

Kommentarer till utgåvan Debian GNU/Linux 4.0 (“etch”), IA-64

Josip Rodin, Bob Hilliard, Adam Di Carlo, Anne Bezemer, Rob Bradford, Frans Pop
(nuvarande), Andreas Barth (nuvarande), Javier Fernández-Sanguino Peña
(nuvarande), Steve Langasek (nuvarande)
<debian-doc@lists.debian.org>

\$Id: release-notes.sv.sgml,v 1.283 2007/04/07 06:59:57 dnylander Exp \$

Innehåll

1	Introduktion	1
1.1	Rapportera fel i det här dokumentet	1
1.2	Bidra med uppgraderingsrapporter	2
1.3	Källor för det här dokumentet	2
2	Vad är nytt i Debian GNU/Linux 4.0	3
2.1	Vad är nytt i distributionen?	4
2.1.1	Pakethantering	5
2.1.2	debian-volatile är nu en officiell tjänst	5
2.2	Systemförbättringar	6
2.3	Större kärnrelaterade ändringar	7
2.3.1	Ändringar i paketering av kärnan	7
2.3.2	Nya verktyg för att generera initrd-filer	8
2.3.3	Dynamisk hantering av /dev och hårdvaruidentifiering	8
3	Installationssystem	11
3.1	Vad är nytt i installationssystemet?	11
3.1.1	Stora ändringar	11
3.1.2	Automatiserad installation	13
3.2	Popularitetstävling	14
4	Uppgraderingar från tidigare utgåvor	15
4.1	Förberedelse inför uppgraderingen	15
4.1.1	Säkerhetskopiera all data eller konfigurationsinformation	15
4.1.2	Informera användarna i förväg	16

4.1.3	Förbered för återställning	16
4.1.4	Förbered en säker miljö för uppgraderingen	17
4.1.5	Stöd för 2.2-kärnor har uteslutits	17
4.2	Kontrollera systemstatus	17
4.2.1	Granska väntande åtgärder i pakethanteraren	18
4.2.2	Inaktivera APT-nålning	18
4.2.3	Kontrollera paketstatus	18
4.2.4	Inofficiella källor och bakåtporteringar	19
4.3	Avmarkera paket manuellt	20
4.4	Förbered källor för APT	20
4.4.1	Lägg till APT-källor på Internet	20
4.4.2	Lägg till APT-källor för en lokal spegelservr	21
4.4.3	Lägg till APT-källa från cd-rom eller dvd	21
4.5	Uppgradering av paket	22
4.5.1	Spela in sessionen	22
4.5.2	Uppdatering av paketlistan	23
4.5.3	Se till att du har tillräckligt med utrymme för uppgraderingen	23
4.5.4	Minimal systemuppgradering	24
4.5.5	Uppgradering av kärnan	26
4.5.6	Uppgradering av resten av systemet	27
4.5.7	Hämta paketsignaturer	27
4.5.8	Möjliga problem under uppgraderingen	28
4.6	Uppgradering av din kärna och relaterade paket	29
4.6.1	Installera metapaketet för kärnan	29
4.6.2	Uppgradering från en 2.6-kärna	30
4.6.3	Uppgradering från en 2.4-kärna	30
4.6.4	Ny ordning för enhetsnumrering	31
4.6.5	Ny ordning för serieenheter	32
4.6.6	Tidsproblem vid uppstart	32
4.7	Saker att göra före omstart	33
4.7.1	Konvertering från devfs	33

4.7.2	Uppgradering av mdadm	33
4.8	Förberedelse inför nästa utgåva	33
4.9	Föråldrade paket	34
4.9.1	Dummy-paket	35
5	Eventuella problemsituationer för etch	37
5.1	Tänkbara problem	37
5.1.1	Problem med enheter relaterade till udev	37
5.1.2	Vissa program kanske inte längre fungerar med en 2.4-kärna	37
5.1.3	Vissa nätverksplatser kan inte nås med TCP	37
5.1.4	Långsammare uppdatering av APT:s paketindexfiler	38
5.1.5	Asynkron nätverksinitiering kan orsaka oförutsett beteende	38
5.1.6	Problem vid användning av WPA-säkrade trådlösa nätverk	38
5.1.7	Problem med icke-ASCII-tecken i filnamn	39
5.2	Uppgradering till en 2.6-kärna	39
5.2.1	Tangentbordskonfiguration	40
5.2.2	Muskonfiguration	40
5.2.3	Ljudkonfiguration	40
5.3	Övergång från XFree86 till X.Org	40
5.4	Inget stöd för 8-bitars display i många program	41
5.5	Uppgradering från exim till exim4	41
5.6	Uppgradering av apache2	42
5.7	Uppgradering av Zope och Plone	42
5.8	Jokerteckensexpantering (globbing) med GNU tar	43
5.9	NIS och Network Manager	43
5.10	Utfasade osäkra php-konfigurationer	43
5.11	Säkerhetsstatus för Mozilla-produkter	44
5.12	KDE-skrivbordet	44
5.13	Ändringar och stöd för GNOME-skrivbordet	44
5.14	Standardredigerare	45

6 Mer information om Debian GNU/Linux	47
6.1 Ytterligare läsning	47
6.2 Få hjälp	47
6.2.1 Sändlistor	47
6.2.2 Internet Relay Chat	48
6.3 Rapportera fel	48
6.4 Att bidra till Debian	48
A Hantera ditt sarge-system	51
A.1 Uppgradering av ditt sarge-system	51
A.2 Kontrollera dina källistor	51

Kapitel 1

Introduktion

De primära målen för dessa Kommentarer till utgåvan är att informera användare om större ändringar i Debian GNU/Linux-distributionen för att tillhandahålla information om hur man på ett säkert sätt uppgraderar från tidigare utgåva till den aktuella utgåvan och till slut att informera användare om kända tänkbara problem som användare kan påträffa vid uppgradering till eller vid användning av etch-utgåvan.

Observera att det är omöjligt att lista alla kända problem och därför har ett urval gjorts baserat på en kombination av den allmänna förekomsten och problemets inverkan.

Den senaste versionen av det här dokumentet finns alltid tillgängligt på <http://www.debian.org/releases/stable/releasenotes>. Om din version är äldre än en månad¹, kanske du vill hämta ner den senaste versionen.

Observera att vi endast ger stöd för och dokumenterar uppgraderingen från den tidigare utgåvan av Debian (i det här fallet, uppgradering från sarge). Om du behöver uppgradera från äldre utgåvor föreslår vi att du läser tidigare versioner av kommentarerna för utgåvan och uppgraderar till sarge först.

1.1 Rapportera fel i det här dokumentet

Vi har försökt att testa alla de olika uppgraderingsstegen som beskrivs i det här dokumentet och vi har även försökt att förutse alla de tänkbara problem som kan inträffa för våra användare.

Oavsett, om du tror att du har hittat fel i den här dokumentationen (felaktig information eller att information saknas), skicka in en felrapport via felrapporteringsystemet (<http://bugs.debian.org/>) mot paketet `release-notes`.

¹som listas på förstasidan av PDF-versionen och i sidfoten på HTML-versionen

1.2 Bidra med uppgraderingsrapporter

Vi välkomnar all information från användare som relaterar till uppgraderingar från sarge till etch. Om du vill ge oss information kan du skicka in en felrapport via felrapporteringsystemet (<http://bugs.debian.org/>) mot paketet `upgrade-reports` med dina erfarenheter.

Tillhandahåll följande information när du skickar in din uppgraderingsrapport:

- Status för din paketdatabas före och efter uppgraderingen: `dpkg:s` statusdatabas tillgänglig i `/var/lib/dpkg/status` och `aptitude:s` pakettillståndsinformation, tillgänglig i `/var/lib/aptitude/pkgstates`. Du bör även ha gjort en säkerhetskopia innan uppgraderingen, som beskrivs i 'Säkerhetskopiera all data eller konfigurationsinformation' på sidan 15, men du kan även hitta säkerhetskopior av den här informationen i `/var/backups`.
- Sessionsloggar med `script`, som beskrivs i 'Spela in sessionen' på sidan 22.
- Dina `aptitude`-loggar, tillgängliga i `/var/log/aptitude`.

Observera: du bör ta dig tid att granska och ta bort eventuellt känslig och/eller konfidentiell information från loggfilerna innan de inkluderas i en felrapport eftersom informationen kommer att publiceras i en publik databas.

1.3 Källor för det här dokumentet

Det här dokumentet genereras med hjälp av `debiandoc-sgml`. Källorna för Kommentarer till utgåvan finns tillgängliga i cvs-förrådet för *Debian Documentation Project*. Du kan använda webbgränssnittet (<http://cvs.debian.org/ddp/manuals.sgml/release-notes/?root=debian-doc>) för att komma åt dess filer individuellt via webben och se dess ändringar. För mer information om hur man kommer åt cvs-förrådet, konsultera cvs-sidorna för Debian Documentation Project (<http://www.debian.org/doc/cvs>).

Kapitel 2

Vad är nytt i Debian GNU/Linux 4.0

Den här utgåvan innehåller officiellt stöd för AMD64-arkitekturen vilken ger stöd för 64-bitars processorer från både Intel (EM64T) och AMD (AMD64). För den tidigare utgåvan, Debian GNU/Linux 3.1 ("sarge"), fanns en inofficiell version tillgänglig av den här porteringen.

Officiellt stöd för arkitekturen Motorola 680x0 ("m68k") har uteslutits på grund av att det inte mötte kriterierna som ställts av Debians utgåveansvariga. De viktigaste underliggande anledningarna var prestanda och begränsat stöd för viktiga komponenter. Dock förväntas m68k-porteringen att kvarstå som aktiv och finnas tillgänglig för installation även om den inte är en del av den här officiella stabila utgåvan.

Följande arkitekturer stöds officiellt av Debian GNU/Linux etch:

- Intel x86 ("i386")
- Alpha ("alpha")
- SPARC ("sparc")
- PowerPC ("powerpc")
- ARM ("arm")
- MIPS ("mips" (big-endian) och "mipsel" (little-endian))
- Intel Itanium ("ia64")
- HP PA-RISC ("hppa")
- S/390 ("s390")
- AMD64 ("amd64")

Du kan läsa mer om porteringsstatus och porteringsspecifik information för din arkitektur på Debians webbsidor för porteringar (<http://www.debian.org/ports/ia64/>).

2.1 Vad är nytt i distributionen?

Den nya utgåvan av Debian kommer med ännu fler programvaror än dess föregångare sarge; distributionen inkluderar över 6500 nya paket och innehåller totalt 18200 paket. Större delen av programvaran i distributionen har uppdaterats: över 10700 programvarupaket (det är 68 % av alla paket i sarge). Ett stort antal paket (över 3500, 23 % av paketen i sarge) har av olika anledningar tagits bort från distributionen. Du kommer inte att se några uppdateringar för dessa paket och de kommer att markeras som "föråldrade" i din pakethanterare.

Den här utgåvan av Debian GNU/Linux byter från XFree86 till 7.1-utgåvan av X.Org, vilken inkluderar stöd för ett större urval av hårdvara och förbättrad automatisk identifiering. Det här tillåter användningen av Compiz, som är en av de första fönsterhanterarna för kompositering för X Window System vilket erbjuder fullständig OpenGL-accelereringen i hårdvaran för de enheter som stöds.

Debian GNU/Linux skickar med flera olika skrivbordsprogram och miljöer. Bland annat inkluderar skrivbordsmiljöerna GNOME 2.14¹, KDE 3.5.5a och Xfce 4.4. Produktivhetsprogrammen har även uppdaterats, inklusive kontorssviterna OpenOffice.org 2.0.4a och KOffice 1.6 såväl som GnuCash 2.0.5, Gnumeric 1.6.3 och Abiword 2.4.6.

Uppdateringar av andra skrivbordsprogram inkluderar uppgradering till Evolution 2.6.3 och Gaim 2.0. Mozilla-sviten har även uppdaterats och huvudprogrammen har bytt namn: iceweasel (version 2.0.0.2) är webbläsaren Firefox och icedove (version 1.5) är e-postklienten Thunderbird.

Den här utgåvan inkluderar även, bland annat, följande programvaruuppdateringar:

- GNU C-biblioteket, version 2.3.6
- GNU Compiler Collection 4.1 som standardkompilator
- språktolkare: PHP 5.2, Python 2.4
- serverprogramvara:
 - e-postserver: Exim 4.63 (standard-e-postserver för nya installationer), Postfix 2.3, Courier 0.53, Cyrus 2.2
 - webbservrar: Apache 2.2, fnord 1.10
 - databasservrar: MySQL 5.0.32, PostgreSQL 8.1
 - OpenSSH-servern, version 4.3
 - namnservrar: Bind 9.3, maradns 1.2
 - katalogserver: OpenLDAP 2.3

Den officiella Debian GNU/Linux-distributionen levereras på 19 till 23 binär-cd-skivor (beroende på arkitektur) och ett liknande antal källkods-cd-skivor. En dvd-version av distributionen finns även tillgänglig.

¹med några moduler från GNOME 2.16

2.1.1 Pakethantering

Aptitude är det föredragna programmet för pakethanteringen i konsollen. *Aptitude* har stöd för de flesta kommandoradsåtgärder som *apt-get* har och har visats sig vara bättre på att lösa beroendeproblem än *apt-get*. Om du fortfarande använder *dselect*, bör du byta till *aptitude* som det officiella verktyget för pakethantering.

För *etch* har en avancerad konfliktlösarmekanism implementerats i *aptitude* som försöker att hitta den bästa lösningen om konflikter uppstår på grund av ändringar i beroenden mellan olika paket.

Secure APT finns nu tillgänglig i *etch*. Den här funktionen utökar säkerheten för Debian GNU/Linux-system helt enkelt genom att erbjuda stöd för stark kryptografi och digitala signaturer för att validera hämtade paket. Den här utgåvan inkluderar verktyget *apt-key* för att lägga till nya nycklar till *apts* nyckelring, vilken som standard endast inkluderar den aktuella signeringsnyckeln för Debian-arkivet och som tillhandahålls av paketet *debian-archive-keyring*.

I dess standardkonfiguration kommer nu *apt* att varna om paket hämtas från källor som inte har autentiserats. Framtida utgåvor kan tvinga att alla paket måste verifieras innan de kan hämtas. Administratörer av inofficiella *apt*-förråd uppmuntras att generera en kryptografisk nyckel och signera sina Release-filer, såväl som att tillhandahålla ett säkert sätt att distribuera sina publik nycklar.

För mer information kan du läsa *apt* (8), kapitlet Package signing in Debian (<http://www.debian.org/doc/manuals/securing-debian-howto/ch7#s-deb-pack-sign>) i *Securing Debian Manual* och Debians wiki (<http://wiki.debian.org/SecureApt>).

En annan funktion som lades till i *apt* är möjligheten att endast hämta ändringarna i *Packages*-filerna sedan din senaste uppdatering. Mer om den här funktionen går att läsa i 'Långsammare uppdatering av APT:s paketindexfiler' på sidan 38.

2.1.2 debian-volatile är nu en officiell tjänst

Tjänsten *debian-volatile* som introducerades som en inofficiell tjänst med utgåvan av *sarge*, har nu blivit en officiell Debian GNU/Linux-tjänst.

Det här betyder att den nu använder en *.debian.org*-adress². Tänk på att uppdatera din */etc/apt/sources.list* på lämpligt sätt om du redan använder den här tjänsten.

debian-volatile tillåter användare att enkelt uppdatera stabila paket som innehåller information som lätt blir utdaterad. Exempel är signaturlistorna för en antivirusprodukt eller mönsteruppsättningen för ett skräppostfiler. För mer information och en lista över spegelservrar, se arkivets webbsida (<http://volatile.debian.org/>).

²Den gamla adressen *volatile.debian.net* kommer även att vara giltig framöver.

2.2 Systemförbättringar

Det har skett ett antal ändringar i distributionen som nya installationer av etch har nytta av men som inte automatiskt kan tillämpas på uppgraderingar från sarge. Det här avsnittet ger en överblick för de mest relevanta ändringarna.

Prioritet för grundläggande utvecklingspaket har sänkts Ett antal utvecklingspaket som brukade ha prioriteten *standard* har nu prioriteten *optional*, vilket betyder att de inte längre kommer att installeras som standard. Det inkluderar standard-C/C++-kompilatorn `gcc`, såväl som annan programvara (`dpkg-dev`, `flex`, `make`) och utvecklingshuvudfiler (`libc6-dev`, `linux-kernel-headers`).

Om du vill få dessa paket installerade på ditt system är det enklaste sättet att installera paketet `build-essential` som kommer att dra in de flesta av dessa paket.

SELinux-prioriteten är standard men inte aktiverad som standard Paketet som behövs för SELinux-stödet har promoverats till prioriteten *standard*. Det betyder att de kommer att installeras som standard under nya installationer. För befintliga system kan du installera SELinux genom att köra:

```
# aptitude install selinux-basics
```

Observera att SELinux-stödet *inte* är aktiverat som standard. Information om hur man konfigurerar och aktiverar SELinux kan hittas på Debians wiki (<http://wiki.debian.org/SELinux>).

Ny inet-superdemon som standard Standard-inet-superdemonen för etch är `openbsd-inetd` istället för `netkit-inetd`. Den kommer inte att startas om inga tjänster har konfigurerats, vilket som standard är sant. Den nya standarddemonen kommer att installeras automatiskt vid uppgradering.

Standard vi-klon ändrad vi-klonen som installeras som standard är nu en kompakt version av `vim` (`vim-tiny`) istället för `nvi`.

Ändringar i standardfunktioner för ext2/ext3 Nya ext2- och ext3-filsystem kommer att skapas med funktionerna `dir_index` och `resize_inode` aktiverat som standard. Den första funktionen ger högre hastighet för åtgärder på kataloger som innehåller många filer; den andra gör det möjligt att ändra storlek på ett filsystem under drift (exempelvis när det monteras).

Användare som uppgraderar från sarge kan överväga att lägga till flaggan `dir_index` manuellt genom att använda `tune2fs`³; flaggan `resize_inode` kan inte läggas till på ett befintligt filsystem. Det är möjligt att kontrollera vilka flaggor som är inställda för ett filsystem med hjälp av `dumpe2fs -h`.

³Flaggan `filetype` bör redan vara inställd på de flesta filsystem, förutom möjligen på system som installerats före sarge.

Standardteckenkodning för etch är UTF-8 Standardteckenkodningen för nya installationer av Debian GNU/Linux är UTF-8. Ett antal program kommer även att konfigureras till att använda UTF-8 som standard.

Användare som uppgraderar till etch och önskar att byta till UTF-8 kommer att behöva konfigurera om sina miljöer och lokaldefinitioner. Den systembreda standarden kan ändras genom att köra `dpkg-reconfigure locales`; välj först en UTF-8-lokal för ditt språk och land och ställ sedan in den som standard. Observera att byte till UTF-8 betyder att du antagligen även behöver konvertera befintliga filer från din tidigare teckenkodning till UTF-8.

Paketet `utf8-migration-tool` innehåller ett verktyg kan hjälpa till i migreringen men paketet finns endast tillgängligt i unstable eftersom det inte blev färdigt i tid för etch. Det rekommenderas starkt att du skapar en säkerhetskopia av ditt data och konfiguration innan du använder verktyget.

Observera att vissa program kanske ännu inte fungerar korrekt i en UTF-8-miljö, mestadels på grund av visningsproblem.

Debians wiki (<http://wiki.debian.org/Sarge2EtchUpgrade>) innehåller ytterligare information om ändringar mellan sarge och etch.

2.3 Större kärnrelaterade ändringar

Debian GNU/Linux 4.0 skickas med kärnversionen 2.6.18 för alla arkitekturer; utgåvan är fortfarande mestadels ⁴ kompatibel med 2.4-kärnor, men Debian tillhandahåller inte längre 2.4-kärnpaket eller stöd för dem.

Det har skett stora ändringar både i själva kärnan och i paketeringen av kärnan för Debian. Några av dessa ändringar gör uppgraderingsprocessen komplicerad och kan möjligen resultera i problem vid omstart av systemet efter uppgraderingen till etch. Det här avsnittet ger en överblick för de mest viktiga ändringarna; möjliga problem och information om hur man kommer runt dessa problem inkluderas i senare kapitel.

Om du fortfarande använder en 2.4-kärna bör du läsa 'Uppgradering till en 2.6-kärna' på sidan [39](#) noga.

2.3.1 Ändringar i paketering av kärnan

Kärnpaketerna har bytt namn Alla Linux-kärnpaket har bytt namn från `kernel-*` till `linux-*` för att rensa upp i namnrymden. Det gör det enklare att inkludera icke-Linux-kärnor i Debian i framtiden.

⁴Vissa individuella paket kanske inte längre fungerar korrekt med en 2.4-kärna; se 'Vissa program kanske inte längre fungerar med en 2.4-kärna' på sidan [37](#).

Standardkärnor innehåller SMP-förmågor Flerprocessorsystem kräver inte längre en `*-smp`-variant av Linuxkärnan. För IA-64 har `linux-image`-paketen utan ändelsen `-smp` stöd för både en- och flerprocessorsystem.

Där det är möjligt kommer dummyspaket för övergången att vara beroenden av de nya paketen som tillhandahålls för de utelämnade paketen.

2.3.2 Nya verktyg för att generera `initrd`-filer

Debianpaketen för kärnavbilder för IA-64 kräver en `initrd` för uppstart av systemet. På grund av ändringar i kärnan kan inte längre verktyget som användes för att generera `initrd`-filer i `sarge`, `initrd-tools`, användas och har utfasats. Två nya verktyg har utvecklats som ersätter det: `initramfs-tools` och `yaird`. Koncepten bakom de nya verktygen är mycket olika; en överblick finns tillgänglig på Debians wiki (<http://wiki.debian.org/InitrdReplacementOptions>). Båda kommer att generera en `initrd` som använder `initramfs`-filsystemet, vilket är ett komprimerat `cpio`-arkiv. Standard och rekommenderat verktyg är `initramfs-tools`.

Uppgradering till en `etch`-kärna kommer att innebära att `initramfs-tools` blir installerat som standard. Om du uppgraderar från en 2.4-kärna till en av Debians 2.6-kärnor måste du använda `initramfs-tools`. Användning av `yaird` kommer att orsaka att installationer av `linux-image-2.6` misslyckas om du kör en 2.2- eller 2.4-kärna.

Paketet `initrd-tools` inkluderas fortfarande i `etch` på grund av att det behövs för uppgraderingar från `sarge`. Det kommer att utelämnas i nästa utgåva.

2.3.3 Dynamisk hantering av `/dev` och hårdvaruidentifiering

`etch`-kärnor tillhandahåller inte längre stöd för `devfs`.

Ersättningen för `devfs` är `udev`, en implementation av `devfs` i användarrymden.

`udev` monteras över katalogen `/dev` och befolkar den katalogen med enheter som stöds av kärnan. Den kommer även dynamiskt att lägga till och ta bort enheter som kärnmoduler läser in respektive tar bort, baserat på händelser som genereras av kärnan. `udev` är mycket mer mångsidig än `devfs` och erbjuder tjänster som används av andra paket som till exempel `hal` (hardware abstraction layer).

I kombination med kärnan tar även `udev` hand om identifiering av hårdvara och inläsning av moduler för identifierade enheter. På grund av att det här står det i konflikt med `hotplug`. I `sarge` kunde `discover` också användas för att läsa in moduler under uppstartsprocessen, men dess nya version i `etch` tillhandahåller inte längre den funktionen. `discover` används fortfarande av X.Org för att identifiera vilket grafikkort som finns i systemet.

Om du installerar en Debian-kärnavbild kommer `udev` att installeras som standard eftersom `initramfs-tools` är beroende av det.

Du kan undvika att installera `udev` genom att bygga en anpassad ickemodulär kärna eller genom att använda en alternativ `initrd`-generator, såsom `yaird`. Dock rekommenderas `initramfs-tools` som `initrd`-generator.

Kapitel 3

Installationssystem

Debian-installer är det officiella installationssystemet för Debian. Den erbjuder en uppsjö av installationsmetoder. Vilka metoder som finns tillgängliga för att installera ditt system beror på din arkitektur.

Avbilder av installeraren för etch kan hittas tillsammans med Installationsguiden på Debians webbplats (<http://www.debian.org/releases/stable/debian-installer/>).

Installationsguiden finns även inkluderad på den första cd-/dvd-skivan i de officiella Debian cd-/dvd-uppsättningarna:

```
/doc/install/manual/sv/index.html
```

Du kanske även vill se errata (<http://www.debian.org/releases/stable/debian-installer/index#errata>) för debian-installer för en lista över kända problem.

3.1 Vad är nytt i installationssystemet?

Det har skett en hel del utveckling med Debian Installer sedan dess första officiella utgivning med sarge som resulterat i både förbättrat hårdvarustöd och ett antal nya spännande funktioner.

I dessa Kommentarer till utgåvan kommer vi endast att lista de större ändringarna i installeraren. Om du är intresserad i en översikt över detaljerade ändringar sedan sarge, se utgåveinformationen för beta- och RC-utgåvorna av etch, tillgängliga från nyhetshistorik (<http://www.debian.org/devel/debian-installer/News/>) för Debian-installeraren.

3.1.1 Stora ändringar

Ingen omstart under installationen Tidigare brukade installationen vara uppdelad i två delar: ställa in grundsystemet och göra det startbart, följt av en omstart och sedan uppstar-

ten av `base-config` som tog hand om saker som användarkonfigurering, konfigurering av pakethanteringssystem och installation av ytterligare paket (med hjälp av `tasksel`).

För `etch` har det andra steget integrerats i själva Debian Installer. Det här har ett antal fördelar, inklusive ökad säkerhet och att, efter omstarten på slutet av installationen, det nya systemet redan bör ha den korrekta tidszonen och, om du installerade skrivbordsmiljön, kommer att starta det grafiska användargränssnittet.

Teckenkodningen UTF-8 är standard för nya system Installeraren kommer att konfigurera system till att använda teckenkodningen UTF-8 istället för de gamla språkspecifika teckenkodningarna (t.ex. ISO-8859-1, EUC-JP eller KIO-8).

Mer flexibel partitionering Det är nu möjligt att konfigurera filsystem på en LVM-volym med hjälp av guidad partitionering.

Installeraren kan även konfigurera krypterade filsystem. Med manuell partitionering har du möjligheten att välja mellan `dm-crypt` och `loop-aes` med en lösenfras eller en slumpmässig nyckel och du kan finjustera olika alternativ. Med guidad partitionering kommer installeraren att skapa en krypterad LVM-partition som innehåller alla andra filsystem (förutom `/boot`) som logiska volymer.

Räddningsläge Du kan använda installeraren att lösa problem med ditt system, till exempel när det vägrar att starta upp. De första stegen kommer att vara precis som en vanlig installation, men installeraren kommer inte att starta partitioneraren. Istället kommer den att erbjuda dig en meny med räddningsalternativ.

Aktivera räddningsläget genom att starta upp installeraren med `rescue`, eller genom att lägga till uppstartsparametern `rescue/enable=true`.

Använder sudo istället för root-konto Under expertinstallationer kan du välja att inte ställa in root-kontot (det kommer att vara låst), utan istället konfigurera `sudo` så att den första användaren kan använda det för systemadministration.

Kryptografisk verifiering av hämtade paket Paket som hämtas med installeraren kommer numera att kontrolleras kryptografiskt med hjälp av `apt`, vilket gör det svårare att compromettera ett system som blir installerat över nätverket.

Förenklad e-postkonfigurering Om "standardsystem" installeras kommer installeraren att ställa in en grundläggande konfiguration för systemets e-postserver som endast kommer att tillhandahålla lokala e-postleveranser. E-postservern kommer inte vara tillgänglig för andra system anslutna till samma nätverk. Om du vill konfigurera ditt system till att hantera e-post som inte bara är lokal för systemet (antingen för att skicka eller ta emot e-post), behöver du konfigurera om e-postsystemet efter installationen.

Val av skrivbordsmiljö Installationssystemet kommer att installera ett GNOME-skrivbord som standardskrivbord om användaren frågar efter ett.

Användare som önskar installera andra skrivbordsmiljöer kan enkelt göra det genom att lägga till uppstartsparametrar: `tasks="standard, kde-desktop"` för KDE och `tasks="standard, xfce-desktop"` för Xfce. Observera att det inte kommer att fungera när du installerar från en fullständig cd-avbild utan att använda en nätverkspegel

som en ytterligare paketkälla; det kommer att fungera när du använder en dvd-avbild eller någon annan installationsmetod.

Det finns även separata cd-avbilder tillgängliga som installerar skrivbordsmiljöerna KDE eller Xfce som standard.

Nya språk Tack vare massiva insatser från översättarna kan nu Debian installeras på 47 olika språk med det textbaserade installationsgränssnittet. Det är sex språk fler än i sarge. Språken som är nya i den här utgåvan är vitryska, esperanto, estniska, kurdiska, makedonska, tagalog, vietnamesiska och wolof. På grund av brist på uppdaterade översättningar har två språk utgått i den här utgåvan: persiska och kymriska.

Användare som inte önskar använda någon lokalanpassning kan nu välja C som sin föredragna lokal i installerarens språkval. Mer information om språken finns tillgänglig i språklistan för d-i (<http://d-i.aliOTH.debian.org/i18n-doc/languages.html>).

Förenklad lokalanpassning och inställning av tidszon Konfiguration av språk, länder och tidszoner har förenklats för att minimera mängden information som behövs från användaren. Installeraren kommer nu att gissa systemets språk och tidszon baserat på språkvalet eller kommer att tillhandahålla en begränsad lista om den inte kan gissa. Användarna kan fortfarande välja olika kombinationer om det behövs.

Förbättrad systembred lokalanpassning Den mesta av internationalisering- och lokalanpassningsfunktionerna som tidigare hanterades av verktyget `localization-config` är nu inkluderade i Debian-installeraren eller i själva paketen. Det betyder att val av ett språk automatiskt kommer att installera de paket som behövs för det språket (ordböcker, dokumentation, typsnitt...) i både standard- och skrivbordsmiljöer. Konfiguration som inte längre hanteras automatiskt är bland annat konfiguration av pappersstorlek och vissa avancerade tangentbordsinställningar för X Windows för vissa språk.

Observera att språkspecifika paket endast kommer att installeras automatiskt om de finns tillgängliga under installationen.

3.1.2 Automatiserad installation

En hel del av ändringarna som nämndes i föregående avsnitt, antyder även ändringar i installerarens stöd för automatiserade installationer med förkonfigurationsfiler. Det här betyder att om du har befintliga förkonfigurationsfiler som fungerade med installeraren för sarge, kan du inte förvänta dig att dessa fungerar med den nya installeraren utan ändringar.

Bra nyheter är att Installationsguiden (<http://www.debian.org/releases/stable/installmanual>) nu har en separat bilaga med omfattande dokumentation om hur man använder förkonfiguration.

Installeraren för etch introducerar några nya spännande funktioner som tillåter enklare och utförligare automatisering av installationer. Den lägger även till stöd för avancerad partitionering med RAID, LVM och krypterad LVM. Se dokumentationen för detaljer.

3.2 Popularitetstävling

Installationssystemet kommer än en gång att erbjuda installation av paketet `popularity-contest`. Det här paketet installerades inte som standard i sarge men det blev installerat i äldre utgåvor.

`popularity-contest` tillhandahåller värdefull information till Debian-projektet om vilka paket i distributionen som faktiskt används. Den här informationen används huvudsakligen för att bestämma ordningen i vilken paket inkluderas på installationsskivorna, men används också ofta av Debians utvecklare för att bestämma huruvida ett paket som saknar en paketansvarig ska adopteras eller inte.

Informationen från `popularity-contest` behandlas anonymt. Vi skulle uppskatta om du vill delta i den officiella kartläggningen; du kan på så sätt hjälpa till att förbättra Debian.

Kapitel 4

Uppgraderingar från tidigare utgåvor

4.1 Förberedelse inför uppgraderingen

Vi föreslår att du läser informationen i 'Eventuella problemsituationer för etch' på sidan [37](#) innan du uppgraderar. Det kapitlet täcker in möjliga problem som inte direkt relaterar till uppgraderingsprocessen men som fortfarande kan vara viktigt att känna till innan du påbörjar arbetet.

4.1.1 Säkerhetskopiera all data eller konfigurationsinformation

Innan uppgradering av ditt system rekommenderas det starkt att du gör en fullständig säkerhetskopiera, eller åtminstone en säkerhetskopiera av data eller konfigurationsinformation som du inte vill riskera att förlora. Uppgraderingsverktygen och -processen är ganska tillförlitliga men ett hårdvarufel mitt i en uppgradering kan resultera i ett allvarligt skadat system.

De huvudsakliga delar du vill säkerhetskopiera är innehållet i `/etc`, `/var/lib/dpkg` och utdata från `dpkg --get-selections "*" (citationstecknen är viktiga).`

Själva uppgraderingsprocessen ändrar ingenting i katalogen `/home`. Dock är det känt att vissa program (exempelvis delar av Mozilla-sviten och skrivbordsmiljöerna GNOME och KDE) skriver över befintliga användarinställningar med nya standardvärden när en ny version av programmet startas för första gången av en användare. Som en försiktighetsåtgärd bör du göra en säkerhetskopiera av de dolda filerna och katalogerna (så kallade "punktfiler") i användarnas hemkataloger. Den säkerhetskopieringen kan hjälpa till att återställa eller återskapa de gamla inställningarna. Du kanske även vill informera dina användare om det här.

Alla paketinstallationsåtgärder måste köras med superanvändarens rättigheter, så logga antingen in som root eller använd `su` eller `sudo` för att få de nödvändiga åtkomsträttigheterna.

Uppgraderingen innebär att vissa förutsättningar måste mötas; du bör kontrollera dem innan den faktiska uppgraderingen påbörjas.

4.1.2 Informera användarna i förväg

Det är klokt att informera alla användare i förväg angående de uppgraderingar som du planerar att göra, även om användarna som kommer åt ditt system via en `ssh`-anslutning knappt kommer att märka det under uppgraderingen, och bör kunna fortsätta att arbeta som vanligt.

Om du vill vidta extra försiktighetsåtgärder bör du säkerhetskopiera eller avmontera användarnas partitioner (`/home`) före uppgradering.

Du kommer antagligen behöva göra en kärnuppgradering vid uppgradering till `etch`, så en omstart kommer vanligtvis vara nödvändig. Vanligtvis kommer den att göras efter att uppgraderingen är färdig.

4.1.3 Förbered för återställning

På grund av många ändringar i kärnan mellan `sarge` och `etch` gällande drivrutiner, identifiering av hårdvara och namnstandarderna och ordning på enhetsfiler, finns det en risk att du kan uppleva problem vid omstart av ditt system efter uppgraderingen. En del kända tänkbara problem finns dokumenterade i det här och de nästkommande kapitlen av dessa Kommentarer till utgåvan.

Av den anledningen är det klokt att försäkra sig om att du kan återställa om ditt system skulle misslyckas att starta om eller, för fjärrhanterade system, misslyckas att komma åt nätverket.

Om du fjärruppgraderar via en `ssh`-länk är det starkt rekommenderat att du vidtar nödvändiga säkerhetsåtgärder för att kunna komma åt servern genom en fjärrserieterminal. Det finns en chans att, efter uppgraderingen av kärnan och omstart, vissa enheter kommer att få nya namn (som beskrivs i 'Ny ordning för enhetsnumrering' på sidan 31) och du kommer att behöva rätta till systemkonfigurationen genom en lokal konsoll. Om systemet av misstag startas om mitt i en uppgradering finns det en chans att du behöver återställa systemet med hjälp av en lokal konsoll.

Det självklara är att först försöka starta om med din gamla kärna. Dock, av olika anledningar dokumenterade någon annanstans i det här dokumentet, är det inte garanterat att det fungerar.

Om det misslyckas behöver du ett alternativt sätt att starta upp ditt system på så att du kan komma åt och reparera det. Ett alternativ är att använda en speciell räddningsavbild eller en levande Linux-skiva. Efter att du har startat upp från en sådan skicka bör du kunna montera ditt rotfilssystem och använda `chroot` in i det för att undersöka och rätta till problemet.

Ett annat alternativ som vi rekommenderar är att använda *räddningsläget* i Debian-installeraren för `etch`. Fördelen av att använda installeraren är att du kan välja bland dess många installationsmetoder för att hitta en som bäst passar din situation. För mer information, konsultera avsnittet "Återställning av ett trasigt system" i kapitel 8 av Installationsguiden (<http://www.debian.org/releases/stable/installmanual>) och Debian Installer FAQ (<http://wiki.debian.org/DebianInstaller/FAQ>).

Felsökningsskal under uppstart med hjälp av `initrd`

Paketet `initramfs-tools` inkluderar ett felsökningsskal¹ i de `initrd`-filer som den genererar. Om, till exempel, `initrd`-filen inte kan montera ditt rotfilssystem, kommer du att bli förflyttad in i det här felsökningsskalet vilket har grundläggande kommandon tillgängliga för att hjälpa att spåra problemet och möjligen rätta till det.

Grundläggande saker att kontrollera är: närvaron av korrekta enhetsfiler i `/dev`; vilka moduler som läses in (`cat /proc/modules`); utdata för `dmesg` efter fel vid inläsning av drivrutiner. Utdata för `dmesg` kommer även att visa vilka enhetsfiler som har tilldelats till vilka diskar; du bör kontrollera det här mot utdata för `echo $ROOT` för att försäkra dig om att rotfilssystemet finns på den förväntade enheten.

Om du lyckas rätta till problemet, skriv `exit` för att avsluta felsökningsskalet och fortsätta uppstartsprocessen där felet inträffade. Så klart behöver du även rätta till det underliggande problemet och generera om `initrd`-filen så att nästa uppstart inte misslyckas igen.

4.1.4 Förbered en säker miljö för uppgraderingen

Uppgradering av distributionen bör göras antingen lokalt från en virtuell textkonsoll (eller en direktansluten serieterminal), eller från ett fjärrsystem via en `ssh`-anslutning.

För att öka säkerhetsmarginalen vid en fjärruppgradering föreslår vi att du kör uppgraderingsprocesser i den virtuella konsollen som tillhandahålls av programmet `screen`, vilket gör att man säkert kan återansluta till sessionen och försäkra sig om att uppgraderingsprocessen inte avbryts även om fjärranslutningen avbryts.

Viktigt! Du bör *inte* uppgradera via `telnet`, `rlogin`, `rsh`, eller från en X-session som hanteras av `xdm`, `gdm` eller `kdm` etc, på maskinen som du uppgraderar. Det är på grund av att de processer som hanterar dessa tjänster kan avslutas under uppgraderingen, vilket kan resultera i ett *oåtkomligt* system som endast är halvt uppgraderat.

4.1.5 Stöd för 2.2-kärnor har uteslutits

Om du kör en kärnversion lägre än 2.4.1 behöver du uppgradera till (åtminstone) 2.4-serien innan uppgradering av `glibc`. Det bör helst göras innan uppgraderingen påbörjas. Det rekommenderas att du uppgraderar direkt till 2.6.8-kärnan tillgänglig i `sarge`, istället för att uppgradera till en 2.4-kärna.

4.2 Kontrollera systemstatus

Uppgraderingsprocessen som beskrivs i det här kapitlet har designats för uppgraderingar från "rena" `sarge`-system utan tredjepartspaket. Speciellt finns det kända problem med tredjeparts-

¹Den här funktionen kan inaktiveras genom att lägga till parametern `panic=0` till dina uppstartparametrar.

paket som installerar program under `/usr/X11R6/bin/` och orsakar problem med uppgraderingar på grund av övergången till X.org ('Övergång från XFree86 till X.Org' på sidan 40). För största möjliga tillförlitlighet för uppgraderingsprocessen bör du ta bort tredjepartspaket från ditt system innan du påbörjar uppgraderingen.

Den här proceduren antar även att ditt system har uppdaterats till den senaste punktutgåvan av sarge. Om du inte har gjort det här eller är osäker, följ instruktionerna i 'Uppgradering av ditt sarge-system' på sidan 51.

4.2.1 Granska väntande åtgärder i pakethanteraren

I vissa fall kan användningen av `apt-get` för installation av paket istället för med `aptitude` kan göra att `aptitude` anser ett paket som "unused" och schemalägga det för borttagning. I allmänhet bör du se till att systemet är fullständigt uppdaterat och "rent" innan du fortsätter med uppgraderingen.

På grund av det här bör du granska om det finns några väntande åtgärder i pakethanteraren `aptitude`. Om ett paket är schemalagt för borttagning eller uppdatering i pakethanteraren kan det påverka uppgraderingsprocessen negativt. Observera att det endast är möjligt att rätta till det här om din `sources.list` fortfarande pekar på *sarge*; och inte till *stable* eller *etch*; se 'Kontrollera dina källistor' på sidan 51.

För att göra det här måste du köra `aptitude` och trycka på "g" ("Gå"). Om det visar några åtgärder bör du granska dem och antingen rätta till dem eller implementera de föreslagna åtgärderna. Om inga åtgärder föreslås kommer du att se ett meddelande som säger "Inga paket är schemalagda för installation, borttagning eller uppgradering".

4.2.2 Inaktivera APT-nålning

Om du har konfigurerat APT att installera vissa paket från en annan distribution än den stabila (exempelvis från "testing"), kan du ändra din konfiguration för paketnålning i APT (lagrad i `/etc/apt/preferences`) för att tillåta uppgraderingen av paket till versionerna i den nya stabila utgåvan. Ytterligare information om APT-nålning kan hittas i `apt_preferences(5)`.

4.2.3 Kontrollera paketstatus

Oavsett vilken metod som används för uppgradering, rekommenderas det att du kontrollerar statusen på paketen först och verifierar att alla paket är möjliga att uppgradera. Följande kommando kommer att visa de paket som har statusen Half-Installed eller Failed-Config, och de som har någon form av felstatus.

```
# dpkg --audit
```

Du kan även inspektera tillståndet för alla paket på ditt system med `dselect`, `aptitude`, eller med kommandon som

```
# dpkg -l | pager
```

eller

```
# dpkg --get-selections "*" > ~/nuvarande-paket.txt
```

Det är önskvärt att ta bort eventuella tillbakahållna paket innan uppgradering. Om något paket är systemkritiskt och hålls tillbaka för uppgraderingen, kommer uppgraderingen att misslyckas.

Observera att `aptitude` använder en annan metod för att registrera paket som hålls tillbaka än `apt-get` och `dselect`. Du kan identifiera paket som hålls tillbaka med `aptitude` med

```
# aptitude search "~ahold" | grep "^h"
```

Om du vill kontrollera vilka paket du hade hållt tillbaka för `apt-get`, ska du använda

```
# dpkg --get-selections | grep hold
```

Om du ändrat och byggt om ett paket lokalt, och inte bytte namn på det eller la in ett datum i versionen, måste du hålla tillbaka det för att förhindra att det uppgraderas.

Pakettillståndet "hold" för `aptitude` kan ändras med:

```
# aptitude hold paketnamn
```

Ersätt `hold` med `unhold` för att ändra "hold"-tillståndet.

Om det är någonting du behöver rätta till, är det bäst att se till att din `sources.list` fortfarande refererar till sarge vilket förklaras i 'Kontrollera dina källistor' på sidan 51.

4.2.4 Inofficiella källor och bakåtporteringar

Om du har några icke-Debianpaket på ditt system, bör du tänka på att dessa kan tas bort under uppgraderingen på grund av beroendekonflikter. Om dessa paket blev installerade genom att lägga till extra paketarkiv i din `/etc/apt/sources.list`, bör du kontrollera om det arkivet även erbjuder paket som är byggda för `etch` och ändra källraden på lämpligt sätt samtidigt som dina källrader för Debian-paket.

Vissa användare kan ha inofficiella bakåtporterade versioner av paket som är "nyare" än de som finns i Debian installerade på sina sarge-system. Sådana paket kommer med stor sannolikhet att orsaka problem under en uppgradering eftersom de kan resultera i filkonflikter². Avsnittet 'Möjliga problem under uppgraderingen' på sidan 28 innehåller information om hur man hanterar filkonflikter om de skulle inträffa.

²Debian's pakethanteringsystem tillåter vanligtvis inte att ett paket tar bort eller ersätter en fil som ägs av ett annat paket såvida det inte har definierats att ersätta det paketet.

4.3 Avmarkera paket manuellt

För att förhindra att `aptitude` tar bort vissa paket som installerades som beroenden för andra paket behöver du manuellt avmarkera dem som *auto*-paket. Det inkluderar OpenOffice och Vim för skrivbordsinstallationer:

```
# aptitude unmarkauto openoffice.org vim
```

Och 2.6-kärnavbilder om du har installerat dem med ett kärnmetapaketer:

```
# aptitude unmarkauto $(dpkg-query -W 'kernel-image-2.6.*' | cut -f1)
```

Observera: Du kan granska vilka paket som är markerade som *auto* i `aptitude` genom att köra:

```
# aptitude search 'i~M <paketnamn>'
```

4.4 Förbered källor för APT

Innan du påbörjar uppgraderingen måste du redigera konfigurationsfilen för paketlistor i `apt`, `/etc/apt/sources.list`.

`apt` kommer att anse att alla paket som kan hittas via någon "deb"-rad, och installera paketet med högsta versionsnumret, där prioritet ges till de förstnämnda raderna (på så sätt, om det är så att flera speglar, skulle du vanligtvis först namnge en lokal hårddisk, sedan cd-skivor, och sedan HTTP/FTP-speglar).

En utgåva kan ofta refereras till av både dess kodnamn (t.ex. `sarge`, `etch`) och efter dess statusnamn (alltså `oldstable`, `stable`, `testing`, `unstable`). Att referera till en utgåva efter dess kodnamn har fördelen att du aldrig blir överraskad av en ny utgåva och av den anledningen används den här metoden här. Det kan naturligtvis betyda att du själv måste hålla utkik efter nya utgåvor. Om du istället använder statusnamnet, kommer du bara att se massor av uppdateringar för paket som finns tillgängliga så snart en utgivning har skett.

4.4.1 Lägg till APT-källor på Internet

Standardkonfigurationen är inställd för installation från Debians huvudservrar på Internet, men du kanske önskar ändra `/etc/apt/sources.list` till att använda andra speglar, föredragsvis en spegel som är nätverksmässigt närmare dig.

Adresserna till Debians HTTP- eller FTP-speglar kan hittas på <http://www.debian.org/distrib/ftplist> (se avsnittet "Listan över Debianspeglar"). HTTP-speglar är vanligtvis snabbare än FTP-speglar.

Till exempel, anta att din närmaste Debian-spegel är `http://mirrors.kernel.org/debian/`. När den spegeln inspekteras med en webbläsare eller FTP-program, kommer du att märka att huvudkatalogerna är organiserade så här:

```
http://mirrors.kernel.org/debian/dists/etch/main/binary-ia64/...
http://mirrors.kernel.org/debian/dists/etch/contrib/binary-ia64/...
```

Lägg till den här raden till din `sources.list` för att använda den här spegelservern med `apt`:

```
deb http://mirrors.kernel.org/debian etch main contrib
```

Observera att "dists" läggs till automatiskt, och argumenten efter utgåvans namn används för att utöka sökvägen till flera kataloger.

Efter att du har lagt till dina nya källor ska du inaktivera de tidigare befintliga "deb"-raderna i `sources.list` genom att placera ett hash-tecken (#) framför dem.

4.4.2 Lägg till APT-källor för en lokal spegelserver

Istället för att använda HTTP- eller FTP-paketspeglar, kanske du önskar ändra `/etc/apt/sources.list` till att använda en spegel på en lokal hårddisk (möjligen monterad över NFS).

Till exempel, din paketspegel kan finnas under `/var/ftp/debian/`, och innehålla huvudkataloger som dessa:

```
/var/ftp/debian/dists/etch/main/binary-ia64/...
/var/ftp/debian/dists/etch/contrib/binary-ia64/...
```

Lägg till den här raden till din `sources.list` för att använda den här med `apt`:

```
deb file:/var/ftp/debian etch main contrib
```

Observera att "dists" läggs till automatiskt, och argumenten efter utgåvans namn används för att utöka sökvägen till flera kataloger.

Efter att du har lagt till dina nya källor ska du inaktivera de tidigare befintliga "deb"-raderna i `sources.list` genom att placera ett hash-tecken (#) framför dem.

4.4.3 Lägg till APT-källa från cd-rom eller dvd

Om du *endast* vill använda cd-skivor, kommentera ut de befintliga "deb"-raderna i `/etc/apt/sources.list` genom att placera ett hash-tecken (#) framför dem.

Se till att det finns en rad i `/etc/fstab` som aktiverar montering av din cd-rom-enhet på monteringspunkten `/cdrom` (den exakta monteringspunkten `/cdrom` krävs för `apt-cdrom`). Till exempel, om `/dev/hdc` är din cd-rom-enhet, ska `/etc/fstab` innehålla en rad som denna:

```
/dev/hdc /cdrom auto defaults,noauto,ro 0 0
```

Observera att det *inte får finnas några blanksteg* mellan orden `defaults,noauto,ro` i det fjärde fältet.

För att verifiera att det fungerar, mata in en cd och försök köra

```
# mount /cdrom      # det här monterar cd-skivan på monteringspunkten
# ls -alF /cdrom    # det här ska visa cd-skivans rotkatalog
# umount /cdrom     # det här kommer att avmontera cd-skivan
```

Kör sedan:

```
# apt-cdrom add
```

för varje Debian Binary cd-rom som du har, för att lägga till data om varje cd till APT:s databas.

4.5 Uppgradering av paket

Det rekommenderade sättet att uppgradera mellan utgåvor av Debian GNU/Linux är att använda pakethanteringsverktyget `aptitude`. Det här programmet gör säkrare val angående paketinstallationer än att köra `apt-get` direkt.

Glöm inte att montera alla nödvändiga partitioner (speciellt rot- och `/usr`-partitionerna) läs- och skrivbara, med ett kommando som det här:

```
# mount -o remount,rw /monteringspunkt
```

Efter det ska du dubbelkontrollera att källraderna för APT (i `/etc/apt/sources.list`) refererar antingen till `"etch"` eller till `"stable"`. Det ska inte finnas några källrader som pekar till `sarge`. Observera: källrader för en cd-skiva kommer ofta att referera till `"unstable"`, även om det här är konstigt ska du *inte* ändra dem.

4.5.1 Spela in sessionen

Det rekommenderas starkt att du använder programmet `/usr/bin/script` för att spela in en utskrift av uppgraderingssessionen. Om problem uppstår har du en logg på vad som hände och, om det behövs, kan tillhandahålla exakt information i en felrapport. För att påbörja inspelningen, kör:

```
# script -t 2>>~/uppgradering-till-etch.time -a ~/uppgradering-till-etch.scrip
```

eller liknande. Lägg inte typescript-filen i en temporär katalog såsom `/tmp` eller `/var/tmp` (filer i dessa kataloger kan tas bort under uppgraderingen eller under en omstart).

Typescript kommer även att låta dig granska informationen som har rullat ut från skärmen. Växla helt enkelt till VT2 (med `Alt-F2`) och, efter inloggning, använd `less -R ~root/uppgradering-till-etch.script` för att visa filen.

Efter att du har färdigställt uppgraderingen, kan du stoppa script genom att ange `exit` vid prompten.

Om du har använt flaggan `-t` för `script` kan du använda programmet `scriptreplay` för att spela upp hela sessionen:

```
# scriptreplay ~/uppgradering-till-etch.time ~/uppgradering-till-etch.script
```

4.5.2 Uppdatering av paketlistan

Först behöver listan över tillgängliga paket för den nya utgåvan hämtas. Det görs genom att köra:

```
# aptitude update
```

Körning av det här första gången nya källor uppdateras kommer att skriva ut några varningar som relaterar till tillgängligheten för källorna. Dessa varningar är harmlösa och kommer inte att visas om du kör kommandot igen.

4.5.3 Se till att du har tillräckligt med utrymme för uppgraderingen

Du måste kontrollera att ditt system har tillräckligt mycket ledigt hårddiskutrymme innan du påbörjar en fullständig systemuppgradering, som beskrivs i 'Uppgradering av resten av systemet' på sidan 27. Alla paket som behöver hämtas för installation kommer att hämtas från nätverket och lagras i `/var/cache/apt/archives` (och underkatalogen `partial/` under hämtningen) så du måste se till att du har tillräckligt utrymme på filsystemspartitionen som innehåller `/var/` för temporär hämtning av paketen som ska installeras på ditt system. Efter hämtningen kommer du antagligen behöva mer utrymme på de andra filsystemspartitionerna för att både installera de uppgraderade paketen (som kan innehålla större binärfiler eller mer data) och de nya paketen som kommer att inkluderas i uppgraderingen. Om ditt system inte har tillräckligt med utrymme kan det resultera i en ofullständig uppgradering som kan vara svår att rätta till.

Både `aptitude` och `apt` kommer att visa dig detaljerad information om det diskutrymme som behövs för installationen. Du kan se det här estimatet innan den faktiska uppgraderingen påbörjas genom att köra:

```
# aptitude -y -s -f --with-recommends dist-upgrade
[ ... ]
XXX paket uppgraderade, XXX nyinstallerade. XXX att ta bort och XXX inte uppg
```

Om du inte har tillräckligt med utrymme kan du se till att frigöra utrymme innan uppgraderingen. Du kan till exempel:

- Ta bort paket som tidigare har hämtats ner för installation (i `/var/cache/apt/archive`). Rensa upp paketcachen genom att köra `apt-get clean` eller `aptitude clean` vilket kommer att ta bort alla tidigare hämtade paketfiler.
- Ta bort gamla paket som du inte längre använder. Om du har paketet `popularity-contest` installerat kan du använda `popcon-largest-unused` för att lista de paket som du inte använder i systemet och som tar upp mest utrymme. Du kan även använda `deborphan` eller `debfooster` för att hitta föråldrade paket (se 'Föråldrade paket' på sidan 34). Alternativt kan du starta `aptitude` i "visuellt läge" och hitta föråldrade paket under "Föråldrade och lokalt skapade paket".
- Ta bort paket som tar upp för mycket utrymme och som du inte har ett omedelbart behov av (du kan alltid installera om dem efter uppgraderingen). Du kan lista paketen som tar upp mest utrymme genom att köra `dpigs` (tillgängligt i paketet `debian-goodies`) eller med `wajig` (kör `wajig size`).
- Flytta systemloggar från `/var/log/` till ett annat system, eller ta bort permanent.

Observera att för att ta bort paket på ett säkert sätt, rekommenderas det att växla tillbaka din `sources.list` till `sarge` vilket förklaras i 'Kontrollera dina källistor' på sidan 51.

4.5.4 Minimal systemuppgradering

På grund av vissa nödvändiga paketkonflikter mellan `sarge` och `etch` kommer en direkt körning av `aptitude dist-upgrade` ofta att ta bort ett stort antal paket som du vill behålla. Vi rekommenderar därför en tvådelad uppgraderingsprocess, först en minimal uppgradering för att komma runt dessa konflikter och sedan en fullständig `dist-upgrade`.

Kör först:

```
# aptitude upgrade
```

Det här har effekten av att uppgradera de paket som kan uppgraderas utan att kräva att några andra paket tas bort eller installeras.

Följ upp den minimala uppgraderingen med:

```
# aptitude install initrd-tools
```

Det här steget kommer automatiskt att uppgradera `libc6` och `locales` och kommer att dra in stödbibliotek för SELinux (`libselinux1`). Vid det här tillfället kommer vissa körande tjänster

att startas om, inklusive `xdm`, `gdm` och `kdm`. Ett resultat av detta är att lokala X11-sessioner kommer att brytas.

Nästa steg beror på uppsättningen paket som du har installerade. Dessa kommentarer till utgåvan ger allmänna råd om vilken metod som ska användas, men om du är osäker, är det rekommenderat att du undersöker de paketborttagningar som föreslås av varje metod innan du fortsätter.

Några vanliga paket som förväntas att bli borttagna är bland annat `base-config`, `hotplug`, `xlibs`, `netkit-inetd`, `python2.3`, `xfree86-common` och `xserver-common`. För en mer komplett lista över föråldrade paket i `etch`, se 'Föråldrade paket' på sidan [34](#).

Uppgradering av ett skrivbordssystem

Dessa uppgraderingssteg har verifierats att fungera på `sarge`-system med funktionen `desktop` installerad. Det är antagligen den metod som kommer att ge bäst resultat på system med funktionen `desktop` installerad, eller med paketen `gnome` eller `kde` installerade.

Det är antagligen *inte* den korrekta metoden att använda om du inte redan har paketen `libfam0c102` och `xlibmesa-glu` installerade:

```
# dpkg -l libfam0c102 | grep ^ii
# dpkg -l xlibmesa-glu | grep ^ii
```

Om du har ett fullständigt skrivbordssystem installerat kan du köra:

```
# aptitude install libfam0 xlibmesa-glu
```

Uppgradering av ett system med några X-paket installerade

System med vissa X-paket installerade, men inte den fullständiga funktionen `desktop`, kräver en annan metod. Den här metoden gäller i allmänhet för system med `xfree86-common` installerat, inklusive vissa serversystem som har serverfunktioner i `tasksel` installerade eftersom några av dessa funktioner inkluderar grafiska hanteringsverktyg. Det är antagligen den rätta metoden att använda för system som kör X, men som inte har den fullständiga funktionen `desktop` installerad.

```
# dpkg -l xfree86-common | grep ^ii
```

Kontrollera först om du har paketen `libfam0c102` och `xlibmesa-glu` installerade.

```
# dpkg -l libfam0c102 | grep ^ii
# dpkg -l xlibmesa-glu | grep ^ii
```

Om du inte har `libfam0c102` installerat, ska du inte inkludera `libfam0` i följande kommandorad. Om du inte har `xlibmesa-glu` installerat, ska du inte inkludera det i följande kommandorad.³

```
# aptitude install x11-common libfam0 xlibmesa-glu
```

Observera att om du installerar `libfam0` kommer det även att installera File Alteration Monitor (`fam`) såväl som RPC portmapper (`portmap`) om de inte redan finns tillgängliga i ditt system. Båda paketen kommer att aktivera nya nätverkstjänster i systemet även om de båda kan konfigureras till att bindas till det (interna) vändslingegränssnittet.

Uppgradering av ett system som inte har stöd för X installerat

På ett system utan X behövs inga extra installationskommandon för `aptitude` och du kan hoppa till nästa steg.

4.5.5 Uppgradering av kärnan

Versionen för `udev` i `etch` saknar stöd för kärnversioner tidigare än 2.6.15 (som inkluderar `sarge` 2.6.8-kärnor), och `udev`-versionen i `sarge` kommer inte att fungera korrekt med de senaste kärnorna. I tillägg till det kommer en installation av `etch`-versionen av `udev` att tvinga igenom en borttagning av `hotplug` som används av Linux 2.4-kärnor.

Som en konsekvens av detta kommer antagligen det tidigare kärnpaketet inte att starta upp korrekt efter den här uppgraderingen. Det finns även ett tidsfönster under uppgraderingen i vilket `udev` har blivit uppgraderat men den senaste kärnan har inte blivit installerad. Om systemet blev omstartat vid den här tidpunkten, mitt i uppgraderingen, kan det orsaka att systemet inte kan starta upp på grund av att drivrutinerna inte identifieras och läses in korrekt. (Se 'Förbered en säker miljö för uppgraderingen' på sidan 17 för rekommendationer för att förbereda sig för den här tänkbara situationen om du uppgraderar från ett fjärrsystem.)

Såvida inte ditt system har funktionen `desktop` installerad, eller andra paket som skulle orsaka ett inte accepterbart antal paketborttagningar, är det därför rekommenderat att du uppgraderar kärnan för sig själv.

För att fortsätta med den här kärnuppgraderingen ska du köra:

```
# aptitude install linux-image-2.6-variant
```

³Det här kommandot kommer att fastställa huruvida du behöver `libfam0` och `xlibmesa-glu` installerade och automatiskt välja dem åt dig:

```
# aptitude install x11-common \ $(dpkg-query -showformat '${Package} ${Status}\n' -W libfam0c102 xlibmesa-glu \ | grep 'ok installed$' | sed -e's/ .*//; s/c102//')
```

Se 'Installera metapaketen för kärnan' på sidan 29 för hjälp med att fastställa vilken variant av kärnpaketet som du ska installera.

I skrivbordsfallet är det tyvärr inte möjligt att försäkra sig om att det nya kärnpaketet installeras omedelbart efter det nya `udev` installeras, så därför finns det ett tidsfönster när ditt system inte kommer att ha någon kärna med fullständigt stöd för hotplug installerad. Se 'Uppgradering av din kärna och relaterade paket' på sidan 29 för information om hur man konfigurerar systemet till att inte vara beroende av hotplug vid uppstart.

4.5.6 Uppgradering av resten av systemet

Du är nu redo att fortsätta med huvuddelen av uppgraderingen. Kör:

```
# aptitude dist-upgrade
```

Det här kommer att genomföra en fullständig uppgradering av systemet, alltså installera de senaste tillgängliga versionerna av samtliga paket och lösa alla tänkbara beroendeändringar mellan paketen i olika utgåvor. Om det är nödvändigt kommer det även att installera några nya paket (vanligtvis nya versioner av bibliotek eller paket som fått nya namn) samt ta bort eventuella föråldrade paket som står i konflikt med varandra.

Vid uppgradering från en uppsättning cd-skivor, kommer du att bli uppmanad att mata in specifika cd-skivor vid olika tillfällen under uppgraderingen. Du kanske måste mata in samma cd-skiva flera gånger; det här är på grund av sammankopplade paket som har blivit utspridda över cd-skivorna.

Nya versioner av installerade paket, som inte kan uppgraderas utan att ändra installationsstatus för ett annat paket, kommer att lämnas kvar vid deras nuvarande version (visas som "held back"). Det kan lösas genom att antingen använda `aptitude` för att välja dessa paket för installation eller genom att prova `aptitude -f install paket`.

4.5.7 Hämta paketsignaturer

Efter uppgraderingen, med den nya versionen av `apt`, kan du nu uppdatera din paketinformation som kommer att inkludera den nya kontrollmekanismen för paketsignaturer:

```
# aptitude update
```

Uppgraderingen har redan hämtat och aktiverat signeringsnycklarna för Debians paketarkiv. Om du lägger till andra (inofficiella) paketkällor kommer `apt` att skriva ut varningar relaterade till dess oförmåga att bekräfta att paketen som hämtas från dem är legitima och har inte ändrats av någon. För mer information, se 'Pakethantering' på sidan 5.

Du kommer att märka det, eftersom du använder den nya versionen av `apt`, att den hämtar paketskillnadsfiler (`pdiff`) istället för den fullständiga paketindexlistan. För mer information om den här funktionen kan du läsa 'Långsammare uppdatering av APT:s paketindexfiler' på sidan 38.

4.5.8 Möjliga problem under uppgraderingen

Om en åtgärd med `aptitude`, `apt-get` eller `dpkg` misslyckas med felet

```
E: Dynamic MMap ran out of room
```

räcker inte standardutrymmet för cachen till. Du kan lösa det här genom att antingen ta bort eller kommentera rader som du inte behöver i `/etc/apt/sources.list` eller genom att öka storleken på cachen. Storleken på cachen kan ökas genom att ställa in `APT::Cache-Limit` i `/etc/apt/apt.conf`. Följande kommando kommer att ställa in den till ett värde som ska vara tillräckligt för uppgraderingen:

```
# echo 'APT::Cache-Limit "12500000";' >> /etc/apt/apt.conf
```

Det här förutsätter att du inte redan har ställt in variabeln i den filen.

Ibland är det nödvändigt att aktivera alternativet `APT::Force-LoopBreak` i APT för att temporärt ta bort ett systemkritiskt paket på grund av en `Conflicts/Pre-Depends`-slinga. `aptitude` kommer att varna dig om det här och avbryta uppgraderingen. Du kan lösa det genom att ange alternativet `-o APT::Force-LoopBreak=1` på kommandoraden för `aptitude`.

Det är möjligt att beroendestrukturen för ett system kan vara så skadat att det kräver handpåläggning. Vanligtvis innebär det att använda `aptitude` eller

```
# dpkg --remove paketnamn
```

för att plocka bort några av de störande paketen, eller

```
# aptitude --fix-broken install
# dpkg --configure --pending
```

I extrema fall kan du behöva tvinga fram en ominstallation med ett kommando som detta

```
# dpkg --install /sökväg/till/paketnamn.deb
```

Filkonflikter bör inte inträffa om du uppgraderar från ett "rent" sarge-system, men kan inträffa om du har inofficiella bakåtporteringar installerade. En filkonflikt resulterar i ett fel som:

```
Packar upp <paket-foo> (från <paketfil-foo>) ...
dpkg: fel vid hantering av <paket-foo> (--install):
  försöker skriva över "<något-filnamn>",
  som också finns i paketet <paket-bar>
dpkg-deb: underprocessen paste dödad av signal (Brutet rör)
Fel uppstod vid hantering:
<paket-foo>
```

Du kan försöka lösa en filkonflikt genom att tvinga igenom borttagning av paketet som nämns på *sista* raden i felmeddelandet:

```
# dpkg -r --force-depends paketnamn
```

Efter att problemen har lösts, bör du kunna återuppta uppgraderingen genom att upprepa tidigare beskrivna `aptitude`-kommandon.

Under uppgraderingen kommer det att ställas frågor om konfiguration eller omkonfigurationen av flera paket. När du blir tillfrågad om någon fil i katalogerna `/etc/init.d` eller `/etc/terminfo`, eller filen `/etc/manpath.config` ska ersättas av paketansvariges version, är det oftast nödvändigt att svara "ja" för att upprätthålla systemets tillstånd. Du kan alltid återgå till de gamla versionerna, eftersom de kommer att sparas med en `.dpkg-old`-ändelse.

Om du inte är säker på vad som behöver göras, skriv ner namnet på paketet eller filen och red ut saker och ting senare. Du kan söka i typescript-filen för att granska informationen som visades på skärmen under uppgraderingen.

4.6 Uppgradering av din kärna och relaterade paket

Det här avsnittet förklarar hur man uppgraderar sin kärna och identifierar tänkbara problem relaterade till den här uppgraderingen. Du kan antingen installera ett av paketen `linux-image-*` som tillhandahålls av Debian, eller bygga en anpassad kärna från källkod.

Observera att en hel del information i det här avsnittet är baserad på antagelsen att du kommer att använda en av de modulära Debian-kärnorna tillsammans med `initramfs-tools` och `udev`. Om du har valt att använda en anpassad kärna som inte kräver en `initrd` eller om du använder en annan `initrd`-generator kan delar av den här informationen vara irrelevant för dig.

Observera också att om `udev` *inte* är installerat på ditt system är det fortfarande möjligt att använda `hotplug` för att identifiera hårdvara.

Om du för närvarande använder en 2.4-kärna bör du även läsa 'Uppgradering till en 2.6-kärna' på sidan [39](#) noggrant.

4.6.1 Installera metapaketet för kärnan

När du kör `dist-upgrade` från `sarge` till `etch`, rekommenderas det starkt att du installerar ett nytt `linux-image-2.6-*-metapaket`. Det här paketet kan installeras automatiskt av `dist-upgrade`-processen. Du kan verifiera det genom att köra:

```
# dpkg -l "linux-image*" | grep ^ii
```

Om du inte ser något utdata, behöver du installera ett nytt `linux-image`-paket för hand. Kör följande kommando för att se en lista över tillgängliga `linux-image-2.6`-metapaket:

```
# apt-cache search linux-image-2.6- | grep -v transition
```

Om du är osäker på vilket paket du ska välja, kör `uname -r` och leta efter ett paket med liknande namn. Om du till exempel ser "2.4.27-3-686" rekommenderas det att du installerar `linux-image-2.6-686`. Du kan även använda `apt-cache` för att se en lång beskrivning för varje paket för att hjälpa dig att välja den bästa möjliga. Till exempel:

```
# apt-cache show linux-image-2.6-686
```

Du bör sedan använda `aptitude install` för att installera den. När den här nya kärnan har installerats bör du starta om vid nästa möjliga tillfälle för att dra nytta av den nya kärnversionen.

För den mer äventyrlige finns det ett enkelt sätt att bygga din egna anpassade kärna på Debian GNU/Linux. Installera verktyget `kernel-package` och läs dokumentationen i `/usr/share/doc/kernel-package`.

4.6.2 Uppgradering från en 2.6-kärna

Om du för närvarande kör en kärna ur 2.6-serien från `sarge` kommer den här uppgraderingen att ske automatiskt efter att du kör en fullständig uppgradering av systempaketen (som beskrivs i 'Uppgradering av paket' på sidan 22).

Om möjligt är det till din fördel att uppgradera kärnpaketet separat från själva `dist-upgrade` för att minska chanserna för ett temporärt icke-startbart system. Se 'Uppgradering av kärnan' på sidan 26 för en beskrivning av den här processen. Observera att det här bör endast göras efter den minimala uppgraderingsprocessen, beskriven i 'Minimal systemuppgradering' på sidan 24.

Du kan också göra det här steget om du använder din egna anpassade kärna och vill använda kärnan som finns tillgänglig i `etch`. Om din kärnversion inte stöds av `udev` är det rekommenderat att du uppgraderar efter den minimala uppgraderingen. Om din version stöds av `udev` kan du med säkerhet vänta tills efter den fullständiga systemuppgraderingen.

4.6.3 Uppgradering från en 2.4-kärna

Om du har en 2.4-kärna installerad och ditt system förlitar sig på `hotplug` för att identifiera sin hårdvara bör du först uppgradera till en 2.6-kärna från `sarge` innan du försöker med uppgraderingen. Försäkra dig om att 2.6-kärnan startar upp ditt system och att all din hårdvara identifieras korrekt innan du genomför uppgraderingen. Paketet `hotplug` tas bort från systemet (och ersätts av `udev`) när du kör en fullständig systemuppgradering. Om du inte gör kärnuppgraderingen innan det här kanske ditt system inte kan starta upp korrekt från och med nu. När du har gjort en uppgradering till en 2.6-kärna i `sarge` kan du göra en kärnuppgradering som beskrivs i 'Uppgradering från en 2.6-kärna' på denna sida.

Om ditt system inte förlitar sig på `hotplug`⁴ kan du fördröja kärnuppgraderingen till efter att du har gjort en fullständig systemuppgradering, som beskrivs i 'Uppgradering av resten av systemet' på sidan 27. När ditt system har blivit uppgraderat kan du sedan göra följande (ändra kärnpaketets namn till det som mest passar ditt system genom att ersätta `<variant>`):

```
# aptitude install linux-image-2.6-<variant>
```

4.6.4 Ny ordning för enhetsnumrering

etch har en mer robust mekanism för att identifiera hårdvara än tidigare utgåvor. Dock kan den här orsaka ändringar på den ordning som enheter identifieras på ditt system vilket påverkar ordningen i vilken enhetsnamnen tilldelas. Om du till exempel har två nätverkskort som associeras med två olika drivrutiner, kan de enheter som `eth0` och `eth1` refererar till ändras. Observera att den nya mekanismen betyder att om du till exempel byter Ethernet-kort på ett körande etch-system, kommer det nya kortet även att få ett nytt gränssnittsnamn.

För nätverksenheter kan du undvika den här omnumreringen genom att använda regler i `udev`, mer specifikt genom definitionerna i `/etc/udev/rules.d/z25_persistent-net.rules`⁵. Alternativt kan du använda verktyget `ifrename` för att binda fysiska enheter till specifika namn vid uppstarten. Se `ifrename(8)` och `iftab(5)` för mer information. De två alternativen (`udev` och `ifrename`) bör inte användas samtidigt.

För lagringsenheter kan du undvika den här omsorteringen genom att använda `initramfs-tools` och konfigurera det att läsa in drivrutinsmoduler för lagringsenheter i samma ordning som de för närvarande läses in i. För att för det här, identifiera ordningen på lagringsmodulerna på ditt system genom att se på utdatat från `lsmod`. `lsmod` listar moduler i omvänd ordning än den de lästes in i, alltså den första modulen i listan är den som sist lästes in. Observera att det här endast fungerar för enheter som kärnan numrerar i en stabil ordning (som PCI-enheter).

Dock kommer borttagning och omläsning av moduler efter initial uppstart att påverka ordningen. Din kärna kan även ha några drivrutiner som är statiskt länkade, och dessa namn kommer inte att visas i utdatat från `lsmod`. Du kan möjligen läsa ut dessa drivrutinsnamn och inläsningsordning genom att se i `/var/log/kern.log`, eller utdatat från `dmesg`.

Lägg till dessa modulnamn till `/etc/initramfs-tools/modules` i den ordning som de ska läsas in i vid uppstarten. Vissa modulnamn kan ha ändrats mellan `sarge` och `etch`. Till exempel har `sym53c8xx_2` blivit `sym53c8xx`.

Du kommer att behöva generera om dina `initramfs`-avbild(er) genom att köra `update-initramfs -u -k all`.

⁴Du kan ha kärnmodulerna, som behövs av ditt system, statiskt inlästa genom en korrekt konfiguration av `/etc/modules`

⁵Reglerna där genereras automatiskt av skriptet `/etc/udev/rules.d/z45_persistent-net-generator.rules` för att få statiska namn för nätverksgränssnitten. Ta bort den här symboliska länken för att inaktivera den statiska enhetsnamngivningen för nätverkskort genom `udev`.

När du väl kör en etch-kärna tillsammans med `udev` kan du konfigurera om ditt system till att komma åt diskarna genom ett alias som inte är beroende av inläsningsordningen. Dessa alias finns under `/dev/disk/-hierarkin`.

4.6.5 Ny ordning för serieenheter

Om du har en HP-dator och du använder MP-seriekonsollporten (kontakten märkt "console" på 3-kontakters kabeln), denna kärnuppgraderingen kommer att göra din konsoll trasig!

Vid omstart kommer systemet att visa meddelandet "Loading initrd..." men det kommer att stoppa där. Observera att system med utdaterad fast programvara kommer att visa liknande symptom, även om problemet är relaterat till inkompatibiliteter i kärnan (se 'Uppgradering till en 2.6-kärna' på sidan 39).

Läs följande information innan uppgraderingen påbörjas.

- Konsollenheten kommer att ändras från `ttyS0` till `ttyS1`, `ttyS2`, eller `ttyS3`
 - Redigera `/etc/inittab` för att lägga till en `getty`-post för `/dev/ttyS1` (rx4640, rx5670, rx7620, rx8620, Superdome), `/dev/ttyS2` (rx1600), eller `/dev/ttyS3` (rx2600).
 - Redigera `/etc/securetty` för att lägga till `ttyS1`, `ttyS2`, eller `ttyS3`.
 - Lämna kvar de befintliga `ttyS0`-posterna i `/etc/inittab` och `/etc/securetty` så att du fortfarande kan starta upp med äldre kärnor.
- Redigera `/etc/elilo.conf` för att ta bort eventuella "console=" argument.
- Kör `elilo` för att installera starthanteraren med den nya konfigurationen.
- Starta om och använd EFI-uppstartsalternativet för underhållsmenyn för att välja exakt en enhet för konsollutdata, inmatning och standard fel. Gör sedan en kall omstart så att ändringarna börjar gälla.
För MP-konsollen, var noga med att välja enheten med "Acpi(HWP0002,700)/Pci(...)/Uart" i sökvägen.

Mera detaljer om dessa ändringar och problemlösning finns tillgängliga på <http://lists.debian.org/debian-ia64/2005/01/msg00008.html>.

4.6.6 Tidsproblem vid uppstart

Om en `initrd` som skapats med `initramfs-tools` används för att starta upp systemet, kan i vissa fall skapandet av enhetsfiler av `udev` ske för sent för uppstartsskripten att agera på.

De vanliga symptomen är att uppstarten misslyckas på grund av att rotfilssystemet inte kan monteras och du försätts i ett felsökningsskal, men att när du kontrollerar senare så finns alla

nödvändiga enheter i `/dev`. Det har observerats i fall där rotfilsystemet finns på en USB-disk eller på RAID.

Ett sätt att komma runt det här problemet på är att använda uppstartsparametern `rootdelay=9`. Värdet för tidsgränsen (i sekunder) kan behöva justeras.

4.7 Saker att göra före omstart

När `aptitude dist-upgrade` har kört färdigt är den "formella" uppgraderingen färdig, men det finns vissa andra saker som bör tas om hand *före* nästa omstart.

4.7.1 Konvertering från `devfs`

Debian-kärnor inkluderar inte längre stöd för `devfs`. Därför behöver användare med `devfs` manuellt konvertera sina system före uppstart med en `etch`-kärna.

Om du ser strängen "devfs" i `/proc/mounts` så använder du säkerligen `devfs`. De konfigurationsfiler som refererar till `devfs`-liknande namn kommer att behöva justeras till att använda `udev`-liknande namn. Filer som säkerligen refererar till `devfs`-liknande enhetsnamn inkluderar `/etc/fstab`, `/etc/lilo.conf`, `/boot/grub/menu.lst` och `/etc/inittab`.

Mer information om tänkbara problem finns tillgängligt i felrapporten #341152 (<http://bugs.debian.org/341152>).

4.7.2 Uppgradering av `mdadm`

`mdadm` behöver nu en konfigurationsfil för att sätta samman MD-kedjor (RAID) från den initiala ramdisken och under systemets initieringssekvens. Se till att läsa och agera efter de instruktioner som finns i `/usr/share/doc/mdadm/README.upgrading-2.5.3.gz` efter att paketet har uppgraderats **och innan du startar om**. Den senaste versionen av den här filen finns tillgänglig på <http://svn.debian.org/wsvn/pkg-mdadm/mdadm/trunk/debian/README.upgrading-2.5.3?op=file>; läs den om du får problem.

4.8 Förberedelse inför nästa utgåva

Efter uppgraderingen finns det flera saker som du kan göra för att förbereda inför nästa utgåva.

- Om du använder `grub`, redigera `/etc/kernel-img.conf` och justera platsen för programmet `update-grub` genom att ändra `/sbin/update-grub` till `/usr/sbin/update-grub`.
- Om den nya kärnavbildens metapakets drogs in som ett beroende till den gamla, kommer det att markeras som automatiskt installerat, vilket bör korrigeras:

```
# aptitude unmarkauto $(dpkg-query -W 'linux-image-2.6-*' | cut -f1)
```

- Ta bort sarge-kärnans metapakets genom att köra:

```
# aptitude purge kernel-image-2.6-<variant>
```

- Flytta eventuella konfigurationsvärden för kärnan i `/etc/network/options` till `/etc/sysctl.conf`.
- Ta bort föråldrade och oanvända paket som beskrivs i 'Föråldrade paket' på denna sida. Du bör granska vilka konfigurationsfiler som de använder och överväga att avinstallera paketen fullständigt för att ta bort deras konfigurationsfiler

4.9 Föråldrade paket

etch introducerar tusentals nya paket men pensionerar och utelämnar mer än två tusen gamla paket som fanns i sarge. Den tillhandahåller inget uppgraderingssätt för dessa föråldrade paket. Då ingenting hindrar dig från att fortsätta att använda ett föråldrat paket om så önskas, kommer Debian-projektet vanligtvis att sluta ge säkerhetsstöd för det ett år efter utgivningen av etch⁶, och kommer normalt sett inte att tillhandahålla annat stöd under den tiden. Att ersätta dem med tillgängliga alternativ, om det finns några, rekommenderas.

Det finns många anledningar till varför paket kan ha tagits bort från distributionen: de underhålls inte längre av upphovsmännen; det finns inte längre någon Debian-utvecklare som är intresserad i att underhålla paketen; funktionaliteten de tillhandahåller har ersatts av en annan programvara (eller en ny version); eller så anses de inte längre vara lämpliga för etch på grund av fel i dem. I det senare fallet kan paket fortfarande finnas i "unstable"-distributionen.

Att identifiera vilka paket på ett uppdaterat system som är "föråldrade" är enkelt eftersom pakethanteringsvertygen markerar dem så. Om du använder `aptitude`, kommer du att se en lista över dessa paket under "Föråldrade och lokalt skapade paket". `dselect` tillhandahåller en liknande sektion men listning som visas kan skilja sig. Dessutom, om du har använt `aptitude` för att manuellt installera paket i sarge kommer den att hålla kontroll på de paket som du manuellt installerat och kommer att kunna markera de paket som själva har hämtats in av beroenden, vilka inte längre behövs om ett paket har tagits bort. Dessutom, `aptitude`, till skillnad från `deborphan` kommer inte markera de manuellt installerade som föråldrade paket, i motsats till de som blev automatiskt installerade genom beroenden.

Det finns ytterligare verktyg som du kan använda för att hitta föråldrade paket, såsom `deborphan`, `debfooster` eller `cruft`. `deborphan` rekommenderas starkt, även om det endast kommer (i standardläget) rapportera om föråldrade bibliotek: paket i sektionerna "libs" eller "oldlibs" som inte används av några andra paket. Ta inte helt blint bort de paket som dessa verktyg presenterar, speciellt om du använder aggressiva ickestandardalternativ som är

⁶Eller så länge som ingen annan utgivning sker i den tidsperioden. Normalt sett stöds endast två stabila utgåvor åt gången.

benägna att producera felaktigheter. Det rekommenderas starkt att du manuellt granskar paketen som föreslås för borttagning (alltså deras innehåll, storlek och beskrivning) innan du tar bort dem.

Debians felhanteringssystem (<http://bugs.debian.org/>) tillhandahåller ofta ytterligare information om varför paketet blev borttaget. Du bör granska både de arkiverade felrapporterna för själva paketet och de arkiverade felrapporterna för pseudopaketet på ftp.debian.org (<http://bugs.debian.org/cgi-bin/pkgreport.cgi?pkg=ftp.debian.org/&archive=yes>).

4.9.1 Dummy-paket

Vissa paket från sarge har delats upp i flera paket i etch, ofta för att förbättra systemunderhållet. För att göra uppgraderingssättet enklare i sådana fall, tillhandahåller etch ofta så kallade "dummy"-paket: tomma paket som har samma namn som det gamla paketet i sarge med beroenden som gör att de nya paketen blir installerade. Dessa "dummy"-paket anses som föråldrade paket efter uppgraderingen och kan med säkerhet tas bort.

Beskrivningarna för de flesta (men inte alla) dummy-paket indikerar deras syfte. Paketbeskrivningar för dummy-paket är inte enhetliga, dock kan `deborphan` med flaggan `--guess` vara användbar för att identifiera dem på ditt system. Observera att vissa dummy-paket inte är tänkta att tas bort efter en uppgradering men används istället för att hålla kontroll på den för närvarande tillgängliga versionen av ett program över tid.

Kapitel 5

Eventuella problemsituationer för etch

5.1 Tänkbara problem

Ibland kan ändringar ge bieffekter som vi rimligen inte kan undvika, eller att vi blottlägger fel någon annanstans. Vi dokumenterar de problem vi känner till här. Läs även errata, de relevanta paketens dokumentation, felrapporter och annan information som nämns i 'Ytterligare läsning' på sidan [47](#).

5.1.1 Problem med enheter relaterade till udev

Även om `udev` har utförligt testats, kan du uppleva mindre problem med vissa enheter som behöver rättas till. De mest vanliga problemen är ändrade rättigheter och/eller ägarskap på en enhet. I vissa fall kan en enhet inte skapas som standard (t.ex. `/dev/video` och `/dev/radio`).

`udev` tillhandahåller konfigurationsmekanismer för att hanera dessa problem. Se `udev(8)` och `/etc/udev` för ytterligare information.

5.1.2 Vissa program kanske inte längre fungerar med en 2.4-kärna

Vissa program i etch kanske inte längre fungerar med en 2.4-kärna, till exempel på grund av att de kräver stöd för `epoll()`, som inte finns tillgängligt i 2.4-kärnor. Sådana program kanske antingen inte fungerar alls eller inte fungerar korrekt tills systemet har startats om med en 2.6-kärna.

Ett exempel är HTTP-proxyservern `squid`.

5.1.3 Vissa nätverksplatser kan inte nås med TCP

Sedan 2.6.17 använder Linux "TCP window scaling" på ett aggressivt sätt, vilket finns beskrivet i RFC 1323. Vissas servrar har ett trasigt beteende och annonserar felaktiga "window sizes"

för sig själva. Se felrapporterna #381262 (<http://bugs.debian.org/381262>) och #395066 (<http://bugs.debian.org/395066>) för mer information.

5.1.4 Långsammare uppdatering av APT:s paketindexfiler

Som standard använder etch-versionen av apt ett nytt sätt att uppdatera paketindexfilerna för APT (när du kör `aptitude update`) som hämtar filskillnader (istället för den fullständiga paketindexfilen) kallad `pdiff`. Den här nya funktionen bör använda mindre bandbredd och vara snabbare för de flesta system. Tyvärr kan den även ha den motsatta effekten genom att göra uppdateringar långsammare på system med snabba nätverksanslutningar (eller en mycket närliggande spegelservare) som inte ofta uppdateras, eftersom den kan ta längre tid på sig att sammanfoga skillnadsfilerna än att hämta ner det fullständiga paketindexet. Det är möjligt att inaktivera den här funktionen genom att lägga till `Acquire::Pdiffs "false"`; till konfigurationsfilen `/etc/apt/apt.conf`.

Den här ändringen påverkar mestadels användare som kör *unstable* och *testing* av Debian GNU/Linux, på grund av ändringens natur för dessa arkiv. Användare som kör etch kommer huvudsakligen att märka av den här funktionen när de uppdaterar sin paketstatus för säkerhetsarkiven.

5.1.5 Asynkron nätverksinitiering kan orsaka oförutsett beteende

På system som använder `udev` för att läsa in drivrutiner för nätverksgränssnitt är det möjligt, på grund av den asynkrona naturen för `udev`, att nätverksdrivrutinen inte läses in före `/etc/init.d/networking` körs vid systemets uppstart. Även om man inkluderar `allow-hotplug` i `/etc/network/interfaces` (i tillägg till `auto`) kommer att säkerställa att nätverksgränssnittet är aktiverat när det blir tillgängligt, är det inte garanterat att det kommer att färdigställas innan uppstartssekvensen börjar att starta upp nätverkstjänster, vissa av dem kanske inte beter sig korrekt då nätverksgränssnittet saknas.

5.1.6 Problem vid användning av WPA-säkrade trådlösa nätverk

I sarge blev paketet `wpasupplicant` konfigurerat som en systemtjänst och konfigurerades via `/etc/default/wpasupplicant` och en användarens egna `/etc/wpasupplicant.conf`.

I etch har filen `/etc/init.d/wpasupplicant` utelämnats och Debian-paketet har nu integrerats med `/etc/network/interfaces`, liknande andra paket såsom `wireless-tools`. Det betyder att `wpasupplicant` inte längre tillhandahåller en systemtjänst direkt.

För information om hur man konfigurerar `wpasupplicant`, referera till `/usr/share/doc/wpasupplicant/README.modes.gz`, som ger exempel på `/etc/network/interfaces`. Uppdaterad information om användningen av paketet `wpasupplicant` i Debian kan hittas på Debians wiki (<http://wiki.debian.org/WPA>).

5.1.7 Problem med icke-ASCII-tecken i filnamn

Montering av filsystemstyperna `vfat`, `ntfs` eller `iso9660` med filer som inkluderar icke-ASCII-tecken i sina filnamn kommer att ge fel när man försöker att använda filnamnen såvida inte monteringen sker med alternativet `utf8`. En indikering kan vara följande felmeddelande: "Invalid or incomplete multibyte or wide character". En möjlig lösning är att använda `defaults,utf8` som monteringsalternativ för filsystemstyperna `vfat`, `ntfs` och `iso9660` när de innehåller filnamn med icke-ASCII-tecken.

Observera att Linux-kärnan inte har stöd för skiftlägesokänslig hantering av filnamn för `vfat` när alternativet `utf8` används.

5.2 Uppgradering till en 2.6-kärna

2.6-serien av kärnan innehåller stora ändringar mot 2.4-serien. Moduler har bytt namn och en hel del drivrutiner har till viss del eller ibland skrivits om totalt. Uppgradering till en 2.6-kärna från en tidigare version är därför inte en process som ska underskattas. Det här avsnittet har som mål att göra dig medveten om några av problemen du kan möta.

Om du bygger din egna kärna från källkod, se till att du installerar `module-init-tools` före du startar om med 2.6-kärnan. Det här paketet ersätter `modutils` för 2.6-kärnor. Om du installerar ett av Debians `linux-image`-paket, kommer det här paketet att installeras automatiskt på grund av beroenden.

Om du använder *LVM*, bör du även installera `lvm2` före du startar om eftersom 2.6-kärnan inte direkt stöder *LVM1*. För att komma åt *LVM1*-volym, används kompatibilitetslagret i `lvm2` (modulen `dm-mod`). Du kan lämna `lvm10` installerat; `init`-skripten kommer att identifiera vilken kärna som används och köra den lämpliga versionen.

Om du har poster i filen `/etc/modules` (listan över moduler som ska läsas in under systemets uppstart), tänk på att vissa modulnamn har ändrats. Om det här händer måste du uppdatera den här filen med de nya modulnamnen.

HP Itanium-system som kör äldre versioner av den fasta programvaran är inte kompatibla med 2.6-kärnan i etch. Det betyder att du bör uppgradera ditt system till den senaste versionen av den fasta programvaran innan du uppgraderar din kärna. Det rekommenderas att du gör det här innan systemuppgraderingen, eftersom om du redan kör en 2.6-kärna kommer du automatiskt att hämta den senaste kärnan när du uppgraderar resten av systemet (se 'Uppgradering av resten av systemet' på sidan 27). Misslyckas det här kommer det resultera i att systemet inte kan starta upp.

När du har installerat din 2.6-kärna, men före du startar om, se till att du har en återställningsmetod. Se först till att konfigurationen av starthanteraren innehåller poster för den nya kärnan och den gamla fungerande 2.4-kärnan. Du bör även se till att du har en "rescue"-diskett eller cd-skiva till hands, i fall att en felkonfiguration av starthanteraren förhindrar dig från att starta upp den gamla kärnan.

5.2.1 Tangentbordskonfiguration

Den största ändringen i 2.6-kärnorna är en huvudsaklig ändring av inmatningslagret. Den här ändringen gör att alla tangentbord ser ut som "normala" PC-tangentbord. Det här betyder att om du för närvarande har en annan typ av tangentbord vald (t.ex. ett USB-MAC- eller Sun-tangentbord), kommer du troligen inte ha ett fungerande tangentbord efter omstart med den nya 2.6-kärnan.

Om du kan köra SSH till burken från ett annat system, kan du lösa det här problemet genom att köra `dpkg-reconfigure console-data`, välj alternativet "Välj tangentuppsättning från komplett lista" och välj ett "pc"-tangentbord.

Om ditt konsolltangentbord berörs, behöver du antagligen även konfigurera om ditt tangentbord för X Window System. Du kan göra det här genom att antingen köra `dpkg-reconfigure xserver-xorg` eller genom att redigera `/etc/X11/xorg.conf` direkt. Glöm inte att läsa dokumentationen som refereras till i 'Saker att göra före omstart' på sidan 33.

Observera att om du använder ett USB-tangentbord, kan det konfigureras som antingen ett "normalt" PC-tangentbord eller som ett USB-MAC-tangentbord. I det första fallet kommer du inte drabbas av det här problemet.

5.2.2 Muskonfiguration

Precis som ovanstående ändringar i inmatningslagret, kan du även behöva konfigurera om X Window System och `gpm` om din mus inte fungerar efter uppgradering till en 2.6-kärna. Den mest troliga orsaken är att enheten som får data från musen har ändrats. Du kan även behöva att läsa in andra moduler.

5.2.3 Ljudkonfiguration

För 2.6-serien av kärnor rekommenderas ALSA-ljuddrivrutinerna framför de äldre OSS-ljuddrivrutinerna. ALSA-ljuddrivrutinerna tillhandahålls som standard som moduler. För att ljudet ska fungera, måste ALSA-modulerna som lämpar sig för din ljudhårdvara läsas in. Generellt sett händer det här automatiskt om du har, i tillägg till `alsa-base`-paketet, antingen paketet `hotplug` eller paketet `discover` installerade. Paketet `alsa-base` "svartlistar" även OSS-moduler för att förhindra att `hotplug` och `discover` läser in dem. Om du har OSS-moduler listade i `/etc/modules`, bör du ta bort dem.

5.3 Övergång från XFree86 till X.Org

Övergången till X.Org innefattar även några strukturella ändringar. Om alla installerade paket kommer från Debian och även finns inkluderade i etch bör uppgraderingen fungera utan några som helst problem. Erfarenheter har dock visat att det kan finnas vissa ändringar som man bör känna till eftersom de möjligen kan orsaka problem under uppgraderingen.

Den mest viktiga ändringen är att `/usr/X11R6/bin` har tagits bort och återfinns endast som en symbolisk länk till `/usr/bin`. Det här betyder att katalogen måste vara tom när de nya paketen installeras. De nya paketen står i konflikt med de flesta paket som använder `/usr/X11R6/bin`, men i vissa fall kan manuell handpåläggning behövas. Kom ihåg att inte köra uppgradering av distributionen inne i en X-session.

I fall att uppgraderingen avbryts under installation av X.Org, bör du kontrollera om några filer fortfarande finns kvar i `/usr/X11R6/bin`. Du kan sedan använda `dpkg -S` för att ta reda på vilket Debian-paket som installerade den filen (om någon), och ta bort sådana paket med `dpkg --remove`. Tänk på att skriva ner vilka paket du tar bort, så att du kan installera ersättningspaket senare. Innan du fortsätter med uppgraderingen, måste alla filer i `/usr/X11R6/bin` tas bort.

Läs <http://wiki.debian.org/Xorg69To7> för fler detaljer och andra problem.

5.4 Inget stöd för 8-bitars display i många program

Efter uppgradering till Xorg och de senaste biblioteken kommer X-terminaler som endast kan visa 8-bitars färgdjup inte längre att fungera. Det är på grund av att vektorgrafikbiblioteket Cairo 2D (`libcairo2`) inte har stöd för 8-bitars pseudofärger. Det biblioteket används av GNOME- och Xfce-skrivborden såväl som av många skrivbordsprogram som byggts med Gtk2+, såsom `abiword`.

Kända system som påverkas av det här är bland annat vissa Sun-maskiner och X-terminaler från Tektronix, NCD, IBM och SGI, och även andra fjärr-X-fönstersystem. Du bör konfigurera dessa terminaler till att använda 16-bitars färgdjup, om det är möjligt.

Mer information finns tillgänglig i Freedesktops felrapport #4945 (https://bugs.freedesktop.org/show_bug.cgi?id=4945).

5.5 Uppgradering från `exim` till `exim4`

Ett av paketen som har föråldrats av etch-utgåvan är e-postservern `exim`, som har ersatts av det helt nya paketet `exim4`.

`exim` (version 3.xx) har inte underhållits på flera år, och Debian har även tagit bort stödet för den versionen. Om du fortfarande använder `exim 3.xx` bör du manuellt uppgradera din `exim`-installation till `exim4`. Eftersom `exim4` redan är en del av `sarge` kan du välja att göra uppgraderingen på ditt `sarge`-system innan uppgradering till etch, eller efter uppgraderingen till etch. Kom ihåg att ditt gamla `exim`-paket inte kommer att uppgraderas och att det inte kommer att få säkerhetsstöd därför att stödet för `sarge` har upphört.

Observera att, beroende på din konfiguration av `debconf`, du kanske inte blir tillfrågad om konfiguration under installationen av `exim4`. Om inga frågor ställs kommer systemet att välja standardkonfigurationsvalet "lokal leverans". Konfiguration är möjlig med kommandot `dpkg-reconfigure exim4-config`.

Paketet för `exim4` i Debian är väldokumenterade. Paketets hemsida är <http://wiki.debian.org/PkgExim4> på Debians wiki, och filen README kan hittas på <http://pkg-exim4.alioth.debian.org/README/README.Debian.html> och även inne i paketet.

Filen README innehåller ett kapitel om paketering (Packaging), som förklarar de olika paketvarianterna som vi erbjuder, och den har ett kapitel om uppdatering från `Exim 3` (Updating from `Exim 3`), som hjälper dig att göra den faktiska övergången.

5.6 Uppgradering av apache2

Apache har uppgraderats till nya versionen 2.2. Även om det här inte bör ha stor inverkan på den genomsnittlige användaren, finns det vissa tänkbara problem man bör känna till.

<http://httpd.apache.org/docs/2.2/upgrading.html> innehåller "uppströms"-ändringarna. Läs den här sidan och tänk speciellt på:

- alla moduler behöver byggas om
- behörighetsmoduler har sorterats om och fått nya namn
- vissa konfigurationsalternativ har fått nya namn

Debian-specifika ändringar är bland annat att SSL-strängen inte längre definieras eftersom `ssl` numera stöds av standardpaketet.

Om du använder den experimentella ITK MPM (från paketet `apache2-mpm-itk`), kommer `cgi`-modulen inte vara korrekt aktiverad som standard. För att aktivera den korrekt behöver du manuellt inaktivera `mod_cgid` och aktivera `mod_cgi`:

```
# cd /etc/apache2/mods-enabled
# rm cgid.conf cgid.load
# ln -s ../mods-available/cgi.load .
# /etc/init.d/apache2 force-reload
```

5.7 Uppgradering av Zope och Plone

Zope och alla relaterade produkter har uppdaterats. Många produkter blev även utelämnade från distributionen (antingen på grund av att de var föråldrade eller på grund av att de inte är kompatibla med senare versioner av Zope, CMF eller Plone).

Tyvärr finns det inget enkelt eller garanterat sätt att uppgradera en komplex `zope`- eller `plone`-server. Även om Plone inkluderar ett verktyg för migrering har erfarenheter visat att automatiska migreringar lätt kan misslyckas.

Av den här anledningen rekommenderas det att användarna konfigurerar sina system så att de kan fortsätta att köra sarge-installationen av Zope/Plone vid sidan om de nya etch-versionerna under tiden migrationstester utföres.

Det enklaste och säkraste sättet att uppnå det här på är att skapa en kopia av ditt sarge-system till en annan hårddisk eller partition, och sedan uppgradera en av kopiorna. Du kan sedan använda `chroot` för att köra sarge-versionen parallellt med etch-versionen.

Det är inte möjligt att ha gamla och nya versioner av Zope/Plone installerade tillsammans på ett etch-system, dels på grund av att de gamla paketen är beroende av `python2.3` som inte kan vara installerad samtidigt som `python2.4`.

5.8 Jokerteckensexpantering (globbing) med GNU tar

Tidigare versioner av GNU `tar` antog en skulliknande "globbing" vid extrahering av filer från eller vid listning av ett arkiv. Till exempel:

```
tar xf foo.tar '*.c'
```

skulle extrahera alla filer vars namn slutade med ".c". Det här beteendet blev inte dokumenterat och var inte kompatibelt med traditionella implementationer av `tar`. Därför, med start från version 1.15.91, använder GNU `tar` inte längre "globbing" som standard. Till exempel tolkas numera ovanstående som en begäran att extrahera filen med namnet "*.c" från arkivet.

Se `/usr/share/doc/tar/NEWS.gz` för ytterligare information.

5.9 NIS och Network Manager

Versionen av `yplibind` inkluderad i `nis` för etch innehåller stöd för Network Manager. Det här stödet orsakar att `yplibind` inaktiverar NIS-klientfunktionalitet när Network Manager rapporterar att datorn är fränkopplad från nätverket. Eftersom Network Manager vanligtvis rapporterar att datorn är fränkopplad när det inte används bör NIS-användare med NIS-klientsystem försäkra sig om att stödet för Network Manager har inaktiverats på dessa system.

Det här kan göras genom antingen avinstallation av paketet `network-manager` eller genom redigering av `/etc/default/nis` för att lägga till `-no-dbus` till `YPBINDARGS`.

Användningen av `-no-dbus` är standard för nya installationer av Debian, men var inte standard i tidigare utgåvor.

5.10 Utfasade osäkra php-konfigurationer

I många år har påslagningen av inställningen `register_globals` i PHP varit känd som osäker och farlig, och den här inställningen har under en tid varit avstängd som standard. Den här

konfigurationen är nu äntligen utfasad på Debian-system då den är för farlig. Det samma gäller för sårbarheter i `safe_mode` och `open_basedir`, vilka heller inte har underhållits under en tid.

Från och med den här utgåvan kommer Debians säkerhetsteam inte att tillhandahålla säkerhetsstöd för ett antal PHP-konfigurationer som är kända att vara osäkra. Mest viktigast är problem som resulterar från att `register_globals` har slagits på, kommer inte längre att hanteras.

Om du kör gamla program som kräver `register_globals`, aktivera det endast för respektive sökvägar, t.ex. genom Apaches konfigurationsfil. Mer information finns tillgänglig i filen `README.Debian.security` i PHP:s dokumentationskatalog (`/usr/share/doc/php4`, `/usr/share/doc/php5`).

5.11 Säkerhetsstatus för Mozilla-produkter

Mozilla-programmen `firefox` och `thunderbird` (omdöpta i Debian till `iceweasel` respektive `icedove`), är viktiga verktyg för många användare. Tyvärr är upphovsmännens säkerhetspolicy att skynda på användare att uppdatera till nya versioner, vilket står i konflikt med Debians policy att inte skicka ut stora funktionella ändringar i säkerhetsuppdateringar. Vi kan inte förutse det här idag, men under livstiden för etch kan Debians säkerhetsgrupp komma till en punkt där stöd för Mozilla-produkter inte längre är möjlig och annonsera ut slutet för säkerhetsstöd för Mozilla-produkter. Du bör överväga det här när du installerar Mozilla och överväger alternativ som finns tillgängliga i Debian om avsaknaden av säkerhetsstödet skulle innebära ett problem för dig.

5.12 KDE-skrivbordet

KDE-mediahantering har ändrats i versionen som finns tillgänglig i etch från att använda `device:/` till `media:/`. Vissa användarkonfigurationsfiler kan ha lagrat `device:/`-länkar i dem som bör anpassas. Nämnvärt är `~/.kde/share/apps/konqsidebar/tnq/virtual_folders/services` som innehåller den här referensen och kan med säkerhet tas bort eftersom den inte kommer att skapas när nya användare skapas.

Det har gjorts många ändringar i skrivbordsmiljön KDE från versionen som skickades med sarge till versionen i etch, du kan hitta mer information i KDE 3.5 Release Notes (<http://www.kde.org/announcements/announce-3.5.php>).

5.13 Ändringar och stöd för GNOME-skrivbordet

Om du tidigare använt GNOME-skrivbordet i sarge kommer du inte kunna dra nytta av några av de ändringar som introducerats i standardkonfigurationen i Debian för etch. I vissa extrema

fall kanske GNOME-skrivbordet inte hanterar din gamla konfiguration på ett korrekt sätt och kanske inte heller uppträder korrekt.

Om du inte har spenderat mycket tid på att konfigurera ditt GNOME-skrivbord kanske du vill flytta katalogen `.gconf` i användarnas hemkataloger till ett annat namn (exempelvis `.gconf.old`) så att den blir återskapad med standardkonfigurationen för etch, nästa gång en ny session startas.

Med utgåvan av etch kommer Debian inte längre att innehålla paket för mestadelen av den föråldrade version 1 av GNOME, även om några paket fortfarande finns kvar för att ge stöd till vissa Debian-paket som ännu inte har uppdaterats till GNOME 2. Paket för GTK1.2 är fortfarande fullständigt underhållna.

Det har skett många ändringar i skrivbordsmiljön GNOME från versionen som levererades med sarge till versionen i etch, du kan hitta mer information i Versionsfakta för GNOME 2.14 (<http://www.gnome.org/start/2.14/notes/sv/>).

5.14 Standardredigerare

Om du tidigare har använt `vim` som din standardredigerare, kan den ha ändrats till `nano` under uppgraderingen.

Administratörer som önskar att ändra standardredigeraren för alla användare behöver uppdatera alternativsystemet genom att köra:

```
# update-alternatives --config editor
```

Användare som önskar ändra standardredigeraren kan definiera miljövariabeln `EDITOR` genom att lägga till följande rader i sina egna profiler:

```
EDITOR=vi
export EDITOR
alias editor=$EDITOR
```


Kapitel 6

Mer information om Debian GNU/Linux

6.1 Ytterligare läsning

Förutom dessa kommentarer till utgåvan och installationsguiden finns ytterligare dokumentation för Debian GNU/Linux tillgänglig från Debian Documentation Project (DDP), som har som mål att skapa högkvalitativ dokumentation för Debian-användare och utvecklare. Dokumentation som till exempel Debian Reference, Debian New Maintainers Guide, och Debian FAQ finns tillgängliga tillsammans med många andra dokument. För fullständiga detaljer om tillgängliga resurser, se webbplatsen för DDP (<http://www.debian.org/doc/ddp>).

Dokumentation för individuella paket installeras i `/usr/share/doc/paket`. Den kan inkludera information om copyright, Debian-specifika detaljer och dokumentation från upphovsmännen.

6.2 Få hjälp

Det finns många källor för hjälp, råd och stöd för Debian-användare, men dessa bör endast användas om dokumentationen inte har hjälpt till att lösa problemet. Det här avsnittet tillhandahåller en kort introduktion till dessa, vilka kan vara till hjälp för nya Debian-användare.

6.2.1 Sändlistor

De sändlistor som är mest intressanta för Debian-användarna är listan `debian-user` (engelsk) och andra `debian-user-språkkod`-listor (för andra språk). För information om dessa listor och detaljer om hur man prenumererar, se <http://lists.debian.org/>. Leta i arkiven efter svar på dina frågor innan du postar en ny fråga och därmed följer listetiketten.

6.2.2 Internet Relay Chat

Debian har en IRC-kanal som är dedicerad till att ge stöd och hjälp till Debian-användare. Kanalen finns på IRC-nätverket OFTC och finns där för att tillhandahålla interaktiva tjänster till punktdrivna projektgemenskaper. För att komma åt kanalen, peka din favorit-IRC-klient till `irc.debian.org` och gå in i kanalen `#debian`.

Följ kanalens riktlinjer, respektera andra användare. För mer information om OFTC, besök dess webbplats (<http://www.oftc.net/>).

6.3 Rapportera fel

Vi strävar mot att göra Debian GNU/Linux till ett högkvalitativt operativsystem, det betyder dock inte att paketen som vi tillhandahåller är helt felfria. I enlighet med Debians filosofi om "öppen utveckling" och som en tjänst till våra användare, tillhandahåller vi all information om rapporterade fel på vårt eget felhanteringssystem (BTS). BTS är bläddringsbar på `bugs.debian.org` (<http://bugs.debian.org/>).

Om du hittar ett fel i distributionen eller i paketerad programvara som är en del av den, vänligen rapportera felet så att det kan rättas till i framtida utgåvor. Felrapportering kräver att du har en giltig e-postadress. Vi frågar efter den så att vi kan spåra fel och utvecklingarna kan komma i kontakt med de som rapporterat felet om de skulle behöva ytterligare information.

Du kan skicka in en felrapport med programmet `reportbug` eller manuellt via e-post. Du kan läsa mer om Bug Tracking System och hur man använder det genom att läsa referenskortet (tillgängliga på `/usr/share/doc/debian` om du har `doc-debian` installerat) eller online på Bug Tracking System (<http://bugs.debian.org/>).

6.4 Att bidra till Debian

Du behöver inte vara en expert för att bidra till Debian. Genom att hjälpa användare med problem på de olika sändlistorna (<http://lists.debian.org/>) för användarstöd bidrar du till gemenskapen. Identifiering (och hitta en lösning, viktigt nog) av problem relaterade till utveckling av distributionen genom att delta i sändlistorna (<http://lists.debian.org/>) för utveckling är också mycket hjälpsamt. För att underhålla Debians högkvalitativa distribution skicka in felrapporter (<http://bugs.debian.org/>) och hjälp utvecklingarna att spåra upp och rätta till dem. Om du är bra på att sätta ihop ord kanske du vill bidra mer aktivt genom att hjälpa till att skriva dokumentation (<http://www.debian.org/doc/ddp>) eller översätta (<http://www.debian.org/international/>) befintlig dokumentation till ditt eget språk.

Om du kan avsätta mer tid, skulle du kunna ansvara för en del av den fria programvaran i Debian. Speciellt behjälpligt är det om personer adopterar eller ansvarar för saker som folk har frågat efter om att inkluderas i Debian. Databasen Work Needing and Prospective Packages (<http://www.debian.org/devel/wnpp/>) har detaljer om den här informationen. Om du

har intresse av mer specifika grupper kan du finna glädje i att bidra till några av Debians underprojekt vilka inkluderar porteringar till specifika arkitekturer, Debian Jr. (<http://www.debian.org/devel/debian-jr/>) och Debian Med (<http://www.debian.org/devel/debian-med/>).

I vilket fall som helst, om du arbetar i den fria programvarugemenskapen på något sätt, som en användare, programmerare, författare eller översättare hjälper du redan den fria programvaran. Att bidra är belönande och roligt, såväl som att det låter dig träffa nya människor som att det ger dig den där varma känslan inom dig.

Bilaga A

Hantera ditt sarge-system

Detta appendix innehåller information om hur du kontrollerar att du kan installera eller uppgradera paket från sarge innan du uppgraderar till etch. Det här bör endast vara nödvändigt i specifika situationer.

A.1 Uppgradering av ditt sarge-system

Det är inga grundläggande skillnader mot någon annan uppgradering av sarge som du gjort. Den enda skillnaden är att du först behöver se till att din paketlista fortfarande innehåller paket från sarge vilket förklarades i 'Kontrollera dina källistor' på denna sida.

Om du uppgraderar ditt system med en Debian-spegel kommer den automatiskt att uppgraderas till den senaste punktutgåvan av sarge.

A.2 Kontrollera dina källistor

Om någon av raderna i din `/etc/apt/sources.list` refererar till "stable", använder du faktiskt redan etch. Om du redan har kört `apt-get update`, kan du fortfarande komma tillbaka utan problem om du följer nedanstående procedur.

Om du även har installerat paket från etch, är det antagligen inte så stor mening att installera paket från sarge längre. I det fallet måste du bestämma dig om du vill fortsätta eller inte. Det är möjligt att nedgradera paket, men det beskrivs inte här.

Öppna filen `/etc/apt/sources.list` med din favoritredigerare (som root) och kontrollera alla rader som börjar med `deb http:` eller `deb ftp:` efter en referens till "stable". Om du hittar någon, ändra `stable` till `sarge`.

Om du har vissa rader som börjar med `deb file:` måste du själv kontrollera om platsen som de refererar till innehåller ett arkiv för sarge eller etch.

Viktigt! Ändra inte några rader som börjar med `deb cdrom:`. Om du gör det så ogiltigförklaras raden och du måste köra `apt-cdrom` igen. Bli inte rädd om en "cdrom"-källrad refererar till "unstable". Även om det är förvirrande, är det normalt.

Om du har gjort några ändringar, spara filen och kör

```
# apt-get update
```

för att uppdatera paketlistan.