



The

DEBIAN SYSTEM

CONCEPTS AND TECHNIQUES



Martin F. Krafft



NO STARCH
PRESS

Table of Contents

1	Introduction	17
1.1	About this book	19
1.2	Target audience	20
1.2.1	The Linux administrator	20
1.2.2	The Unix administrator	20
1.2.3	The Debian user	21
1.2.4	The Linux apprentice	22
1.3	How to use this book	22
1.4	Final notes	24
1.4.1	Conventions	24
1.4.2	Keeping up to date	24
1.4.3	An urgent plea for feedback	25
1.5	About the author	25
1.6	Acknowledgements	26
2	The Debian project in a nutshell	29
2.1	A history lesson	30
2.2	The Debian philosophy	34
2.2.1	Debian and its users	37
2.2.2	Free beer and free speech	39
2.2.3	Debian and the market	43
2.3	Licencing issues	44
2.4	The Debian community	46
2.4.1	Organisation of the project	46

2.4.2	Social aspects of the community	50
2.4.3	Social aspects of the group of developers	52
2.5	Helping the Debian project	57
2.5.1	Contributing to the project	58
2.5.2	Becoming a Debian developer	62
2.6	The Debian swirl	65
3	Installing Debian the right way	67
3.1	The Debian installer	68
3.1.1	Features of the new installer	69
3.1.2	System requirements	70
3.2	The <i>minimalistic approach to installation</i>	71
3.2.1	Installing the base system	72
3.2.2	Configuring the base system	94
3.3	Configuring the X server	96
3.3.1	An overview of X in Debian	97
3.3.2	<i>Integrating automatic hardware detection</i>	98
3.3.3	Dealing with unsupported hardware	99
3.3.4	Customising the X session	100
4	Debian releases and archives	103
4.1	Structure of the Debian archive	105
4.1.1	The package pool	106
4.1.2	Package indices	107
4.1.3	The Release files	108
4.2	The package upload	108
4.3	The official releases	110
4.3.1	The <i>unstable</i> release	110
4.3.2	The <i>testing</i> release	111
4.3.3	The <i>stable</i> release	113
4.4	Unofficial APT archives	114
4.4.1	The <i>experimental</i> archive	114
4.4.2	The <i>volatile</i> archive	115

4.4.3	The amd64 archive	116
4.4.4	The <i>*-proposed-updates</i> archives	116
4.4.5	The backports.org archive	116
4.4.6	The apt-get.org directory	118
4.4.7	Christian Marillat's multimedia archive	118
4.5	Architecture support	119
4.5.1	80386 – the processor	121
4.5.2	The amd64 architecture	122
4.5.3	Multi-arch	123
5	The Debian package management system	125
5.1	Requirements	125
5.2	Introducing Debian packages	128
5.2.1	Package categories	128
5.2.2	Package priorities	130
5.2.3	Anatomy of binary packages	131
5.2.4	The control files	134
5.3	Dealing with packages: dpkg	135
5.3.1	Handling binary packages	137
5.3.2	Installing packages	139
5.3.3	Configuration file handling	141
5.3.4	Interacting with the package database	144
5.3.5	Deinstalling packages	151
5.3.6	Overriding dpkg's sanity and policy checks	153
5.3.7	Dealing with errors in packages	156
5.3.8	dpkg configuration	159
5.3.9	dselect	159
5.4	Managing packages: APT	163
5.4.1	Specifying repositories	164
5.4.2	APT configuration	168
5.4.3	Installing packages	170
5.4.4	Searching the APT database	175
5.4.5	Inquiring about package dependencies	177

5.4.6	<i>Deinstalling and purging packages</i>	179
5.4.7	<i>Seamless upgrades</i>	181
5.4.8	<i>Enacting requests with APT</i>	184
5.4.9	<i>APT housekeeping</i>	186
5.4.10	<i>Resolving problems with APT</i>	186
5.4.11	<i>aptitude</i>	188
5.4.12	<i>synaptic</i>	197
5.5	<i>Debian tasks</i>	198
5.6	<i>Package management compared</i>	199
5.7	<i>Power from within: the Debian policy</i>	202
5.7.1	<i>The sacred configuration files</i>	205
5.7.2	<i>Mediating between packages</i>	207
5.7.3	<i>Package relations</i>	209
5.7.4	<i>The Filesystem Hierarchy Standard</i>	214
5.7.5	<i>Version numbers</i>	217
5.7.6	<i>Upgrading packages</i>	219
5.8	<i>debconf: configuration of Debian packages</i>	220
5.8.1	<i>An overview of debconf</i>	221
5.8.2	<i>Priority levels</i>	222
5.8.3	<i>debconf front-ends</i>	223
5.8.4	<i>Reconfiguring packages</i>	224
5.8.5	<i>debconf in action</i>	225
5.8.6	<i>Using a remote database back-end</i>	227
5.8.7	<i>Problems and shortcomings</i>	230
5.9	<i>Modifying packages</i>	232
5.9.1	<i>Recompiling packages</i>	232
5.9.2	<i>Repacking packages</i>	238
5.10	<i>Integrating non-Debian software</i>	239
5.10.1	<i>alien</i>	240
5.10.2	<i>checkinstall</i>	241
5.10.3	<i>equivs</i>	242
5.11	<i>Miscellaneous package tools</i>	243
5.11.1	<i>debsums</i>	243

5.11.2	apt-listchanges	244
5.11.3	apt-listbugs	246
5.11.4	cron-apt	246
5.11.5	deborphan	247
5.11.6	Keeping a clean system: debfoster	248
5.11.7	Caching APT archives	249
5.11.8	Mirroring the Debian archive: debmirror	253
5.11.9	Enhanced queries of the package database	254
5.11.10	Package popularity contest	256
5.11.11	Purposely omitted tools	257
5.12	Debian kernels	257
5.12.1	Kernel support	258
5.12.2	Anatomy of the kernel packages	259
5.12.3	Sources, headers, and documentation	266
5.12.4	Kernel modules and patches	267
6	Debian system administration	271
6.1	Fundamentals	272
6.1.1	Using directories instead of configuration files	272
6.1.2	Overriding permissions	274
6.1.3	Overriding files	275
6.1.4	The alternatives system	276
6.1.5	The Debian menu system	278
6.2	Users and authentication	281
6.2.1	System users and groups	281
6.2.2	User and group management	284
6.2.3	PAM — Pluggable Authentication Modules	290
6.3	System initialisation and automatic processes	291
6.3.1	The system initialisation process	292
6.3.2	Regular maintenance processes	305
6.4	Backups	307
6.5	Device management	309
6.5.1	discover and hotplug	309

6.5.2	kmod, the kernel autoloader	315
6.5.3	Loading modules during startup	316
6.6	Configuring kernel parameters	316
6.7	Log file management	317
6.7.1	Monitoring logs with logcheck	321
6.8	Network configuration management	323
6.8.1	Network configuration with ifupdown	323
6.8.2	Using DHCP to obtain a network address	338
6.8.3	Managing /etc/resolv.conf	339
6.8.4	Connectivity via PPP	342
6.8.5	Integrating PCMCIA network cards	349
6.8.6	Integrating wireless network interfaces	350
6.8.7	Miscellaneous network options	351
6.9	Administering inetd, the Internet superserver	352
6.10	Integrated management tools	354
6.10.1	wajig	354
6.10.2	feta	355
6.11	System administration resources	356
7	Security of the Debian system	357
7.1	Handling security problems	359
7.2	Security updates	363
7.3	Security out of the box	366
7.4	Package quality	368
7.5	Package integrity	369
7.5.1	Manual verification of package integrity	371
7.5.2	Secure APT	373
7.5.3	debsigs and dpkg-sig	377
8	Advanced concepts	381
8.1	Building kernel packages with make-kpkg	382
8.1.1	Using initial ramdisks	385
8.1.2	Patching the kernel	386

8.1.3	Compiling modules	388
8.1.4	Cross-compiling for other architectures	391
8.1.5	Symlink farming	393
8.1.6	Configuring make-kpkg	393
8.2	Mixing releases	394
8.2.1	Pinning releases with APT	395
8.2.2	Selecting target releases	401
8.2.3	Extending APT's internal cache	403
8.2.4	Mixing releases and security updates	404
8.2.5	aptitude and multiple releases	406
8.3	Alternative approaches to installing a Debian system	406
8.3.1	Bootstrapping an installation	407
8.3.2	Booting the installation from the network (PXE)	421
8.3.3	Customising the installer	423
8.3.4	Preseeding the installer	424
8.3.5	FAI: Fully automatic installations	426
9	Creating Debian packages	431
9.1	Manual packaging	432
9.2	Debianising with the package maintainer tools	436
9.2.1	A closer look at source packages	437
9.2.2	Investigating the upstream source tree	439
9.2.3	dh_make	441
9.2.4	Building source packages	444
9.2.5	Jumpstarting with dh_make	445
9.2.6	Writing debian/rules	449
9.2.7	Modifying the debian/* files	451
9.2.8	Creating the DEB file	455
9.2.9	Cleaning the source tree	458
9.2.10	Splitting and updating a package	460
9.2.11	The debhelper suite	465
9.2.12	The changes file	473
9.2.13	Verifying new packages	475

9.2.14	Signing the package files	477
9.2.15	Checking packages	478
9.2.16	Automating the package build	479
9.3	Local APT repositories	482
9.3.1	Anatomy of a personal repository	482
9.3.2	Upload tools	484
9.3.3	Automated repository management	486
9.4	Advanced package concepts	489
9.4.1	Package hook scripts	489
9.4.2	Using <i>debconf</i>	490
9.4.3	Library packages	498
9.5	Alternative build tools	501
9.5.1	<i>cdb</i> s	501
9.5.2	<i>yada</i>	503
9.6	Automating clean builds with <i>pbuilder</i>	504
9.6.1	Setting up a base tarball	505
9.6.2	Building packages with <i>pbuilder</i>	506
9.6.3	Using <i>pbuilder</i> to set up test systems	508
9.6.4	Mounting host directories inside the <i>chroot</i>	510
9.6.5	Modifying the tarball	510
10	Documentation and resources	513
10.1	Local documentation	514
10.2	Online resources	514
10.2.1	Official documentation and manuals	515
10.2.2	Semi-official resources	517
10.2.3	Unofficial resources	517
10.3	Printed resources	520
10.4	Discussion forums	521
10.4.1	Mailing lists	521
10.4.2	Web forums	527
10.4.3	IRC – Internet Relay Chat	528
10.5	Contacting people	530

10.6	The bug tracking system	532
10.6.1	Querying the BTS	533
10.6.2	Querying bugs from the command line	534
10.6.3	Bug severities	536
10.6.4	Bug tags	538
10.6.5	Reporting bugs	539
10.6.6	Mail traffic following a bug report	544
10.6.7	Interacting with the BTS	545
10.6.8	Bugs against pseudo-packages	548
10.6.9	Subscribing to a package's bug reports	552
10.6.10	Fixing bugs	553
Appendix		557
A	Debian flavours and other Debian-based operating systems	559
A.1	CDDs — Custom Debian Distributions	560
A.2	Debian derivatives	561
A.2.1	Knoppix	562
A.2.2	Ubuntu	563
A.2.3	Gnoppix	564
A.2.4	MEPIS	564
A.2.5	MNIS	565
A.2.6	Quantian	565
A.2.7	Skolelinux	565
A.2.8	Adamantix	566
A.2.9	SELinux	566
B	When is Debian the right choice?	569
B.1	You should run Debian if...	569
B.2	You should probably choose something else, if...	571
C	Miscellaneous	573
C.1	Important GPG keys related to Debian	573

Table of Contents

C.1.1	Official Debian archive signing keys	574
C.1.2	Other relevant signing keys	575
C.2	Setting up the filesystems	577
C.2.1	A sensible partition table	577
C.2.2	Supported filesystems	578
C.3	Extra packages	579
C.4	Configuring a local packet filter	580
C.5	Dual-booting with other operating systems	582
C.5.1	Chain-loading other bootloaders	583
C.5.2	Dealing with Windows peculiarities	583
D	The Debian Linux Manifesto	587
E	Debian Social Contract	591
E.1	The current Social Contract	592
E.2	The future Social Contract	593
F	The Debian Free Software Guidelines	595