu	List connected CPUs	lscpu
less	displays texts, allows navigation, terminates upon pressing „q“	less <textfile>
cat	dumps a file onto the screen P.S.: works on everything	cat <textfile> cat <imagefile>
reset	resets the terminal window back to defaults	reset
nano	Texteditor ; CTRL-O saves to file,	nano <textfile>
	CTRL-X quits	
|	„pipe“ output of the left command to the right command:	
	read a long directory listing comfortably	ls -l | less
>	put output of left command into right-side file	ls -l >index.txt
>>	append output of left command to right side file	whoami >>index.txt
locate	locate all instances of a file	locate <filename(part)>
updatedb	update locate's database (has to be done as root)	updatedb
grep	Search for a pattern/string (in a file) and display the line where the pattern occurs	grep <string> <file>
	Search the output of a command for a string	<command> | grep <string>
	Example: locate all files with 'dolphin' somewhere in their name in the /usr directory	locate dolphin | grep /usr/
screen	Opens a virtual terminal, optionally runs a command in a virtual terminal. You can disconnect from a virtual terminal pressing CTRL-A („Call Attention“) followed by pressing 'd' (detach).	
	Open a virtual terminal (press space to access it)	screen
	Run a program in a virtual terminal	screen top
	Reattach to a virtual terminal	screen -r
	Reattach when several virtual terminals are running:	screen -r <PID>
	Force detaching of an elsewhere attached virtual terminal (to attach oneself)	screen -d <PID>
man	Open up the manual of a program, 'q' quits	man <program>
du	Directory Usage (LOTS OF OUTPUT!)	du
	Directory Usage – Short summary	du -sch
exit	terminate current session / quit / exit / …	exit
su	Switch user. With no user given, switches to root	su <user>
passwd	Change password for current user	passwd
	As root: change <users> password	passwd <user>
find	find file [-iname <name>][-type <f|d>(file or directory)]	find </in/directory> -iname *<partofname>** -type f
crontab	configure automatically executed jobs (weekly backups)	crontab -e (starts editor ; more with *man crontab*)

User Management

/etc/group = Textfile which lists which groups contain which users. Groups are used to manage permissions of users. There one can see which groups the default user belongs to, and to which additional users should be added.

Name	**Description / Function**	**Syntax on the Commandline**
adduser	Add a user to the system or to a group Add user „hans“ to the system Add user „hans“ to the group „audio“ to be able to hear audio output	adduser <username> <group> adduser hans adduser hans audio
deluser	Delete a user from the system Delete a user from a group	deluser <username> deluser <username> <group>
addgroup	Add a group to the system	addgroup <groupname>
delgroup	Delete an (empty) group from the system	delgroup <groupname>

Group	**Description / Function**
root	Technically not a group, but the system administrator. There should be no members here.
dialout	This group is able to use old landline modems, but also general serial-port communication. This group remains neccessary especially for industrial applications and machines.
cdrom	read / write access to cd-rom and dvd drives.
sudo	Ability to use the sudo command.
audio	Ability to access and use audio-hardware.
www-data	The web-servers group (if installed). files and folders with this ownership can be read and written by the web-server.
plugdev	Pluggable Devices, namely USB devices, are accessible only to members of this group.
users	The traditional group in multi-users systems to add users to, to allow them sharing files. In a single-user system it's superfluous and thus normally empty.
scanner	Access to scanner hardware. To use USB scanners, membership in *plugdev* is needed, too.
netdev	Network Devices – Without membership in this group, users can't access LAN or WLAN.
bluetooth	Bluetooth – membership is needed to utilize bluetooth devices.
<username>	Users are basically just other groups. Logically, without members. Adding one user to another user is possible, however it'd violate the security idea behind groups.

Group-ID's

Group-ID's below 1000 are „System Groups“, i.e. groups intended to have members, and which do not have a homedirectory and cannot log into the system:

addgroup –system <system-group-name>

Group-ID's above 1000 are „User-Groups“, which can be both groups as well as users. More details can be found using *man addgroup* .

Encryption

Encrypt the home partition – Makes the home partition inaccessible for all non-authorized persons

1) Install cryptsetup:
root@debian:/home/user# apt-get install cryptsetup

2) Backup current /home contents and unmount the partition.
root@debian:/home/user# mkdir /root/backup
root@debian:/home/user# mv /home/ /root/backup/*
root@debian:/home/user# umount /home

3) Create encrypted LUKS partition:
root@debian:/home/user# cryptsetup luksFormat /dev/sda2 (replace sda2 with your partion name).

4) Open LUKS partition and map it to 'crhome' (this name can be arbitrary):
root@debian:/home/user# cryptsetup luksOpen /dev/sda2 crhome

5) Format encrypted partition, e.g.:
root@debian:/home/user# mkfs.ext4 /dev/mapper/crhome

6) Mount it:
root@debian:/home/user# mount /dev/mapper/crhome /home

7) Restore /home contents from the backup.
root@debian:/home/user# mv /root/backup/ /home/*

8) Recreate initrd:
root@debian:/home/user# update-initramfs -u

9) Create /etc/crypttab entry for encrypted volume:
root@debian:/home/user# nano /etc/crypttab
#
crhome /dev/sda2 none luks

10) Change /etc/fstab entry for /home, .e.g:
root@debian:/home/user# nano /etc/fstab

/dev/mapper/crhome /home ext4 defaults 0 2

11) Reboot!

Encrypt Linux SWAP (virtual memory) on Debian – Reduces an attackers chance of password-recovery /-reconstruction in case of theft

1) Prepare
root@debian:/home/user# apt-get install cryptsetup
Deactivate current swap
root@debian:/home/user# swapoff /dev/<swap-device>
Destroy old swap-structure
root@debian:/home/user# dd if=/dev/urandom of=/dev/<swap-device> *count=10240*

2) Edit the configuration, assuming a single swap-device, you want to add following line to /etc/crypttab:
root@debian:/home/user# swap1 /dev/<swap-device> */dev/urandom swap*
The encrypted device then will show up as /dev/mapper/swap1

3)Then add the swap to /etc/fstab, here is the line (don't forget to uncomment the old swap entry)
/dev/mapper/swap1 none swap sw 0 0

4) Activate: Now do a
root@debian:/home/user# /etc/init.d/cryptdisks reload
If the swaps are not already active, you need a
root@debian:/home/user# swapon -a

5) You can see the result with
root@debian:/home/user# dmsetup info
*root@debian:/home/user# cat /proc/swap*s

6) Reboot. After the reboot, check with
root@debian:/home/user# free
if you now have a working (and encrypted) swap.