

# FAQ Debian GNU/Linux

Автори перераховані в Debian FAQ Authors

версія CVS від 19 червня 2006 року

## Анотація

Цей документ дає відповіді на часті запитання про Debian GNU/Linux.

## Авторські права

Copyright © 1996-2005 by Software in the Public Interest

Permission is granted to make and distribute verbatim copies of this document provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this document under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this document into another language, under the above conditions for modified versions, except that this permission notice may be included in translations approved by the Free Software Foundation instead of in the original English.

# Зміст

1	Означення та загальний огляд	1
1.1	Чим є ці FAQ?	1
1.2	Що таке Debian GNU/Linux?	1
1.3	Гаразд, я вже знаю, що таке Debian... а що таке Linux?	2
1.4	Чи Debian створюється лише для GNU/Linux?	3
1.5	Чим Debian GNU/Linux відрізняється від інших Linux-дистрибутивів? Чому я повинен серед інших дистрибутивів обрати саме Debian?	3
1.6	Як проект Debian відноситься до Фонду вільного програмного забезпечення GNU, вони співпрацюють?	5
1.7	Як вимовляється слово „Debian“ та що воно означає?	5
2	Отримання та встановлення Debian GNU/Linux	7
2.1	Яка версія Debian є найсвіжішою?	7
2.2	Чи оновлюються пакунки у стабільній збірці?	8
2.3	Де і як я можу отримати встановлювальні диски Debian?	8
2.4	Як мені встановити Debian з компакт-диску?	8
2.5	Чому компакт-диски з офіційним стабільним випуском містять символічні посилання „frozen“ та „unstable“? Я думав, що на цьому компакт-диску тільки стабільний випуск!	9
2.6	Я маю власний записувач компакт-дисків, чи є десь образи КД?	9
2.7	Чи можу я встановити її з купи дискет?	9
2.8	Чи можу я отримати та встановити Debian прямо з Інтернету?	10
3	Питання сумісності	11
3.1	На яких апаратних архітектурах та системах запускається Debian GNU/Linux?	11
3.2	Наскільки Debian сумісний з іншими збірками Linux?	12

3.3	Наскільки джерельні коди Debian сумісні з іншими Unix-системами? . . . . .	12
3.4	Чи можу я використовувати пакунки Debian (файли „.deb“) в моїй системі RedHat/Slackware/. . . Linux? Чи можу я використовувати пакунки RedHat (файли „.rpm“) у моїй Debian GNU/Linux? . . . . .	13
3.5	Чи зможе Debian запустити мої старі libc5-програми? . . . . .	13
3.6	Чи можна використовувати Debian для компіляції libc5-програм? . . . . .	13
3.7	Як я можу встановити не-Debian програму? . . . . .	14
3.8	Чому я не можу скомпілювати програми, що вимагають libtermcap? . . . . .	14
3.9	Чому я не можу встановити AccelX? . . . . .	14
3.10	Чому мої старі XFree 2.1 Motif-додатки призводять до збоїв у роботі? . . . . .	14
4	Програмне забезпечення, доступне в Debian. . . . .	17
4.1	Які типи додатків та програмного забезпечення для розробників доступні в Debian GNU/Linux? . . . . .	17
4.2	Хто написав всі ці програми? . . . . .	18
4.3	Як я можу отримати поточний список програм для Debian? . . . . .	18
4.4	Чого немає в Debian GNU/Linux? . . . . .	18
4.5	Чому при компіляції програм я отримую повідомлення „ld: cannot find -lfoo“? Чому у двійкових пакунках бібліотек немає файлів libfoo.so? . . . . .	18
4.6	Як Debian підтримує Java і чи підтримує взагалі? . . . . .	19
4.7	Як я можу перевірити, чи я використовую Debian і якої саме версії? . . . . .	19
4.8	Як справи в Debian з підтримкою інших мов, окрім англійської? . . . . .	19
4.9	Як щодо обмежень законодавства США на експорт? . . . . .	20
4.10	Куди подівся pine? . . . . .	20
4.11	А де qmail/ezmlm/djbdns? . . . . .	21
5	FTP-архіви Debian . . . . .	23
5.1	Що означають всі ті теки на ftp-архівах Debian? . . . . .	23
5.2	Скільки збірок Debian знаходиться в теці dists? . . . . .	23
5.3	Що означають всі ці імена на кшталт slink, potato та ін.? . . . .	24
5.3.1	Які кодові назви вже використовувались? . . . . .	24
5.3.2	Звідки беруться всі ці імена? . . . . .	24
5.4	Ну а як щодо „sid“? . . . . .	25

5.5	Що міститься в теці <code>stable</code> ?	25
5.6	Що міститься в теці <code>testing</code> ?	25
5.6.1	Як щодо <code>testing</code> ? Чому вона заморожується?	26
5.7	Що знаходиться в теці <code>unstable</code> ?	26
5.8	Що це за теки в <code>dists/stable/main</code> ?	26
5.9	А де джерельні коди?	27
5.10	Що за тека — <code>pool</code> ?	27
5.11	Що таке <code>incoming</code> ?	28
5.12	Як мені створити своє власне <code>apt</code> -сумісне сховище?	28
6	Основи системи управління пакунками Debian	29
6.1	Що таке пакунок Debian?	29
6.2	Який формат двійкових пакунків Debian?	30
6.3	Чому назви пакунків Debian такі довгі?	30
6.4	Що то за файл <code>control</code> ?	31
6.5	Для чого призначені <code>conffile</code> ?	32
6.6	Що роблять сценарії <code>preinst</code> , <code>postinst</code> , <code>prerm</code> , та <code>postrm</code> ?	32
6.7	Що це за пакунки — Essential, Required, Important, Standard, Optional та Extra (Невіддільний, Необхідний, Важливий, Типовий, Необов'язковий та Додатковий)?	33
6.8	Що таке віртуальний пакунок?	34
6.9	Що мається на увазі, коли кажуть, що пакунок залежить, рекомендує, пропонує, конфліктує, замінює або забезпечує ( <code>Depends</code> , <code>Recommends</code> , <code>Suggests</code> , <code>Conflicts</code> , <code>Replaces</code> або <code>Provides</code> ) інший пакунок?	34
6.10	Що означає „попередня залежність“?	35
6.11	Що мається на увазі, кажучи невідомий, встановлений, видалений, очищений чи зафіксований ( <code>unknown</code> , <code>install</code> , <code>remove</code> , <code>purge</code> та <code>hold</code> ) про стан пакунку.	36
6.12	Як мені зафіксувати пакунок?	36
6.13	Як мені встановити джерельний пакунок?	37
6.14	Як мені скомпонувати двійковий пакунок з джерельного?	37
6.15	А як мені створити власний пакунок Debian?	38

7	Інструменти для керування пакунками Debian	39
7.1	Які програми пропонує Debian для керування пакунками?	39
7.1.1	dpkg	39
7.1.2	APT	40
7.1.3	aptitude	42
7.1.4	dselect	42
7.1.5	Інші інструменти для керування пакунками	45
7.2	Debian стверджує, що може оновити запущену програму; яким чином він це робить?	46
7.3	Як мені дізнатись, які пакунки встановлено в системі?	46
7.4	Як мені дізнатись, що за пакунок встановив котрийсь файл?	46
7.5	Чому не видаляється foo-data під час видалення foo? Як мені очистити систему від старих пакунків, що вже не використовуються?	47
8	Підтримування вашої системи Debian в сучасному стані	49
8.1	Як я можу оновити свою збірку Debian 1.3.1 (чи давнішу), що базується на libc5, до 2.0 (чи новішої), котрі базуються на libc6?	49
8.2	Як я можу підтримувати систему Debian у найновішому стані?	50
8.2.1	aptitude	50
8.2.2	apt-get, dselect та apt-cdrom	51
8.2.3	dpkg-ftp	52
8.2.4	mirror	52
8.2.5	dpkg-mountable	52
8.3	Чи повинен я переходити у однокористувацький режим при оновленні пакунків?	53
8.4	Чи потрібно мені зберігати на диску всі ці .deb-файли?	53
8.5	Як я можу вести список всіх пакунків, котрі я встановив у системі? Я хотів би знати коли і які оновлення або видалення відбувались!	53
9	Debian і ядро	55
9.1	Чи можу я встановити та скомпілювати ядро без специфічних для Debian вдосконалень?	55
9.2	Які інструменти пропонує Debian для компонування нестандартних ядер?	55
9.3	Як мені створити нестандартну завантажувальну дискету?	56
9.4	Які особливі можливості пропонує Debian для роботи з модулями ядра?	57
9.5	Чи можу я безпечно видалити старий пакунок ядра, і якщо можу, то як?	57

10	Налаштування вашої Debian GNU/Linux	59
10.1	Як мені переконатись, що всі програми використовують однаковий розмір паперу? . . . . .	59
10.2	Як мені надати доступ до периферійних пристроїв не ставлячи під загрозу безпеку системи? . . . . .	59
10.3	Яким чином в Debian задати консольний шрифт при завантаженні? . . . . .	59
10.4	Як я можу змінити стандартні настройки додатків для X11? . . . . .	60
10.5	Схоже, що кожна збірка завантажується по-своєму. Розкажіть, як це робить Debian. . . . .	60
10.6	Виходить, Debian не використовує <code>gc.local</code> для налаштування процесу завантаження; що ж в такому випадку використовується? . . . . .	61
10.7	Що система керування робить з пакунками, котрі містять конфігураційні файли інших пакунків? . . . . .	61
10.8	Як я можу перезаписати файл, встановлений пакунком, так щоб використовувалась інша його версія? . . . . .	62
10.9	Як мені додати свій пакунок до списку відомих системі керування пакунками? . . . . .	63
10.10	Деяким користувачам подобається <code>mawk</code> , іншим <code>gawk</code> ; одні користуються <code>vim</code> , інші <code>elvis</code> ; дехто уподобав <code>trn</code> , а дехто <code>tin</code> ; як Debian підтримує такі різноманітності? . . . . .	63
11	Отримання підтримки для Debian GNU/Linux	65
11.1	Яка ще документація існує для системи Debian? . . . . .	65
11.2	Чи є якісь мережеві ресурси для обговорення Debian? . . . . .	66
11.2.1	Списки розсилки . . . . .	66
11.2.2	Супроводжуючі . . . . .	67
11.2.3	Групи новин . . . . .	68
11.3	Чи є якийсь швидкий спосіб пошуку інформації про Debian GNU/Linux? . . . . .	68
11.4	Чи ведуться записи про відомі помилки? . . . . .	68
11.5	Як мені відправити в Debian звіт про помилку? . . . . .	69
12	Співробітництво в рамках проекту Debian	71
12.1	Як я можу стати розробником програмного забезпечення Debian? . . . . .	71
12.2	Як я можу допомогти ресурсами проекту Debian? . . . . .	71
12.3	Як я можу допомогти фінансово проекту Debian? . . . . .	72
12.3.1	Software in the Public Interest (Програми для загалу) . . . . .	72
12.3.2	Free Software Foundation (Фонд вільного програмного забезпечення) . . . . .	72

---

13	Перепозирення Debian GNU/Linux як комерційного продукту	73
13.1	Чи можу я створювати та продавати компакт-диски Debian? . . . . .	73
13.2	Чи може бути Debian запакований разом з закритим програмним забезпеченням? . . . . .	73
13.3	Я створюю особливу збірку Linux для „вертикального маркетингу“. Чи можу я використовувати Debian GNU/Linux як базову систему та додати мої програми поверх неї? . . . . .	74
13.4	Чи можу я запакувати комерційну програму у пакунок Debian щоб її можна було легко встановити на будь-якій Debian-системі? . . . . .	74
14	Зміни, що очікуються у новій версії Debian	75
14.1	Покращена безпека . . . . .	75
14.2	Вдосконалена підтримка неангломовних користувачів . . . . .	75
14.3	Більше архітектур . . . . .	75
14.4	Більше ядер . . . . .	76
15	Загальна інформація про FAQ	77
15.1	Автори . . . . .	77
15.2	Зворотній зв'язок . . . . .	77
15.3	Доступність . . . . .	78
15.4	Формат документу . . . . .	78



## Розділ 1

# Означення та загальний огляд

### 1.1 Чим є ці FAQ?

Цей документ містить часті питання (та відповіді на них!), щодо дистрибутива (Debian GNU/Linux та інших) та проекту Debian. По можливості, будуть вказуватись посилання на іншу документацію: нам не хотілося б цитувати тут великі частини зовнішньої документації. Ви помітите, що для розуміння деяких відповідей будуть необхідними деякі початкові знання про Unix-подібні операційні системи. Ми намагатимемось, по можливості, вимагати від читача лише мінімальної кількості попередніх знань: відповіді на загальні питання початківців будуть якомога простішими.

Якщо ви не змогли знайти те, що вас цікавить, в цих FAQ, переконайтесь, що переглянули ‘Яка ще документація існує для системи Debian?’ на стор. [65](#). Якщо навіть це не допомогло, див. ‘Зворотній зв’язок’ на стор. [77](#).

### 1.2 Що таке Debian GNU/Linux?

Debian GNU/Linux — це специфічна збірка операційної системи Linux та величезна кількість пакунків, що запускаються під нею.

В принципі, користувачі можуть отримати ядро Linux через Інтернет чи яким-небудь іншим шляхом та скомпілювати його самостійно. Так само вони можуть роздобути джерельні коди численних додатків, скомпілювати програми та встановити їх до своєї системи. Для великих програм цей процес може бути не лише довготривалим, але й зі значною кількістю помилок. Щоб уникнути цього, користувачі як правило обирають собі операційну систему та набір пакунків від одного з поширювачів Linux. Вони відрізняються набором програм, протоколів та підходів, котрі використовуються для пакування, встановлення та керування пакунками додатків у вашій системі в поєднанні з інструментами встановлення і налаштування, підтримкою, документацією й іншими послугами.

Debian GNU/Linux — це результат зусиль добровольців по створенню вільної, високоякісної Unix-сумісної операційної системи з набором численних додатків. Ідея вільної Unix-системи

породжена проектом GNU. Багато інших програм, що роблять Debian GNU/Linux таким зручним, були розроблені саме в рамках цього проекту.

Для Debian слово „вільний“ має таке ж значення, як його трактує GNU (перегляньте Критерії Debian щодо вільного програмного забезпечення ([http://www.debian.org/social\\_contract#guidelines](http://www.debian.org/social_contract#guidelines))). Коли ми говоримо про вільне програмне забезпечення, ми маємо на увазі саме свободу, а не ціну. Вільне програмне забезпечення означає, що ви можете поширювати його копії, мати джерельні коди, чи отримати їх при бажанні; що ви можете вносити зміни в тексти програм чи використовувати його частину у інших програмних проектах, і що ви знаєте про ці свої можливості.

Проект Debian започатковано Яном Мардоком (Ian Murdock) у 1993 році за початкової фінансової підтримки проектом Вільного програмного забезпечення GNU (FSF GNU). Зараз розробники Debian вважають його прямим дітищем проекту GNU.

Debian GNU/Linux це:

- функціональна повнота: на даний момент Debian містить понад 15400 пакунків. Користувачі можуть обирати, які пакунки їм встановлювати; для цього Debian має необхідні інструменти. Ви можете переглянути список та описи доступних на даний момент пакунків на будь-якому з дзеркальних сайтів (<http://www.debian.org/distrib/ftplist>) Debian.
- свобода використання та подальшого поширення: не вимагається ніякого членства в консорціумі або оплати для того, щоб брати участь у розповсюдженні чи розробці. Всі пакунки, що формально є частиною Debian GNU/Linux можна вільно поширювати, як правило на засадах Загальної публічної ліцензії GNU.  
Ftp-архіви Debian містять також близько 450 програмних пакунків (в секціях non-free та contrib), що поширюються на особливих умовах, котрі описані в кожному з пакунків.
- динамічність: з 1649 добровольцями, що постійно додають новий та вдосконалений код, Debian швидко розвивається. Нові випуски планується робити кожних кілька місяців, а ftp-архіви поновлюються щоденно.

Окрім того, що Debian GNU/Linux сама по собі є вільним програмним забезпеченням, вона є базою, на основі якої формуються численні додаткові збірки. Пропонуючи повноцінну надійну систему, Debian пропонує користувачам Linux дедалі кращу сумісність та дозволяє дистрибуторам вилучати непотрібні їм можливості і концентруватись на потрібних для створення спеціалізованих дистрибутивів. Перегляньте ‘Я створюю особливу збірку Linux для „вертикального маркетингу“. Чи можу я використовувати Debian GNU/Linux як базову систему та додати мої програми поверх неї?’ на стор. 74, щоб отримати більше інформації.

### 1.3 Гаразд, я вже знаю, що таке Debian... а що таке Linux?

Якщо коротко, Linux — це ядро Unix-подібної операційної системи. Спочатку воно розроблялось для ПК з процесорами 386 (і вище); зараз перенесено на значну кількість інших

систем, в тому числі багатопроцесорних, та продовжує інтенсивно розвиватись. Linux написано Лінусом Торвальдсом (Linus Torvalds) та багатьма іншими комп'ютерними спеціалістами з усього світу.

Окрім ядра система Linux зазвичай має:

- файлову систему, сумісну зі Стандартом ієрархії файлової системи Linux (Linux Filesystem Hierarchy Standard, <http://www.pathname.com/fhs/>).
- велику кількість утиліт Unix, більшість з яких була розроблена проектом GNU та Фондом вільного програмного забезпечення (FSF, Free Software Foundation).

Сукупність ядра Linux, файлової системи та утиліт GNU, FSF і інших, компонується таким чином, щоб забезпечити сумісність зі стандартом POSIX (IEEE 1003.1); перегляньте 'Наскільки джерельні коди Debian сумісні з іншими Unix-системами?' на стор. 12.

Щоб більше дізнатися про Linux, перегляньте Інформаційну сторінку Linux (<ftp://ibiblio.org/pub/Linux/docs/HOWTO/INFO-SHEET>) Міхаеля К. Джонсона (Michael K. Johnson) та Meta-FAQ (<ftp://ibiblio.org/pub/Linux/docs/HOWTO/META-FAQ>).

## 1.4 Чи Debian створюється лише для GNU/Linux?

На сьогодні Debian доступний лише під Linux, але з Debian GNU/Hurd і Debian'ом на ядрах BSD ми почали пропонувати також не-Linux операційні системи в якості серверних рішень, платформ для розробки або для робочого столу. Однак, ці не-Linux портування все ж не є офіційними випусками.

Найдавнішою спробою з портування Debian є Debian GNU/Hurd.

Hurd — це ряд серверів, запущених на мікроядрі GNU Mach. Разом вони складають основу операційної системи GNU.

Будь ласка, перегляньте <http://www.gnu.org/software/hurd/>, щоб отримати загальну інформацію про GNU/Hurd, або <http://www.debian.org/ports/hurd/>, якщо бажаєте дізнатись більше про Debian GNU/Hurd.

Спроба номер два — це портування на ядро BSD. Люди працюють з ядрами NetBSD та FreeBSD.

Про ці не-linux перенесення Debian ви можете більше детально дізнатись на сторінці <http://www.debian.org/ports/#nonlinux>.

## 1.5 Чим Debian GNU/Linux відрізняється від інших Linux-дистрибутивів? Чому я повинен серед інших дистрибутивів обрати саме Debian?

Від інших збірок Linux Debian відрізняють такі особливості:

Свобода: Як затверджено в Суспільному договорі Debian ([http://www.debian.org/social\\_contract](http://www.debian.org/social_contract)), Debian залишиться 100% вільним. Debian є дуже суворим в питаннях визнання програмного забезпечення „істинно вільним“. Критерії, що використовуються для визначення „вільності“ роботи, виділяють Вільне програмне забезпечення Debian ([http://www.debian.org/social\\_contract#guidelines](http://www.debian.org/social_contract#guidelines)).

Система підтримки пакунків Debian: Вся система, чи будь-який її компонент можуть бути оновлені одразу ж без будь-яких перетворень, без втрати звичних конфігураційних файлів та (в більшості випадків) без перезавантажень системи. Більшість доступних сьогодні збірок Linux мають у своєму складі систему підтримки пакунків, проте в Debian вона справді унікальна та особливо надійна (див. ‘Основи системи управління пакунками Debian’ на стор. 29).

Відкрита розробка: В той час як інші дистрибутиви Linux розробляються окремими людьми, маленькими закритими групами або ж комерційними поставниками, Debian є єдиною великою збіркою, що спільно розробляється великою кількістю різних людей через Інтернет, так само, як Linux та інше вільне програмне забезпечення.

Понад 1649 добровільних супроводжуючих працюють над більш ніж 15400 пакунками та покращенням Debian GNU/Linux. Розробники Debian переважно беруть участь у проекті не написанням нових програм, але пакуванням існуючих згідно стандартів проекту, формуванням звітів про помилки оригінальним розробникам, та забезпеченням підтримки користувачів. Для додаткової інформації щодо того, як стати розробником, перегляньте ‘Як я можу стати розробником програмного забезпечення Debian?’ на стор. 71.

Універсальна операційна система: Debian постачається з більш ніж 15400 пакунками (<http://packages.debian.org/stable/>) та запускається на 11 архітектурах (<http://www.debian.org/ports/>). Це більше, ніж в будь-якій іншій збірці GNU/Linux. Див. ‘Які типи додатків та програмного забезпечення для розробників доступні в Debian GNU/Linux?’ на стор. 17, щоб переглянути список програмного забезпечення, що підтримується, або ‘На яких апаратних архітектурах та системах запускається Debian GNU/Linux?’ на стор. 11, щоб переглянути описи платформ апаратного забезпечення, які підтримуються.

Система відслідковування помилок: Географічна розпорошеність розробників Debian вимагає витончених інструментів, швидкого поширення інформації про помилки та способи їх усунення для прискорення розвитку системи. Користувачів заохочують відправляти звіти про помилки у стандартному вигляді, що дозволяє за короткий час оприлюднити їх через тенета або електронну пошту. Додаткову інформацію з приводу використання журналу помилок можна знайти в цьому FAQ, перегляньте ‘Чи ведуться записи про відомі помилки?’ на стор. 68.

Політика Debian: Debian має широку специфікацію стандартів якості, так звану „Політику Debian“. Цей документ визначає якість та стандарти, котрих ми дотримуємось у пакунках.

Веб-сторінка про причини вибору Debian ([http://www.debian.org/intro/why\\_debian](http://www.debian.org/intro/why_debian)) містить додаткову інформацію з цих питань.

## 1.6 Як проект Debian відноситься до Фонду вільного програмного забезпечення GNU, вони співпрацюють?

Система Debian будується на ідеалах вільного програмного забезпечення, що проповідуються Фондом вільного програмного забезпечення (<http://www.gnu.org/>) та особливо Річардом Столменом (<http://www.stallman.org/>) (Richard Stallman). Потужна система інструментів розробки Фонду ВПЗ, утиліти та додатки також є ключовими частинами системи Debian.

Проект Debian незалежний від Фонду ВПЗ, проте ми регулярно спілкуємось та співпрацюємо у різноманітних проектах. Фонд ВПЗ попросив, щоб ми називали нашу систему „Debian GNU/Linux“ і ми з радістю виконуємо це прохання.

Стратегічною метою Фонду ВПЗ є розробка нової операційної системи під назвою GNU, що базується на Hurd (<http://www.gnu.org/software/hurd/>). Debian співпрацює з Фондом ВПЗ над цією системою, що має назву Debian GNU/Hurd (<http://www.debian.org/ports/hurd/>).

## 1.7 Як вимовляється слово „Debian“ та що воно означає?

Назва проекту вимовляється як „Дебієн“ (Deb'-ee-en) з коротким „е“ в Deb та наголосом на першому складі. Це слово є поєднанням імен Дебри та Яна Мердоків (Debra та Ian Murdock), котрі заснували проект. (Схоже, словники допускають деяку невизначеність щодо вимови імені „Ian“, проте сам він схиляється до „Ієн“ („ee'-en“)).



## Розділ 2

# Отримання та встановлення Debian GNU/Linux

Офіційною документацією, в якій можна знайти вказівки щодо встановлення системи є Посібник по встановленню Debian GNU/Linux (<http://www.debian.org/releases/stable/installmanual>). Ми ж тут звернемо увагу на деякі загальні речі щодо отримання та встановлення Debian GNU/Linux.

### 2.1 Яка версія Debian є найсвіжішою?

Зараз є три версії Debian GNU/Linux:

версія 3.1, відома також як „стабільна“ збірка, *stable* Це стабільне та добре перевірене програмне забезпечення; вона випускається після внесення значної кількості виправлень, що стосуються безпеки та зручності у використанні.

„тестова“ збірка, *testing* Саме сюди вносяться пакунки, котрі будуть випущені у наступній стабільній версії; вони вже пройшли деяку перевірку у нестабільній версії, проте ще не до кінця готові для випуску. Ця збірка оновлюється значно частіше, аніж стабільна, але рідше, аніж нестабільна.

„нестабільна“ збірка, *unstable* Ця версія завжди знаходиться в процесі розробки; вона поновлюється постійно. Ви можете отримати пакунки з нестабільного архіву з будь-якого з ftp-сайтів Debian та використовувати їх для оновлення вашої системи у будь-який момент, але не очікуйте, що ваша система буде настільки ж зручна та стабільна, як раніше — ось чому вона називається „нестабільною“!

Будь ласка, перегляньте ‘Скільки збірок Debian знаходиться в теці dists?’ на стор. 23, щоб отримати більше інформації.

## 2.2 Чи оновлюються пакунки у стабільній збірці?

До стабільного випуску не додається жодної нової функціональності. Як тільки версія Debian випущена та позначена як стабільна, для неї робляться лише оновлення безпеки. Тому після випуску оновлюються лише пакунки, в котрих були знайдені проблеми з безпекою. Всі оновлення безпеки обслуговуються security.debian.org ([ftp://security.debian.org](http://security.debian.org)).

Оновлення безпеки слугують одній меті: виправити помилки, що загрожують безпеці. Вони не є методом внесення прихованих додаткових змін до стабільної збірки без проходження стандартної процедури. А тому виправлення помилок у пакунках, що мають проблеми із безпекою, не призводить до оновлення їх до наступної версії. Команда безпеки Debian вносить необхідні зміни саме в ті версії програм, котрі є у стабільній збірці.

Щоб отримати більше інформації щодо підтримки безпеки, будь ласка, прочитайте FAQ щодо безпеки (<http://www.debian.org/security/faq>) або Підручник по безпеці Debian (<http://www.debian.org/doc/manuals/securing-debian-howto/>).

## 2.3 Де і як я можу отримати встановлювальні диски Debian?

Ви можете отримати встановлювальні диски, завантаживши необхідні файли з одного з дзеркал Debian (<http://www.debian.org/mirror/list>).

Встановлювальні системні файли розділені в підтеках теки dists/stable/main і назви цих підтек вказують на архітектуру вашої системи наступним чином: disks-arch, де arch може бути „i386“, „sparc“ і так далі; повний список дивіться на сайті. В кожній з цих підтек може бути ще декілька підтек — по одній для кожної версії; поточна версія знаходиться також в теці current (це символічне посилання).

За подальшими інструкціями зверніться до файлу README.txt у відповідній теці.

## 2.4 Як мені встановити Debian з компакт-диску?

Linux підтримує файлову систему ISO 9660 (CD-ROM) з розширеннями Rock Ridge (понайбільше відому як „High Sierra“). Деякі постачальники (<http://www.debian.org/CD/vendors/>) пропонують Debian GNU/Linux в такому форматі.

Увага: Під час встановлення з компакт-диску не є хорошою ідеєю вибирати метод доступу cdrom для dselect, оскільки він є дуже повільним. Методи mountable та apt, наприклад, є значно кращими для встановлення з компакт-диску (див. ‘dpkg-mountable’ на стор. 52 та ‘apt-get, dselect та apt-cdrom’ на стор. 51).



## 2.5 Чому компакт-диски з офіційним стабільним випуском містять символічні посилання „frozen“ та „unstable“? Я думав, що на цьому компакт-диску тільки стабільний випуск!

Офіційні КД Debian дійсно містять символічні посилання. Щось таке:

```
/dists/frozen -> sarge/  
/dists/stable -> sarge/  
/dists/testing -> sarge/  
/dists/unstable -> sarge/
```

Отже, вони спрацюють, якщо ви зробите у файлі `sources.list` запис, подібний до наступного:

```
deb cdrom:[<назва мітки на КД>]/ unstable main [...]
```

Той факт, що вказані символічні посилання існують, не означає, що образ — це „unstable“, „testing“ або ще щось. Щоб дізнатись яка версія Debian знаходиться на компакт-диску, можна прочитати його мітку в файлі `/.disk/info`. Ця інформація також є в `/README.txt`.

Переглянувши сторінку <http://www.debian.org/releases/>, ви зможете дізнатись який випуск на даний момент є стабільним чи тестовим.

## 2.6 Я маю власний записувач компакт-дисків, чи є десь образи КД?

Так. Щоб полегшити постачальникам доступ до високоякісних дисків ми пропонуємо офіційні образи компакт-дисків (<http://cdimage.debian.org/>).

## 2.7 Чи можу я встановити її з купи дискет?

Перш за все зауважте: вся Debian GNU/Linux є занадто великою, щоб встановлювати її з таких малих носіїв, як стандартні дискети 1.44 МБ — такий досвід навряд чи видасться вам особливо приємним.

Скопіюйте пакунки Debian на форматовані дискети. Підійде як формат DOS, так і звичний для Linux „ext2“, або „minix“; вам потрібно буде лише правильно ввести команду `mount`.

Використання дискет тягне за собою наступні обмеження:

- Короткі назви файлів для MS-DOS: якщо ви хочете помістити пакунки Debian на дискети, форматовані в MS-DOS, ви побачите, що їх назви в переважній більшості занадто довгі і не влізають в обмеження „вісім крапка три“ для файлів MS-DOS. Щоб обійти це, вам потрібно буде використовувати диски, форматовані у VFAT, котра підтримує довгі назви файлів.

- Великий розмір файлів: багато з пакунків більші за 1.44 МБ і не помістяться на одну дискету. Щоб вирішити цю проблему, використовуйте інструмент `dpkg-split` (див. ‘`dpkg-split`’ на стор. 45), що знаходиться у теці `tools` на дзеркалах Debian (<http://www.debian.org/mirror/list>).

У вашому ядрі повинна бути увімкнена підтримка читання/запису дискет; у більшості ядер вона підключена за замовчуванням.

Щоб примонтувати дискету у точку монтування `/floppy` (тека, що повинна створюватись під час встановлення), вживайте:

- `mount -t msdos /dev/fd0 /floppy/`  
якщо дискета знаходиться в пристрої A: та має файлову систему MS-DOS,
- `mount -t msdos /dev/fd1 /floppy/`  
якщо дискета знаходиться в пристрої B: та має файлову систему MS-DOS,
- `mount -t ext2 /dev/fd0 /floppy/`  
якщо дискета знаходиться в пристрої A: та має файлову систему `ext2` (типову для Linux).

## 2.8 Чи можу я отримати та встановити Debian прямо з Інтернету?

Так. Ви можете завантажити систему встановлення Debian з нашого FTP сайту чи його дзеркал.

Вам потрібно завантажити невеликий файл образу компакт-диску, створити з нього завантажувальний компакт-диск, і встановити базову систему з нього та решту через мережу. Щоб отримати додаткову інформацію, будь ласка, перегляньте сторінку <http://www.debian.org/CD/netinst/>.

Ви також можете завантажити навіть маленькі образи дискет, створити з них завантажувальні дискети, розпочати процедуру встановлення та отримати решту елементів системи Debian через мережу. За додатковою інформацією звертайтеся, будь ласка, до <http://www.debian.org/distrib/floppyinst>.

## Розділ 3

# Питання сумісності

### 3.1 На яких апаратних архітектурах та системах запускається Debian GNU/Linux?

Debian GNU/Linux містить повні джерельні коди для всіх включених в неї програм, тож вона повинна працювати на всіх системах, що підтримуються ядром Linux. Зверніться до Linux FAQ (<http://en.tldp.org/FAQ/Linux-FAQ/intro.html#DOES-LINUX-RUN-ON-MY-COMPUTER>) для детальнішого розгляду цього питання.

Поточна версія Debian GNU/Linux, 3.1, містить повні двійкові збірки для наступних архітектур:

i386: це всі ПК, основані на процесорах Intel та сумісних з ними, включаючи Intel 386, 486, Pentium, Pentium Pro, Pentium II (Klamath і Celeron), та Pentium III, та більшість сумісних процесорів виробництва AMD, Cyrix та інших.

m68k: це машини Amiga та ATARI з процесорами Motorola 680x0 для  $x \geq 2$ , з MMU.

alpha: системи Alpha виробництва Compaq/Digital.

sparc: системи виробництва Sun, SPARC та більшість систем UltraSPARC.

powerpc: деякі системи PowerPC виробництва IBM/Motorola включно з CHRP, PowerMac і PReP.

arm: системи ARM і StrongARM.

mips: системи SGI MIPS, Indy та Indigo2; mipsel: little-endian MIPS-системи, Digital DECstation..

hppa: комп'ютери PA-RISC виробництва Hewlett-Packard (712, C3000, L2000, A500).

ia64: комп'ютери Intel IA-64 („Itanium“).

s390: Великі системи IBM S/390.

Розглядається питання розробки двійкових збірок Debian для архітектури Sparc64 (UltraSPARC).

З інших питань про завантаження, поділ вашого диску на розділи, підключення пристроїв PCMCIA та подібних, будь ласка, користайтесь інструкціями, викладеними у Підручнику по встановленню, що доступний в тенетах за адресою <http://www.debian.org/releases/stable/installmanual>.

## 3.2 Наскільки Debian сумісний з іншими збірками Linux?

Розробники Debian взаємодіють з авторами інших збірок Linux щоб досягти двійкової сумісності між дистрибутивами Linux. Більшість комерційних програм під Linux запускаються в Debian так само добре, як і в системі, для котрої вони були розроблені.

Debian GNU/Linux дотримується стандарту ієрархії файлової системи Linux (<http://www.pathname.com/fhs/>). Оскільки деякі положення стандарту можна трактувати по-різному, існують відмінності між Debian та іншими Linux-системами.

Debian GNU/Linux підтримує програмне забезпечення, розроблене в рамках стандарту Linux Standard Base (LSB) (<http://www.linuxbase.org/>). LSB — це специфікація, що дозволяє одному й тому ж двійковому пакунку використовуватись в різних дистрибутивах. Пакунки для випуску Debian Etch не повинні протирічити вимогам LSB, версії 1.3. На момент написання цього тексту Debian GNU/Linux формально не є LSB-сертифікованою. Однак, деякі збірки, похідні від Debian, є такими. Дискусія та координація зусиль з приводу приведення Debian до відповідності вимогам LSB має місце в списку розсилки [debian-lsb](http://lists.debian.org/debian-lsb/) (<http://lists.debian.org/debian-lsb/>).

## 3.3 Наскільки джерельні коди Debian сумісні з іншими Unix-системами?

У більшості Linux-додатків джерельні коди сумісні з іншими Unix-системами. Вони підтримуються майже усіма доступними системами Unix System V та вільними і комерційними BSD-подібними системами. Проте у світі Unix такі заяви можна робити лише приблизно, оскільки немає способів перевірити їх. У площині розробки програмного забезпечення потрібна повна сумісність, а не „сумісність у більшості випадків“. Тож потреба у стандарті виникла давно і зараз POSIX.1 (IEEE Standard 1003.1-1990) є одним з основних стандартів сумісності джерельних кодів у Unix-подібних операційних системах.

Linux має намір дотримуватись стандарту POSIX.1, але стандарти POSIX вартують неабияких грошей і сертифікація POSIX.1 (та FIPS151-2) є доволі дорогою; це значно ускладнює роботу розробників Linux стосовно повної відповідності стандартам POSIX. Через вартість сертифікації схоже, що Debian не отримає офіційне сертифіковане підтвердження, навіть якщо він повністю пройде випробувальні тести. З недавнього часу перевірочний комплект знаходиться у вільному доступі, тож очікується, що все більше людей будуть писати програми у відповідності до вимог POSIX.1.

Unifix GmbH (Брауншвайг, Німеччина) розробили систему Linux, що була сертифікована FIPS 151-2 (надмножина POSIX.1). Ця технологія була доступна у власній збірці Unifix, що отримала назву Unixfix Linux 2.0 та у Linux-FT виробництва Lasermoon.

### 3.4 Чи можу я використовувати пакунки Debian (файли „deb“) в моїй системі RedHat/Slackware/... Linux? Чи можу я використовувати пакунки RedHat (файли „rpm“) у моїй Debian GNU/Linux?

Різні збірки Linux використовують різні формати пакунків та різні програми для керування ними.

Ви, ймовірно, зможете: Якщо у „чужій“ збірці встановлена та налаштована програма для розпаковування пакунків Debian, то ви зможете розпакувати пакунок в такій системі. Зворотне твердження також вірне, тобто ви зможете розпакувати та правильно розташувати файли з пакунку RedHat чи Slackware на комп'ютері з Debian GNU/Linux за допомогою спеціальної програми. Це відбувається завдяки існуванню та підтримці Стандарту ієрархії файлової системи Linux. Для перетворення між різними форматами пакунків використовують пакунок `alien` (<http://packages.debian.org/alien>).

Ви, мабуть, не захочете: Більшість програм для керування пакунками використовують спеціальні керуючі файли для стискання та пакування архівів. Ці файли не є стандартизованими. А тому результат розпаковування пакунку Debian на чужорідній системі може мати непередбачувані (як правило, руйнівні) наслідки для керівника пакунків даної системи. Аналогічно, інструменти з інших збірок можуть успішно розпакуватися в Debian, але при цьому призведуть до помилок в системі керування пакунками, коли буде потрібно оновити систему чи видалити якийсь інший пакунок або ж навіть просто при виводі списку встановлених пакунків.

Кращий шлях: Стандарт файлової системи Linux (а отже і Debian GNU/Linux) вимагає, щоб підтеки в `/usr/local/` були віддані на розсуд користувача. Таким чином користувач може розпакувати чужорідні пакунки сюди, а далі налаштовувати і видаляти їх окремо.

### 3.5 Чи зможе Debian запустити мої старі libc5-програми?

Так. Просто встановіть необхідні libc5 з секції `oldlibs`, котра містить застарілі пакунки, що були включені для сумісності з старими програмами..

### 3.6 Чи можна використовувати Debian для компіляції libc5-програм?

Так. Встановіть пакунки `libc5-altdev` та `altgcc` з секції `oldlibs`. В теці `/usr/i486-linuxlibc1/bin` ви знайдете спеціальний gcc та g++. Додайте її до вашої змінної `$PATH` щоб make та інші програми могли їх викликати.

Майте на увазі, що середовище libc5 не повністю підтримується іншими нашими пакунками.

### 3.7 Як я можу встановити не-Debian програму?

Файли в теці `/usr/local` не контролюються системою керування пакунків Debian. Тому гарним вибором буде розміщення джерельних кодів таких програм в `/usr/local/src/`. Наприклад, ви можете розпакувати файли з архіву `foo.tar` до теки `/usr/local/src/foo/`. Після того, як ви скомпілюєте їх, помістіть двійкові файли до `/usr/local/bin/`, бібліотеки — до `/usr/local/lib/`, а конфігураційні файли — до `/usr/local/etc/`.

Якщо ваша програма та/або інші файли справді мусять знаходитись у якомусь іншому місці, ви все ж можете залишити їх в `/usr/local/` та задати необхідний символічне посилання з необхідного місцезнаходження до `/usr/local/`, наприклад:

```
ln -s /usr/local/bin/foo /usr/bin/foo
```

В будь-якому випадку, якщо ви отримали пакунок, що дозволяє перезбірку, вам потрібно задуматись над створенням Debian-пакунку з нього, та завантаженням до системи Debian. Посібник для розробника-початківця пакунків входить до підручника „Політика Debian“ (див. ‘Яка ще документація існує для системи Debian?’ на стор. 65).

### 3.8 Чому я не можу скомпілювати програми, що вимагають libtermcap?

Debian радше використовує базу даних `terminfo` та бібліотеку процедур термінальних інтерфейсів `ncurses`, аніж базу даних `termcap` та бібліотеку `libtermcap`. Користувачі, що компілюють програми, яким потрібно щось знати про інтерфейси терміналу, повинні замінити звернення до `libtermcap` зверненням до `libtermcurses`.

Для підтримки двійкових файлів, що вже зв’язані з бібліотекою `termcap` і для яких ви не маєте джерельних кодів, Debian пропонує пакунок, що називається `termcap-compat` з файлами `/etc/termcap` та `libtermcap.so.2`. Встановіть цей пакунок, якщо програма виходить з ладу з повідомленням „Не можу завантажити бібліотеку `libtermcap.so.2`“ або повідомляє про відсутній файл `/etc/termcap`.

### 3.9 Чому я не можу встановити AccelX?

AccelX вимагає бібліотеки `termcap` для роботи. Див. ‘Чому я не можу скомпілювати програми, що вимагають `libtermcap`?’ на стор. 14 вище.

### 3.10 Чому мої старі XFree 2.1 Motif-додатки призводять до збоїв у роботі?

Вам потрібно встановити пакунок `motifnls`, котрий містить конфігураційні файли XFree-2.1, що дозволяють програмам Motif, скомпільованим для XFree-2.1 запускатись під XFree-3.1.

---

Без цих файлів деякі Motif-додатки, скомпільовані на інших машинах (як наприклад Netscape), можуть призводити до аварії при спробах копіювання чи вставки у текстове поле та спричиняти ряд інших проблем.





## Розділ 4

# Програмне забезпечення, доступне в Debian.

### 4.1 Які типи додатків та програмного забезпечення для розробників доступні в Debian GNU/Linux?

Як і більшість дистрибутивів, Debian GNU/Linux пропонує:

- основні додатки GNU для розробки програмного забезпечення, маніпуляції файлами та обробки текстів, включаючи gcc, g++, make, texinfo, Emacs, оболонку Bash та значну кількість оновлених Unix-утиліт,
- Perl, Python, Tcl/Tk, та величезну кількість пов'язаних з ними програм, бібліотек і модулів,
- TeX (LaTeX) і Lyx, dvips, Ghostscript,
- систему X Window, що надає мережевий графічний інтерфейс користувача в Linux, та незліченну кількість додатків, включаючи як GNOME та KDE, так і GIMP (GNU Image Manipulation Program) — програму для маніпуляцій з зображеннями.
- повний набір мережевих додатків, включаючи сервери інтернет-протоколів, таких як HTTP (WWW), FTP, NNTP (новини), SMTP та POP (електронна пошта), DNS (сервер доменних назв); реляційні бази даних, такі як PostgreSQL, MySQL; окрім того пропонуються веб-переглядачі, включаючи різноманітні продукти Mozilla,
- повний комплект офісних додатків, включаючи набір OpenOffice.org, Gnumeric та інші електронні таблиці, WYSIWYG-редактори (скорочення від „What You See Is What You Get“ — „що бачиш, те і отримувеш“), календарі.

Понад 15180 пакунків — від серверів та переглядачів новин до підтримки музики, факс-програм, баз даних та електронних таблиць, програм для обробки графіки, спілкування,

мережевих та поштових утиліт, веб-серверів і навіть програм для прийому радіопередач — входять до збірки. Ще 450 комплектів програм доступні у вигляді пакунків Debian, але формально не входять до нього через ліцензійні обмеження.

## 4.2 Хто написав всі ці програми?

Для кожного пакунку автори програм(и) вказані у файлі `/usr/doc/ПАКУНОК/copyright`, де ПАКУНОК — це назва пакунка.

Супроводжуючі, що створили сам пакунок для системи Debian GNU/Linux, перелічені у файлі `control` (перегляньте ‘Що то за файл `control`?’ на стор. 31), який є в кожному пакунку Debian. В журналі змін Debian, що знаходиться в `/usr/share/doc/PACKAGE/changelog.Debian.gz`, також згадуються всі, ті хто працював над пакунками Debian.

## 4.3 Як я можу отримати поточний список програм для Debian?

Повний список доступний на будь-якому з дзеркал Debian (<http://www.debian.org/distrib/ftplist>) в файлі `indices/Maintainers`. В ньому перелічені назви пакунків та імена й електронні адреси відповідних супроводжуючих.

WWW-інтерфейс для Debian-пакунків (<http://packages.debian.org/>) зручно об’єднує пакунки з майже 20 секцій архіву.

## 4.4 Чого немає в Debian GNU/Linux?

Існує список пакунків, котрі потрібно перенести в Debian. Його можна знайти на сторінці Список пакунків, що потребують доопрацювання та перспективних пакунків (<http://www.debian.org/devel/wnpp/>).

Додаткові відомості про те, як додавати відсутні можливості можна знайти в ‘Як я можу стати розробником програмного забезпечення Debian?’ на стор. 71.

## 4.5 Чому при компіляції програм я отримую повідомлення „ld: cannot find -lfoo”? Чому у двійкових пакунках бібліотек немає файлів `libfoo.so`?

Політика Debian вимагає щоб такі символічні посилання (на `libfoo.so.x.y.z` абощо) знаходились у окремих, розробницьких пакунках. Такі пакунки зазвичай називаються `libfoo-dev` чи `libfooX-dev` (відповідні їм пакунки бібліотек називаються `libfooX`, де X — деяке число).

## 4.6 Як Debian підтримує Java і чи підтримує взагалі?

Оскільки офіційні Java Development kit та Runtime Environment виробництва Sun Microsystems не є вільним програмним забезпеченням, воно не може бути офіційно включеним в Debian. Якщо ви, не зважаючи на це, хочете включити його у свою систему, зверніться до пакунку `java-package` в каталозі `contrib`.

Проте, обидва JDK та декілька вільних реалізацій технології Java доступні у вигляді пакунків Debian. Ви можете писати, налагоджувати та запускати Java-програми у Debian.

Для запуску Java-апплетів потрібен веб-браузер що міг би їх розпізнати та виконати. Декілька браузерів, доступних у Debian, як наприклад Konqueror чи Mozilla підтримують спеціальні втулки Java, що роблять можливим всередині них запуск Java-апплетів. Netscape Navigator, хоча він і не є вільним, але також доступний у якості пакунка для Debian і вміє запускати апплети Java.

За додатковою інформацією зверніться до Debian Java FAQ (<http://www.debian.org/doc/manuals/debian-java-faq/>).

## 4.7 Як я можу перевірити, чи я використовую Debian і якої саме версії?

Щоб переконатись, що ваша система справді встановлена з базових дисків Debian, перевірте наявність файлу `/etc/debian_version`, що містить єдиний простий рядок — номер стабільної версії, як визначено у пакунку `base-files`.

Наявність програми `dpkg` свідчить про те, що ви можете встановлювати пакунки Debian у вашій системі, але оскільки ця програма була перенесена у багато інших операційних систем та архітектур, не можна стверджувати, що це є надійний спосіб перевірки того, що у вас саме Debian GNU/Linux.

Користувачі повинні знати, що хоча система Debian складається з багатьох частин, кожна з них можна оновити (майже) незалежно. Кожна стабільна версія Debian має визначений та незмінний вміст. Оновлення доступні окремо. Для короткого опису стану встановлення пакунку `foo` використовуйте команду `dpkg --get-selections foo`. Щоб переглянути версії всіх встановлених пакунків, введіть:

```
dpkg -l
```

Для перегляду більш детального опису запустіть:

```
dpkg --status foo
```

## 4.8 Як справи в Debian з підтримкою інших мов, окрім англійської?

- Debian GNU/Linux поширюється з приблизно двома сотнями клавіатурних карт та інструментами (в пакунку `kbd`) для їх встановлення, перегляду і модифікації.

Під час встановлення системи користувача просять вказати, яку саме клавіатурну розкладку він буде використовувати.

- Переважна більшість програм, котрі ми пакуємо, підтримують ввід не-US-ASCII символів, що використовуються у інших абетках (як, наприклад, ISO-8859-1 чи ISO-8859-2); окрім того, численні програми підтримують багатобайтові мови, як японська чи китайська.
- На даний момент підтримуються сторінки довідки для німецької, іспанської, фінської, французької, угорської, італійської, японської, корейської та польської мов у пакунках manpages-MOBA, де MOBA — дволітерний код країни згідно стандарту ISO. Щоб отримати доступ до такої сторінки, необхідно відповідним чином змінити значення змінної LC\_MESSAGES.

Наприклад, у випадку італомовної довідки змінна LC\_MESSAGES повинна бути встановлена у 'italian'. Програма man шукатиме італійські сторінки довідки у каталозі /usr/share/man/it/.

## 4.9 Як щодо обмежень законодавства США на експорт?

Американське законодавство забороняє експорт оборонних технологій, що включає деякі типи криптографічних програм. Серед інших у цю категорію потрапляють PGP та ssh. У випуску sarge такі пакунки були перенесені до каталогу main (або ж non-free, якщо це було необхідним) у зв'язку з лібералізацією американського законодавства на експорт криптографії.

Для запобігання непотрібного ризику деякі пакунки Debian GNU/Linux були доступні лише з неамериканського сайту [ftp://non-US.debian.org/debian-non-US/](http://non-US.debian.org/debian-non-US/) та його численних дзеркал, кожне з яких знаходилось за межами США (повний список можна знайти за адресою [ftp://non-US.debian.org/debian-non-US/README.non-US](http://non-US.debian.org/debian-non-US/README.non-US)). Ці сайти все ще існують (для підтримки користувачів woody), але їхній вміст більше не підтримується і вважається застарілим. Будь ласка, видаліть будь-які згадки про non-US з вашого конфігураційного файлу /etc/apt/sources.list.

## 4.10 Куди подівся pine?

Через його обмежувальну ліцензію він перенесений до non-free. Більш того, оскільки ліцензія забороняє поширення модифікованих двійкових файлів, ви повинні скопіювати його самостійно з наданих джерельних кодів та латок Debian.

Джерельний пакунок називається pine. Ви можете використовувати пакунок pine-tracker щоб отримати попередження якщо буде потрібно провести оновлення.

Зауважте, що є багато альтернатив як для pine, так і для pico, зокрема mutt і nano, що знаходяться в розділі main.

## 4.11 А де qmail/ezmlm/djbdns?

Ден Бернштейн (Dan J. Bernstein) поширює все написане ним програмне забезпечення (<http://cr.yp.to/software.html>) під обмежувальною ліцензією і, як наслідок, воно знаходиться у розділі non-free. Оскільки ліцензія, котру він використовує, забороняє поширення модифікованих двійкових файлів, ви повинні скомпілювати їх самостійно з джерельних кодів та латок Debian, щоб отримати двійкові пакунки, котрі ви зможете встановити у вашій системі Debian GNU/Linux.

Джерельні пакунки називаються qmail-src, ezmlm-src та djbdns-installer відповідно.

Для qmail вам потрібно спершу встановити qmail-src, а потім запустити build-qmail щоб сформувати пакунок Debian. Вам також потрібно встановити пакунок ucspi-tcp-src щоб отримати ucspi-tcp, від котрого залежить qmail.

Ден Бернштейн (Dan J. Bernstein) також підтримує сторінку FAQ from distributors (<http://cr.yp.to/distributors.html>), якщо вам цікаво почитати про його мотиви (одним з них є міжплатформена сумісність (<http://cr.yp.to/compatibility.html>)).



## Розділ 5

# FTP-архіви Debian

### 5.1 Що означають всі ті теки на ftp-архівах Debian?

Програмне забезпечення у вигляді пакунків для Debian GNU/Linux доступне у одному з декількох дерев тек на кожному дзеркальному сайті Debian.

Тека dists це скорочення від „distributions“ (дистрибутиви, збірки); вона є канонічним шляхом для отримання доступу до існуючих на даний момент версій Debian.

Тека pool містить поточні пакунки, перегляньте ‘Що за тека — pool?’ на стор. [27](#).

Існують такі додаткові теки:

/tools/: DOS-утиліти для створення завантажувальних дисків, поділу вашого диску на розділи, стиснення/розстиснення файлів та завантаження Linux.

/doc/: Основна документація Debian, така як FAQ, інструкції до системи відслідковування помилок та ін.

/indices/: Файл Maintainers та файли override.

/project/: Переважно матеріали, що призначені для розробників, наприклад:

project/experimental/: Ця тека містить пакунки та інструменти, які все ще розробляються і знаходяться в режимі попереднього тестування. Користувачі не повинні використовувати пакунки звідси, тому що це може бути небезпечно і завдати шкоди навіть для системи під керуванням доволі досвідченої людини.

### 5.2 Скільки збірок Debian знаходиться в теці dists?

Є три збірки — стабільна (stable), тестова (testing) та нестабільна (unstable). Тестова збірка іноді „заморожується“ (frozen) (див. ‘Як щодо testing? Чому вона заморожується?’ на стор. [26](#)).

### 5.3 Що означають всі ці імена на кшталт slink, potato та ін.?

Це просто умовні кодові назви. Коли збірка Debian знаходиться в стадії розробки, вона не має номера версії, а лише умовну назву. Вони використовуються щоб полегшити віддзеркалення збірок Debian (якщо справжня тека, наприклад, unstable раптом змінить свою назву на stable, потрібно буде заново переписувати занадто багато файлів).

На даний момент stable є символічним посиланням на sarge (себто Debian GNU/Linux 3.1), а testing — на etch. Тобто sarge є поточною стабільною версією, а etch — тестовою.

unstable є постійним символічним посиланням на sid, оскільки sid — це завжди нестабільна збірка (перегляньте ‘Ну а як щодо „sid“?’ на стор. 25).

#### 5.3.1 Які кодові назви вже використовувались?

Окрім вищезгаданих використовувались наступні кодові назви: buzz для версії 1.1, rex для версії 1.2, bo для версій 1.3.x, hamm для версії 2.0, slink для версії 2.1, potato для версії 2.2 та woody для версії 3.0.

#### 5.3.2 Звідки беруться всі ці імена?

Це все імена персонажів з мультфільму „Toy Story“ компанії Pixar.

- buzz (Buzz Lightyear) був космонавтом,
- rex був тиранозавром,
- bo (Bo Peep) дівчинка, котра опікувалась овечкою,
- hamm був поросям-скарбничкою,
- slink (Slinky Dog®) був іграшковим песиком,
- potato це, звісно, пан Potato®,
- woody був ковбоєм,
- sarge був сержантом Зеленою Пластикової Армії,
- etch це іграшкова класна дошка (Etch-a-Sketch®).
- sid був сусідським хлопчаком, котрий ламав іграшки.



## 5.4 Ну а як щодо „sid“?

sid або unstable — це місце куди початково завантажуються всі пакунки. Вони ніколи не додаються відразу в збірку, тому що пакунки, котрі додаються спочатку повинні бути включені до testing, для того щоб через деякий час бути доданими в наступний випуск stable. sid містить пакунки як для випущених, так і для не випущених архітектур.

Ім'я sid також прийшло з „Toy Story“; так звали сусідського хлопчика, котрий ламав іграшки :o)

1

## 5.5 Що міститься в теці stable?

- stable/main/: Ця тека містить пакунки, що формально складають найсвіжішу версію Debian GNU/Linux.

Всі ці пакунки сформовано згідно з критеріями Debian щодо вільного програмного забезпечення (DFSG) ([http://www.debian.org/social\\_contract#guidelines](http://www.debian.org/social_contract#guidelines)) і вони повністю готові до вживання та поширення.

- stable/non-free/: В цій теці знаходяться пакунки, поширення яких обмежується в спосіб, що вимагає від поширювача дотримуватись специфічних копірайтних умов.

Наприклад, деякі пакунки поширюються під ліцензіями, що забороняють комерційне поширення. Інші дозволяють поширювати, але як умовно-безкоштовне, а не вільне програмне забезпечення. Ліцензії всіх цих пакунків повинні бути детально вивченими та, можливо, узгодженими, перш ніж їх додати у якусь збірку (наприклад, на КД).

- stable/contrib/: Ця тека містить пакунки, що є вільними з точки зору DFSG та можуть вільно поширюватись самі по собі, але якимось чином залежать від пакунків, що не можуть поширюватись вільно і, отже, знаходяться у секції non-free.

## 5.6 Що міститься в теці testing?

Пакунки переносяться в теку testing після того, як пройдуть деякі етапи тестування в unstable.

---

<sup>1</sup>Коли теперішнього sid ще не було, у організації FTP-сайтів був недолік: робилось припущення, що архітектура, внесена до поточної нестабільної версії, буде додана у наступну стабільну. Для багатьох архітектур цього не відбувалось і в момент випуску нової версії відповідні теки переміщувались. Це було непрактичним, бо сильно навантажувало пропуски лінії. Адміністратори архівів працювали над цією проблемою кілька років, розміщуючи двійкові пакунки для не випущених архітектур у спеціальну теку sid. При першому їх випуску робились посилання з поточної стабільної версії у sid і відтоді вони, зазвичай, створювались всередині нестабільної гілки. Таке розміщення призводило до непорозумінь з користувачами. З появою пакункових пулів (див. 'Що за тека — pool?' на стор. 27) двійкові пакунки почали розміщувати в теці pool незалежно від збірки, тож випуск нової версії перестав спричиняти інтенсивне навантаження на канали дзеркал (натомість, воно поступово зростає під час розробки).

Вони мусять узгоджуватись з усіма архітектурами, у які їх будуть переносити та не повинні мати залежностей, що зроблять їх непридатними до встановлення; вони також повинні мати якомога менше блокуючих випуск помилок в тих версіях, що входять у testing. Таким чином ми сподіваємось, що testing завжди достатньо готовий щоб бути кандидатом до випуску.

Додаткову інформацію про статус testing в цілому та кожного пакунка зокрема можна отримати на <http://www.debian.org/devel/testing>.

### 5.6.1 Як щодо testing? Чому вона заморожується?

Як тільки тестова збірка стає достатньо зрілою, керівник випуску „заморожує“ її. Прийняття пакунків до неї сповільнюється та подовжується в часі, щоб переконатись, що з нестабільної збірки в тестову перейшла мінімальна кількість помилок.

Через деякий час тестова збірка стає повністю замороженою. Це означає, що всі нові пакунки, які подаються до тестової версії, не буде прийнято до тих пір, поки вони містять критичні, блокуючі випуск помилки. Тестова збірка може залишатись у такому повністю замороженому режимі під час так званого „тестового циклу“, аж поки випуск не стає неминучим.

Ми ведемо записи про помилки в testing, котрі можуть затримати випуск якогось пакунка, і помилок, що можуть затримати випуск повністю. За деталями зверніться до інформації про поточний тестовий випуск (<http://www.debian.org/releases/testing/>).

Як тільки кількість помилок знижується до прийнятних значень, заморожена тестова збірка оголошується стабільною та випускається з номером версії.

З кожним новим стабільним випуском попередній стає застарілим та поміщається в архів. За додатковою інформацією перегляньте сторінку архіву Debian (<http://www.debian.org/distrib/archive>).

## 5.7 Що знаходиться в теці unstable?

Тека unstable містить знімок поточної розроблюваної системи. Користувачів запрошують використовувати та перевіряти ці пакунки, але попереджають про стан їх готовності. Переваги використання нестабільної версії в тому, що ви завжди знаходитесь на вістрі індустрії програмного забезпечення GNU/Linux, але якщо щось зламається: you get to keep both parts :-)

В теці unstable також знаходяться підтеки main, contrib і non-free; пакунки в них розподіляються за тими ж критеріями, що й в stable.

## 5.8 Що це за теки в dists/stable/main?

Всередині кожного основного дерева тек <sup>2</sup>, є три набори підтек, що містять індексні файли.

<sup>2</sup>dists/stable/main, dists/stable/contrib, dists/stable/non-free, dists/unstable/main/, тощо;

Один набір — це теки `binary`-дещо, що містять індексні файли для двійкових пакунків кожної доступної архітектури, наприклад `binary-i386` для пакунків, що будуть виконуватись на машинах Intel x86, чи `binary-sparc` для пакунків, що будуть виконуватись на робочих станціях Sun SPARCStations.

Повний список доступних архітектур для кожного випуску доступний на присвяченій випуску веб-сторінці (<http://www.debian.org/releases/>). Для поточного випуску, будь ласка, перегляньте ‘На яких апаратних архітектурах та системах запускається Debian GNU/Linux?’ на стор. 11.

Індексні файли у `binary-*` називаються `Packages(.gz)` і містять резюме для кожного двійкового пакунку, що включений у збірку. Поточні двійкові пакунки (для `woody` та наступних випусків) знаходяться на верхньому рівні теки `pool`.

Крім того існує підтека `source/`, котра містить індексні файли для джерельних пакунків, включених у збірку. Індексний файл називається `Sources(.gz)`.

І останнє, але не менш важливе — є набір підтек, призначених для файлів системи встановлення. У випуску `woody` вони називаються `disks`-архітектура; в `sarge` — `debian-installer/binary`-архітектура.

## 5.9 А де джерельні коди?

Джерельний код є для будь-чого з системи Debian. Більш того, ліцензійні положення більшості програм у системі вимагають, щоб разом з програмою розповсюджувався джерельний код, або ж пропонують це робити.

Джерельні коди знаходяться в теці `pool` (див. ‘Що за тека — `pool`?’ на стор. 27) разом з усіма архітектурно-специфічними теками для двійкових пакунків. Щоб отримати джерельний код без ознайомлення з структурою FTP-архіву, спробуйте команду на кшталт `apt-get source назва_пакунка`.

Деякі пакунки поширюються винятково у джерельних кодах через їх ліцензійні обмеження. В першу чергу це стосується `pine`, перегляньте ‘Куди подівся `pine`?’ на стор. 20 щоб отримати більше інформації.

Джерельні коди можуть і не бути доступними для пакунків, що розташовані в теках `contrib` та `non-free`, що формально не є частиною системи Debian.

## 5.10 Що за тека — `pool`?

Пакунки зберігаються у величезному сховищі (`pool`), структурованому за назвами джерельних пакунків. Щоб зробити це діло керованим, сховище поділяється по секціям (`main`, `contrib` та `non-free`) та по перших літерах назв джерельних пакунків. Ці теки містять декілька файлів: двійкові пакунки для кожної архітектури та джерельні пакунки, з яких формуються двійкові.

Ви можете дізнатись місцезнаходження кожного пакунка виконавши команду `apt-cache showsrc назва_пакунка` та подивившись на рядок „Directory:“. Наприклад, пакунки `apache` знаходяться в `pool/main/a/apache/`.

Крім цього, оскільки пакунків `lib*` дуже багато, вони розглядаються особливим чином: наприклад, пакунки `libpaper` знаходяться в теці `pool/main/libp/libpaper/`.

3

## 5.11 Що таке incoming?

Після того, як супроводжувач завантажує пакунок на сервер сховища, він деякий час знаходиться в теці „incoming“, доки не буде перевірено, що він є справжнім і додано його до архіву.

Як правило, ніхто не повинен встановлювати пакунки звідти, проте, на випадок деяких рідкісних, надзвичайних ситуацій тека `incoming` доступна за адресою <http://incoming.debian.org/>. Ви можете вручну викачувати пакунки і, перевіривши PGP-підписи та md5-суми в файлах `.changes` і `dsc`, встановлювати їх.

## 5.12 Як мені створити своє власне apt-сумісне сховище?

Якщо ви хочете збудувати деякі власні пакунки Debian, які б ви хотіли встановлювати за допомогою стандартних інструментів управління пакунками Debian, ви можете створити ваш власний apt-сумісний архів пакунків. Це також корисно, якщо ви не хотіли б оприлюднювати ваші пакунки, доки їх не буде включено до проекту Debian. Вказівки щодо того, як це можна зробити ви знайдете в Debian Repository HOWTO (<http://www.debian.org/doc/manuals/repository-howto/repository-howto>).

---

<sup>3</sup>Історично склалося так, що пакунки зберігались у підтеці `dists` збірки, до котрої вони належали. Це спричиняло деякі проблеми, як наприклад велике завантаження пропускних каналів дзеркал при виході нових версій. Ця проблема була вирішена введенням єдиного сховища пакунків. Теки `dists` досі використовуються для зберігання індексних файлів, котрі в свою чергу використовуються програмами на кшталт `apt`. Ви також можете зустріти шляхи `dists/potato` чи `dists/woody` у полі заголовку `Filename` деяких старих пакунків.

## Розділ 6

# Основи системи управління пакунками Debian

### 6.1 Що таке пакунок Debian?

Пакунок зазвичай містить всі файли, що потрібні для реалізації певних команд чи можливостей. Є два типи пакунків Debian:

- Двійкові пакунки, що містять виконувані і конфігураційні файли, сторінки довідки `info/man`, інформацію про авторські права та іншу документацію. Ці пакунки поширюються у специфічному для Debian архівному форматі (див. ‘Який формат двійкових пакунків Debian?’ на стор. 30); як правило вони вирізняються закінченням `„.deb“`. Двійкові пакунки можуть бути розпаковані за допомогою спеціального інструменту — `dpkg`; деталі описано в його довідці.
- Джерельні пакунки, що містять файл `.dsc`, котрий описує пакунок (включаючи назви наступних файлів), файл `.orig.tar.gz`, що містить оригінальний, незмінений джерельний код у форматі `tar`, стиснутому `gzip`’ом, та зазвичай `.diff.gz` файл, що містить специфічні для Debian зміни джерельного коду. Утиліта `dpkg-source` запаковує та розпаковує джерельні архіви Debian; деталі описано в її довідці.

Встановлення програмного забезпечення за допомогою системи керування пакунками використовує „залежності“, які старанно формуються супроводжуваними пакунками. Ці залежності описані у файлі `control`, що знаходиться в кожному пакунку. Наприклад, пакунок, що містить компілятор GNU C (`gcc`) „залежить“ від пакунка `binutils`, котрий містить зв’язувач (`linker`) та асемблер. Якщо користувач спробує встановити `gcc` не маючи при цьому встановленого `binutils`, система керування пакунками (`dpkg`) надрукує повідомлення про помилку та перерве встановлення `gcc` (однак, наполегливий користувач може скорегувати таку поведінку; див. `dpkg(8)`). Щоб дізнатись більше, перегляньте ‘Що мається на увазі, коли кажуть, що пакунок залежить, рекомендує, пропонує, конфліктує, замінює або забезпечує (Depends, Recommends, Suggests, Conflicts, Replaces або Provides) інший пакунок?’ на стор. 34 нижче.

Пакувальні інструменти Debian можуть вживатись для:

- маніпуляцій та керування пакунками чи частинами пакунків,
- допомоги користувачу у разі необхідності розділення пакунків, наприклад для копіювання на дискети,
- допомоги розробникам у формуванні архівів пакунків, та
- допомоги користувачам у встановленні пакунків, що розташовані на віддаленому ftp-сайті.

## 6.2 Який формат двійкових пакунків Debian?

Пакунок, або ж архівний файл Debian містить виконувані файли, бібліотеки та документацію, що відносяться до певного набору відповідних програм. Зазвичай назва такого файлу закінчується на `.deb`.

Нутроці формату двійкових пакунків Debian описані у сторінці довідки `deb(5)`. Внутрішній формат є предметом змін (між основними версіями Debian GNU/Linux), тому, будь ласка, завжди використовуйте `dpkg-deb(1)` для маніпуляцій з `.deb`-файлами.

## 6.3 Чому назви пакунків Debian такі довгі?

Назва двійкового Debian-пакунку формується за наступною домовленістю: `<Дещо>_<НомерВерсії>-<НомерРедакціїBDebian>.deb`

Мається на увазі, що дещо насправді є назвою пакунка. Для перевірки ви можете дізнатись назву пакунка, що пов'язаний файлом архіву Debian одним з двох шляхів:

- перевірити файл `Packages` в теці, в котрій він розміщувався на ftp-сайті Debian. Цей файл містить опис кожного пакунку; в тому числі й формальна назва пакунка.
- за допомогою команди `dpkg --info foo_VVV-RRR.deb` (де VVV та RRR — версія та редакція пакунка відповідно). Вона відображає, поміж інших речей, ще й відповідну назву пакунка для даного архіву.

Компонента VVV — це номер версії, що визначається основним розробником. Він не підкоряється жодним стандартам і може мати різні формати, як „19990513“ так і „1.3.8pre1“.

Компонента RRR — це номер редакції пакунка в Debian, він визначається розробником (чи окремим користувачем, якщо він захоче сформувати пакунок самостійно). Цей номер вказує на кількість редагувань пакунка; таким чином новий рівень редакції як правило означає зміни у `Makefile` (`debian/rules`), контролюючому файлі (`debian/control`), встановлювальному чи видаляючому сценаріях (`debian/p*`), або ж у конфігураційних файлах, що використовуються пакунком.

## 6.4 Що то за файл control?

Детальний опис вмісту файлу control можна знайти в 5-му розділі Підручника політики Debian, див. ‘Яка ще документація існує для системи Debian?’ на стор. 65.

Тут ми коротко розглянемо приклад файлу control пакунка hello:

```
Package: hello
Priority: optional
Section: devel
Installed-Size: 45
Maintainer: Adam Heath <doogie@debian.org>
Architecture: i386
Version: 1.3-16
Depends: libc6 (>= 2.1)
Description: The classic greeting, and a good example
The GNU hello program produces a familiar, friendly greeting. It
allows nonprogrammers to use a classic computer science tool which
would otherwise be unavailable to them.
.
Seriously, though: this is an example of how to do a Debian package.
It is the Debian version of the GNU Project's 'hello world' program
(which is itself an example for the GNU Project).
```

У полі Package вказується назва пакунка. Саме за ним паунком можна маніпулювати за допомогою спеціальних інструментів, і вона як правило (але не обов’язково) є такою ж, як і перша частина назви архівного файлу Debian.

У полі Version вказується як версія виробника програми, так і рівень редакції пакунка Debian, як це описано в ‘Чому назви паунків Debian такі довгі?’ на стор. 30.

Поле Architecture вказує, для якої системи скомпільований двійковий файл.

В полі Depends перелічуються пакунки, котрі повинні бути встановлені для успішного встановлення даного пакунку.

Поле Installed-Size вказує, скільки дискового простору займе встановлений паунок. Передбачається, що його будуть використовувати інші програми для того, щоб визначити, чи достатньо на розділі місця для встановлення пакунка.

Рядок Section вказує на розділ, в котрому міститься паунок на ftp-сайтах Debian. Це — назва підтеки (в одній з основних тек — див. ‘Що означають всі ті теки на ftp-архівах Debian?’ на стор. 23), де розташовано паунок.

Priority вказує на те, наскільки даний паунок важливий при встановленні, то ж напівінтелектуальні програми на кшталт dselect чи console-apt можуть поділити пакунки на категорії, наприклад за ступенем необхідності їх встановлення. Див. ‘Що це за пакунки — Essential, Required, Important, Standard, Optional та Extra (Невіддільний, Необхідний, Важливий, Типовий, Необов’язковий та Додатковий)?’ на стор. 33.

В полі Maintainer можна знайти електронну адресу людини, котра що на даний момент відповідає за супровід пакунка.

В полі Description знаходиться короткий перелік основних можливостей пакунка.

За додатковою інформацією про всі можливі поля, що може мати пакунок, зверніться до розділу 5 Підручника політики Debian, „Контрольні файли та їх поля“.

## 6.5 Для чого призначені conffile?

У цьому файлі перелічуються конфігураційні файли (розміщених, як правило, в теці /etc), котрі система керування пакунками не буде перезаписувати при оновленні пакунка. Таким чином досягається захист вмісту конфігураційних файлів, що є критичним на працюючій системі.

Щоб дізнатись точно, які саме файли не будуть затерті при оновленні пакунка, запустіть

```
dpkg --status package
```

та перегляньте поле Conffiles.

## 6.6 Що роблять сценарії preinst, postinst, prerm, та postrm?

Ці файли — це виконувані сценарії, що автоматично запускаються перед чи після встановлення пакунку. Разом з файлом control всі ці файли є частиною „контролюючої“ секції файлу архіву Debian.

Кожен окремо вони роблять наступне:

**preinst** Цей сценарій запускається перед розпаковуванням пакунку з його Debian-архіву („deb“-файлу). Багато preinst-сценаріїв зупиняють служби для пакунків, що оновлюються, аж доки їх встановлення чи оновлення не завершаться (з наступним успішним виконанням сценарію postinst).

**postinst** Цей сценарій, як правило, завершує необхідні конфігурації пакунка foo коли його вже було розпаковано з архівного „deb“-файлу. Часто postinst просить ввести користувача якусь інформацію та/або попереджає його у випадку використання стандартних значень, щоб він не забув повернутись та переконфігурувати пакунок у разі необхідності. Потім багато сценаріїв виконують команди, необхідні для запуску чи перезапуску різноманітних служб, оскільки новий пакунок вже встановлено чи оновлено.

**prerm** Цей сценарій як правило зупиняє всі демони, що мають певне відношення до пакунку. Він виконується перед видаленням файлів, пов'язаних з пакунком.



postrm Цей сценарій зазвичай змінює відсилачі та інші файли, пов'язані з foo та/або видаляє файли, створені пакунком (див. також 'Що таке віртуальний пакунок?' на стор. 34).

На даний момент всі керуючі файли можна знайти в теці `/var/lib/dpkg/info`. Файли, що відносяться до пакунка foo починаються з foo і мають розширення відповідно `preinst`, `postinst` і т.д. Файл `foo.list` з вищевказаної теки — це список всіх файлів, встановлених з пакунка foo. (Зауважте, що місцезнаходження цих файлів визначається внутрішніми налаштуваннями dpkg; ви не повинні орієнтуватись саме на цей шлях.)

## 6.7 Що це за пакунки — Essential, Required, Important, Standard, Optional та Extra (Невіддільний, Необхідний, Важливий, Типовий, Необов'язковий та Додатковий)?

Кожному пакункові Debian супроводжуваними збірки присвоюється пріоритет, щоб допомогти системі керування пакунками. Пріоритети бувають:

- Required (Необхідний): пакунки, що необхідні для правильного функціонування системи.

Сюди входять всі інструменти, необхідні для усунення дефектів системи. Ви не повинні видаляти ці пакунки, інакше ваша система буде повністю виведена з ладу і швидше за все ви навіть не зможете використати dpkg щоб встановити їх назад. В системах з набором лише необхідних пакунків, швидше за все, не можна працювати, але вони надають системному адміністратору достатньо функціональності, щоб завантажити систему та встановити додаткове програмне забезпечення.

- Important (Важливі) пакунки повинні бути встановлені на будь-якій Unix-системі. До таких відносяться всі інші пакунки, без яких використання системи буде незручним або маломожливим. Сюди НЕ входять Emacs, X11, TeX чи інші великі додатки; ці пакунки лише формують кістяк.

- Standard (Типові) пакунки є стандартом для будь-якої Linux-системи, включаючи але не обмежуючись текстовими системами.

Сюди входить все те, що встановлюється по замовчуванню, якщо користувачі не вибрали що-небудь ще. Це не стосується багатьох великих додатків, але все ж до таких пакунків відносяться деякі програми для розробників на кшталт компіляторів GNU C та C++ (gcc, g++), GNU make, а також інтерпретатор мови Python, деяке серверне програмне забезпечення типу OpenSSH, демон друку BSD (lpr) і дзеркало портів (portmapper) RPC (portmap).

- Optional (Необов'язкові) пакунки включають в себе все те, що вам є сенс встановити, якщо ви не знаєте, що воно таке, або ж не має якихось особливих забаганок.

Сюди входять X11, повна збірка TeX та багато інших програм.

- Extra (Додаткові): пакунки, котрі або конфліктують з іншими, з вищим пріоритетом, або мають сенс до використання, лише якщо ви точно знаєте, що воно таке, або ж мають якісь спеціалізовані вимоги, що не дозволили включити їх до Optional.

Якщо ви встановлюєте стандартний Debian, всі пакунки з пріоритетом Standard та вищим буде встановлено до вашої системи. Якщо ви оберете перед-визначення задач встановлення ви отримаєте доступ і до пакунків з нижчим пріоритетом також.

На додаток, деякі пакунки позначено як Essential (Невіддільні), оскільки вони абсолютно необхідні для успішного функціонування системи. Інструменти системи керування пакунами не зможуть їх видалити.

## 6.8 Що таке віртуальний пакунок?

Віртуальний пакунок — це загальна назва, що дається пакункові чи групі пакунків, кожен з яких пропонує схожу базову функціональність. Наприклад, обидві програми `tin` та `trn` призначені для читання новин, а отже повинні задовольнити залежність іншої програми, що потребує встановленої програми для читання новин у системі. Тому обидві вони забезпечують „віртуальний пакунок“, що називається `news-reader`.

Аналогічно, програми `smail` та `sendmail` забезпечують функціональність транспортного агента електронної пошти. Тому кажуть, що вони обидві забезпечують віртуальний пакунок „mail transport agent“. Якщо хоча б одна з них встановлена, будь-яка інша програма, що залежить від `mail-transport-agent`, буде задоволена наявністю віртуального пакунка.

Debian забезпечує цей механізм таким чином, що якщо в системі встановлено більш ніж один пакунок з однаковою базовою функціональністю, то системний адміністратор може зробити один із них привілейованим. Для ознайомлення з описом можливостей відповідної команди, `update-alternatives`, перегляньте ‘Деяким користувачам подобається `mawk`, іншим `gawk`; одні користуються `vim`, інші `elvis`; дехто уподобав `trn`, а дехто `tin`; як Debian підтримує такі різноманітності?’ на стор. 63.

## 6.9 Що мається на увазі, коли кажуть, що пакунок залежить, рекомендує, пропонує, конфліктує, замінює або забезпечує (Depends, Recommends, Suggests, Conflicts, Replaces або Provides) інший пакунок?

Система керуваннями пакунків Debian оперує шкалою залежностей, щоб вказати, в який спосіб програма А може реагувати на наявність в системі програми В:

- Пакунок А залежить (`depends`) від пакунку В, якщо В безумовно повинен бути встановленим у системі щоб запустити програму А. В деяких випадках А залежить не просто від В, але від певної версії В. В такому випадку залежність як правило має

нижню межу, в цьому випадку А залежить від будь-якої версії В, що новіша за деяку визначену версію.

- Паунок А рекомендує (recommends) паунок В, якщо супроводжуючий паунок А вважає, що переважній більшості користувачів не потрібен паунок А без функціональності, що її забезпечує паунок В.
- Паунок А пропонує (suggests) паунок В, якщо В містить файли, пов'язані з функціональністю А (та, зазвичай, покращують її).
- Паунок А конфліктує (conflicts) з паунком В, якщо А не зможе функціонувати при встановленому у системі В. Значно рідше конфлікт виникає через те, що А містить файли, відмінні від аналогічних у паунку В. Часто conflicts поєднується з replaces.
- Паунок А замінює (replaces) паунок В, коли файли, встановлені В видаляються та (в деяких випадках) перезаписуються файлами з А.
- Паунок А забезпечує (provides) паунок В, коли всі файли та функціональність В включені в А. Цей механізм дає можливість користувачам з обмеженим дисковим простором встановити лише ту частину паунку А, котра їм справді необхідна.

Детальніша інформація по кожному з цих термінів знаходиться у підручнику Політики Debian.

## 6.10 Що означає „попередня залежність“?

„Попередня залежність“ — це особлива залежність. Для більшості паунків dpkg розпакує архівний (.deb) файл незалежно від того, чи є в системі файли, від котрих він залежить. Це означає, що dpkg добуде їх з архіву та помістить туди, де вони повинні знаходитись. Якщо ці паунки залежать від наявності інших паунків у системі, dpkg відмовиться завершити встановлення (виконуючи дію „configure“), поки інші паунки не будуть встановлені.

Проте для деяких паунків dpkg відмовиться проводити розпакування, поки всі залежності не будуть задоволені. Цей механізм називається попередньою залежністю від присутності у системі інших паунків. Проект Debian пропонує цей механізм для безпечного оновлення системи від формату a.out до ELF, де порядок, у якому розпаковуються паунки, є критичним. Є й інші випадки значних оновлень, у яких цей механізм є доволі зручним, наприклад паунки з пріоритетом „необхідний“ та їхня залежність від libc.

Як завжди, більш детальну інформацію про це можна знайти в Підручнику політики Debian.

### 6.11 Що мається на увазі, кажучи невідомий, встановлений, видалений, очищений чи зафіксований (unknown, install, remove, purge та hold) про стан пакунку.

Ці прапорці вказують, що користувач хоче зробити з пакунком (як вказано або діями користувача в секції Select програми dselect, або ж прямими викликами dpkg).

Вони означають наступне:

- unknown (невідомий) — користувач ніколи не цікавився цим пакунком.
- install (встановлений) — користувач бажає встановити чи оновити пакунок.
- remove (видалений) — користувач хоче видалити пакунок, але не хотів видаляти існуючі файли конфігурації.
- purge (очищений) — користувач хотів би видалити пакунок разом з усіма конфігураційними файлами.
- hold (зафіксований) — користувач не бажає обробляти цей пакунок, тобто він хоче зберегти його поточну версію з поточним статусом.

### 6.12 Як мені зафіксувати пакунок?

Є три способи зафіксувати пакунок — за допомогою dpkg, aptitude, або ж dselect.

Для dpkg вам потрібно експортувати список станів пакунків за допомогою команди

```
dpkg --get-selections \* > selections.txt
```

Далі відредагуйте отриманий файл selections.txt, змінивши рядки, що містять назви пакунків, які ви хочете зафіксувати, як наприклад libc6 з ось таких

```
libc6                                install
```

на такі:

```
libc6                                hold
```

Збережіть файл та завантажте його назад в базу даних dpkg таким чином:

```
dpkg --set-selections < selections.txt
```

У випадку aptitude ви можете зафіксувати пакунок такою командою:

```
aptitude hold назва_пакунка
```

та зняти фіксацію командою

```
aptitude unhold назва_пакунка
```

Для `dselect` вам потрібно лише перейти до екрану `[S]elect`, знайти пакунок, котрий ви хочете зафіксувати, та натиснути клавішу '=' або 'H'. Зміни збережуться, як тільки ви покинете екран `[S]elect`.

### 6.13 Як мені встановити джерельний пакунок?

Джерельні пакунки Debian насправді не можуть бути встановлені, вони лише розпаковуються в ту теку, де ви хочете зібрати двійкові пакунки, котрі вони створюють.

Джерельні пакунки розповсюджуються на більшості тих дзеркал, котрі пропонують двійкові пакунки. Якщо ви налаштуєте `sources.list(5)` вашого АРТ таким чином, щоб він включав необхідні рядки „deb-src“, ви зможете легко встановлювати будь-які джерельні пакунки командою

```
apt-get source назва_пакунка
```

Щоб дійсно допомогти вам компонувати джерельні пакунки, Debian пропонує так званий механізм компонування залежностей. Мається на увазі, що супроводжуючий джерельного пакунку веде список інших пакунків, що потрібні для компонування даного. Щоб побачити, як це зручно, запустіть

```
apt-get build-dep назва_пакунка
```

перед збиранням джерельних кодів.

### 6.14 Як мені скомпонувати двійковий пакунок з джерельного?

Вам потрібні всі файли `foo_*.dsc`, `foo_*.tar.gz` та `foo_*.diff.gz` щоб скомпілювати джерельні коди (зауважте, що для деяких пакунків, що є рідними для Debian, немає файлів `diff.gz`).

Після того як ви їх отримали (див. ‘Як мені встановити джерельний пакунок?’ на стор. 37), якщо у вас встановлено пакунок `dpkg-dev`, команда

```
dpkg-source -x foo_version-revision.dsc
```

видобуде пакунок в теку під назвою `foo-version`.

Якщо ви хочете лише скопіювати пакунок, ви можете перейти в цю теку та запустити команду

```
dpkg-buildpackage -rfakeroot -b
```

щоб скомпонувати пакунок (зауважте, що при цьому у вас повинен бути встановлений пакунок `fakeroot`), а потім

```
dpkg -i ../foo_version-revision_arch.deb
```

щоб встановити новозбудований пакунок.

## 6.15 А як мені створити власний пакунок Debian?

Для детального опису прочитайте Довідник нового супроводжуючого, доступний у пакунку `maint-guide`, або за адресою <http://www.debian.org/doc/devel-manuals#maint-guide>.

## Розділ 7

# Інструменти для керування пакунками Debian

### 7.1 Які програми пропонує Debian для керування пакунками?

Для керування пакунками Debian існують численні утиліти, від графічних та псевдографічних інтерфейсів до низькорівневих інструментів, що використовуються для встановлення пакунків. Всі доступні утиліти базуються на низькорівневих інструментах та перераховуються тут по мірі зростання їхньої складності.

Важливо зрозуміти, що інструменти вищого рівня, як `aptitude` чи `dselect` залежать від `apt`, котра, у свою чергу, залежить від `dpkg`.

Перегляньте АРТ HOWTO (<http://www.debian.org/doc/manuals/apt-howto/>), якщо бажаєте дізнатись більше про інструменти керування пакунками Debian. Цей документ є доступним різними мовами та в різних форматах, зверніться до запису про АРТ HOWTO на сторінці проекту документації Debian Огляд підручників для користувачів (<http://www.debian.org/doc/user-manuals#apt-howto>).

#### 7.1.1 `dpkg`

Це основна програма керування пакунками. `dpkg` можна викликати з великою кількістю опцій. Найчастіше вживаються такі:

- Отримати список опцій: `dpkg --help`.
- Надрукувати файл `control` (та іншу інформацію) для заданого пакунку: `dpkg --info foo_VVV-RRR.deb`
- Встановити пакунок (включаючи розпакування та конфігурування) у файлову систему на жорсткому диску: `dpkg --install foo_VVV-RRR.deb`.

- Розпакувати (але не конфігурувати) архівний файл Debian у файлову систему на жорсткому диску: `dpkg --unpack foo_VVV-RRR.deb`. Зауважте, що ця команда не обов’язково робить пакунок готовим до вжитку; деякі файли можуть потребувати наступного налаштування для правильного запуску. Ця команда видаляє вже встановлену версію програми та запускає зв’язаний з паунком сценарій `preinst` (див. ‘Що роблять сценарії `preinst`, `postinst`, `prerm`, та `postrm`?’ на стор. 32).
- Конфігурувати пакунок, що був попередньо розпакований: `dpkg --configure foo`. Окрім іншого, ця команда запускає сценарій `postinst` (див. ‘Що роблять сценарії `preinst`, `postinst`, `prerm`, та `postrm`?’ на стор. 32). Вона також оновлює файли, перелічені у `conffiles`. Зауважте, що операція `configure` приймає в якості аргументу назву пакунку, а не назву архівного файлу Debian (для прикладу, правильно буде вказувати `foo`, а не `foo_VVV-RRR.deb`).
- Видобути файл “`blurf`” (або групу файлів “`blurf*`”) з архівного файлу Debian: `dpkg --fsys-tarfile foo_VVV-RRR.deb | tar -xf - blurf*`.
- Видалити пакунок (але не його конфігураційні файли): `dpkg --remove foo`.
- Видалити пакунок (включно з конфігураційними файлами): `dpkg --purge foo`.
- Перелічити стан встановлення пакунків, що містять послідовність символів (чи регулярний вираз) `foo*`: `dpkg --get-selections | grep foo*`.

### 7.1.2 APT

APT — це вдосконалений пакунковий інструмент (Advanced Package Tool), що постачає програму `apt-get`. `apt-get` забезпечує простий шлях для отримання та встановлення пакунків з декількох джерел за допомогою командного рядка. На відміну від `dpkg`, `apt-get` працює не з `.deb`-файлами, а з назвами пакунків і може встановлювати їх лише з місць, котрі перелічені у файлі `/etc/apt/sources.list`. `apt-get` викликає `dpkg` відразу після завантаження архівного `.deb`-файлу з сконфігурованих джерел.<sup>1</sup>

Ось деякі стандартні методи використання `apt-get`:

- Щоб оновити список пакунків, відомих системі, запустіть:

```
apt-get update
```

(вам потрібно запускати цю команду регулярно для оновлення списку пакунків)

- Щоб оновити всі пакунки у вашій системі, наберіть:

```
apt-get upgrade
```

---

<sup>1</sup>Зверніть увагу, що існують портування, котрі дозволяють використовувати цю утиліту з іншими системами управління пакунками, наприклад, менеджером пакунків Red Hat, відомого як `rpm`



- Щоб встановити пакунок `foo` разом з усіма залежностями, виконайте:

```
apt-get install foo
```

- Щоб видалити пакунок `foo` з вашої системи, введіть:

```
apt-get remove foo
```

- Щоб видалити пакунок `foo` разом з його конфігураційними файлами, наберіть:

```
apt-get --purge remove foo
```

- Щоб оновити всі пакунки у вашій системі до нової версії Debian GNU/Linux, запустіть:

```
apt-get dist-upgrade
```

Зверніть увагу, що ви повинні бути зареєстровані як `root` щоб мати можливість виконувати будь-які команди, котрі змінюють системні пакунки.

Набір інструментів `apt` включає також утиліту `apt-cache` для запитів списків пакунків. Ви можете використовувати її щоб знайти пакунки, котрі відповідають певним вимогам за допомогою простих текстових запитів чи регулярних виразів, або ж запитів щодо залежностей у системі керування пакунками. Ось деякі типові способи використання `apt-cache`:

- Знайти пакунки, чий опис містить слово:

```
apt-cache search слово
```

- Роздрукувати детальну інформацію про пакунок:

```
apt-cache show назва_пакунка
```

- Вивести список пакунків, від котрих залежить даний пакунок:

```
apt-cache depends назва_пакунка
```

- Показати детальну інформацію про доступні версії даного пакунку та пакунків, від котрих він залежить:

```
apt-cache showpkg назва_пакунка
```

Щоб отримати детальнішу інформацію, встановіть пакунок `apt` та прочитайте `apt-get(8)`, `sources.list(5)`, встановіть пакунок `apt-doc` та прочитайте `/usr/share/doc/apt-doc/guide.html/index.html`.

### 7.1.3 aptitude

aptitude — це керівник пакунків для системи Debian GNU/Linux, що являє собою зовнішній інтерфейс до інфраструктури керування пакунками apt. aptitude — це програма з псевдографічним інтерфейсом, що базується на бібліотеці curses, вона може використовуватись для швидкого й зручного керування пакунками.

aptitude забезпечує функціональність dselect і apt-get та багато додаткових можливостей, котрих немає у жодній іншій програмі:

- aptitude забезпечує доступ до всіх версії пакунку.
- aptitude веде журнал своїх дій у файлі `/var/log/aptitude`.
- aptitude полегшує відслідковування застарілого програмного забезпечення ведучи список „Застарілих та локальних пакунків“ (Obsolete and Locally Created Packages).
- aptitude є достатньо потужною системою для пошуку окремих пакунків та виводу короткої інформації про них. Користувачі, котрі знайомі з mutt, швидко засвоять і її, оскільки синтаксис виразів взято саме звідти.
- aptitude відслідковує, які саме пакунки було встановлено завдяки залежностям та видаляє їх, якщо пакунки, які залежали від них, були видалені з системи.
- aptitude може автоматично встановлювати рекомендовані (Recommended:) пакунки<sup>2</sup>.
- В повноекранному режимі aptitude має вбудовану функціональність суперкористувача та може запускатись звичайним користувачем. Вона викличе програму su (та запитає пароль суперкористувача, звісно) якщо вам справді знадобляться привілеї системного адміністратора.

Ви можете працювати з aptitude як через візуальний інтерфейс (просто запустивши aptitude) так і напряду з командного рядка. Синтаксис командного рядка дуже подібний до синтаксису apt-get. Наприклад, щоб встановити пакунок foo, вам потрібно запустити aptitude install foo.

Зверніть увагу, що для встановлення та/або оновлення вашої системи Debian рекомендує саме aptitude.

Для додаткової інформації прочитайте сторінку підручника aptitude(8) та встановіть пакунок aptitude-doc-en.

### 7.1.4 dselect

Ця програма являє собою керований у режимі меню інтерфейс до системи керування пакунками Debian. Вона є особливо зручною для першого встановлення. Деякі користувачі

---

<sup>2</sup>Не зважаючи на те, що це часто призводить до встановлення в систему більшої кількості пакунків, ніж насправді необхідно для її роботи.

можуть почувати себе зручніше у `aptitude`, котра також рекомендується для масштабних оновлень. Щоб отримати додаткову інформацію про `aptitude` див. ‘`aptitude`’ на стор. 42.

`dselect` може:

- бути провідником для користувача, котрий вирішив встановити чи видалити котрийсь із пакунків; перевірити, чи не було встановлено пакунків, що конфліктують з іншими та чи встановлено всі пакунки, котрі необхідні для роботи даного пакунка;
- попередити користувача про суперечливість чи несумісність його вибору;
- визначити порядок встановлення пакунків;
- автоматично виконати встановлення чи видалення;
- провести користувача через будь-який конфігураційний процес, необхідний для встановлення пакунку.

`dselect` розпочинає свою роботу з представлення користувачеві меню з 7 елементів, кожен з котрих є наперед визначеною дією. Користувач може обрати будь-яку з них за допомогою стрілок, пересуваючи підсвітлений рядок та натиснувши `<Enter>` щоб вибрати підсвітлену дію.

Що користувач побачить на наступному екрані залежить від дії, котру він обрав. Якщо він обрав будь-яку з опцій окрім `Access` чи `Select`, `dselect` просто перейде до виконання заданої дії: наприклад, якщо користувач вибрав `Remove`, `dselect` спробує видалити всі файли, що вибрані для видалення користувачем при попередньому відвідуванні екрану `Select`.

Елементи меню `Access` та `Select` виводять додаткові підменю. В обидвох випадках меню представлені у вигляді розділеного екрану: у верхній частині знаходиться прокручуваний список можливостей вибору, а в нижній — коротке пояснення для кожного з елементів верхньої частини.

Додаткова довідка доступна після натиснення в будь-який момент клавіші `??`.

Порядок, в котрому представлені пункти початкового меню `dselect` аналогічний порядку дій користувача коли він хоче встановити нові пакунки. Проте користувач може вибирати будь-який елемент меню стільки разів, скільки вважатиме за потрібне (в залежності від того, що він хоче зробити).

- Розпочніть з вибору методу доступу (`Access Method`). Тут вибирається спосіб, у котрий користувач планує отримувати пакунки Debian; наприклад, деякі користувачі мають пакунки на компакт-диску, тоді як інші планують отримувати їх з FTP. Вибір методу доступу зберігається по завершенню роботи з `dselect`, отож вам не потрібно буде вибирати його знову.
- Далі оновіть (`Update`) список доступних пакунків. Щоб зробити це, `dselect` зчитує файл `„Packages.gz“`, котрий повинен знаходитись в теці, де зберігаються пакунки Debian (Але якщо його там не виявиться, `dselect` зможе створити його для вас).

- Виберіть (Select), які пакунки ви хочете встановити до системи. Після вибору цього пункту меню користувачу спочатку виводиться повноекранна довідка (при умові, що програма запущена без опції `-expert`). Коли користувач покине вікно довідки, він побачить розділений надвоє екран, де він зможе обрати пакунки для встановлення (чи видалення).

Верхня частина екрану — це порівняно вузьке вікно зі списком 15400 пакунків Debian; нижня містить опис вибраного (підсвітленого) пакунку чи групи пакунків.

Ви можете вибрати пакунки, з котрими працюватиме програма, підсвітлюючи назву пакунку або мітку групи пакунків. Після цього ви можете вибирати пакунки:

для встановлення: Це досягається натисненням клавіші `+`.

для видалення: Пакунки можна видаляти двома способами:

- видаляючи (remove): видаляються більшість файлів, пов'язаних з пакунком, але залишаються файли, відмічені як конфігураційні (див. ‘Для чого призначені `conf`file?’ на стор. 32) та конфігураційна інформація про пакунок. Це досягається натисненням клавіші `-`.
- очищуючи (purged): видаляються всі файли, котрі відносяться до пакунку. Це досягається натисненням клавіші `_`.

Зверніть увагу, що всі пакунки („All Packages“) видалити неможливо. Якщо ви спробуєте зробити це, у вашій системі залишаться базові пакунки.

для фіксації: Це досягається натисненням клавіші `=` і вказує `dselect` не оновлювати пакунок навіть якщо його новіші версії є в сховищах пакунків Debian, котрі ви використовуєте (це було визначено, коли ви вибрали метод доступу до пакунків та оновились, обравши відповідний пункт меню).

Точно так само ви можете відмінити фіксування пакунку, натиснувши `:`. В цьому випадку `dselect` може оновити пакунок, якщо буде доступна його новіша версія. Така поведінка є стандартною.

Ви можете вибирати різний порядок сортування пакунків, натискаючи клавішу `o` для циклічного перемикання між різними представленнями. Зазвичай пакунки сортуються за пріоритетом; всередині пріоритету пакунки сортуються за теками (секціями) архіву, у котрому вони зберігаються. В такому порядку деякі пакунки, скажімо, з секції A можуть бути представлені першими, далі йтимуть якісь пакунки з секції B, а після них — інші пакунки з секції A з нижчим пріоритетом.

Ви також можете розширити значення міток вгорі екрану натиснувши клавішу `v` (`verbose` — детально). Ця дія виведе купу тексту, який швидше за все не поміститься на екрані. Щоб його побачити, натисніть праву стрілку; щоб повернутись назад — ліву.

Якщо ви вибрали пакунок для встановлення чи видалення, наприклад `foo.deb` і цей пакунок залежить (або рекомендує) інший пакунок, наприклад `blurf.deb`, тоді `dselect` виведе допоміжне вікно поверх головного. Там ви побачите список залежних пакунків і зможете приймати запропоновані дії (встановлювати чи ні) або ж відкидати їх. Для останнього натисніть `Shift-D`; щоб повернутись до попереднього, натисніть `Shift-U`. В

будь-якому випадку, ви можете зберегти свій вибір та повернутись до головного вікна вибору, натиснувши Shift-Q.

- Повернувшись до головного меню, користувач може обрати пункт „Встановлення“ (Install), щоб розпакувати та сконфігурувати вибрані пакунки. Відповідно, користувач, котрий хоче видалити пакунки, повинен вибрати пункт меню „Видалити“ (Remove). І в будь-якому випадку ви можете вибрати „Вихід“ (Quit), щоб завершити роботу з dselect; при цьому все, що ви вибрали (для встановлення, видалення і т.д.), зберігається.

### 7.1.5 Інші інструменти для керування пакунками

#### dpkg-deb

Ця програма маніпулює архівними (.deb) файлами Debian. Ось її деякі стандартні опції:

- Вивести повний список опцій: `dpkg-deb --help`.
- Отримати список файлів у архівному файлі: `dpkg-deb --contents foo_VVV-RRR.deb`
- Видобути файли з заданого архіву в теку, визначену користувачем: `dpkg-deb --extract foo_VVV-RRR.deb tmp` розпаковує всі файли з архіву `foo_VVV-RRR.deb` до теки `tmp/`. Це зручно при перевірці вмісту пакунок в локальному каталозі без необхідності встановлення його в кореневу файлову систему.

Зауважте, що всі пакунки, що були лише розпаковані за допомогою `dpkg-deb --extract` не є коректно встановленими, натомість вам потрібно використовувати `dpkg --install`.

Більше ви зможете дізнатись, переглянувши сторінку довідки `dpkg-deb(1)`.

#### dpkg-split

Ця програма ділить великі файли пакунків на менші (наприклад, для запису на дискети) та об'єднує набір маленьких файлів назад у один великий файл. Вона може використовуватись лише в Debian-системі (точніше, в системі, що містить пакунок `dpkg`), оскільки вона викликає програму `dpkg-deb`, щоб розділити пакунок Debian на складові частини.

Наприклад, щоб розділити великий „.deb“-файл на N частин,

- Запустіть команду `dpkg-split --split foo.deb`. Вона створить N файлів довжиною приблизно 460 КБайт в поточному каталозі.
- Скопіюйте ці файли на дискети.
- Скопіюйте вміст дискет на жорсткий диск на іншій машині.
- Об'єднайте шматки файлу командою `dpkg-split --join "foo*"`.

## 7.2 Debian стверджує, що може оновити запущену програму; яким чином він це робить?

Ядро (файлова система) в Debian GNU/Linux підтримує заміну файлів, що використовуються.

Ми також постачаємо програму під назвою `start-stop-daemon`, котра використовується для запуску демонів під час завантаження системи, чи їх зупинки під час зміни рівня запуску ядра (наприклад, від багатокористувацького режиму в однокористувацький чи ж при вимкненні). Та ж сама програма використовується встановлювальними сценаріями, коли встановлюється новий пакунок, що містить демона: при зупинці запущених демонів та їх перевантаженні у разі необхідності.

## 7.3 Як мені дізнатись, які пакунки встановлено в системі?

Щоб взнати стан всіх встановлених пакунків у Debian-системі, виконайте команду

```
dpkg --list
```

Вона виведе однорядкове резюме для кожного пакунку, включаючи 2-літерний код стану встановлення (пояснюється у заголовку), назву пакунку, встановлену версію та короткий опис.

Щоб дізнатись про стан встановлення пакунків, чії назви збігаються з набором літер або виразом, запустіть команду:

```
dpkg --list 'foo*'
```

Для отримання детальнішого звіту про певний пакунок, наберіть команду:

```
dpkg --status packagename
```

## 7.4 Як мені дізнатись, що за пакунок встановив котрийсь файл?

Щоб взнати назву пакунку, що встановив певний файл виконайте:

- `dpkg --search filename`

Вона шукає файл `filename` серед встановлених пакунків (на даний момент це еквівалентно пошуку серед файлів з розширенням `.list` в теці `/var/lib/dpkg/info` та виводу тих назв пакунків, котрі містять заданий файл).

Швидшою альтернативою є використання утиліти `dpkg-query`.

- `zgrep foo Contents-ARCH.gz`

Ця команда шукає файли, що містять підланцюжок `foo` в своїй повній назві. Файли `Contents-ARCH.gz` (де `ARCH` — це назва архітектури) розташовані у основних каталогах (`main`, `non-free`, `contrib`) на FTP-сайті Debian (наприклад, в `/debian/dists/sarge`). Файли `Contents` містять дані лише про ті пакунки, котрі розташовані у їх підтеці. Таким чином користувачеві може знадобитись здійснювати пошук не в одному файлі `Contents`, щоб знайти пакунок, котрий містить файл `foo`.

Цей метод має ту перевагу перед `dpkg --search`, бо пошук здійснюється навіть у пакунках, котрі не встановлені у вашій системі

- `apt-file search foo`

Подібно до вищенаведеного, шукаються файли, що містять ланцюжок чи регулярний вираз `foo` в повній назві. Перевага над попереднім прикладом полягає в тому, що вам не потрібно отримувати файли `Contents-ARCH.gz`, оскільки вони оновлюються автоматично для всіх джерел, перелічених у `/etc/apt/sources.list` під час запуску (як `root`) `apt-file update`.

## 7.5 Чому не видаляється `foo-data` під час видалення `foo`? Як мені очистити систему від старих пакунків, що вже не використовуються?

Деякі пакунки розділені на власне програму (`foo`) та дані (`foo-data`). Це справедливо для багатьох ігор, мультимедійних додатків та словників і введено в Debian з тих пір, коли користувачі захотіли мати доступ до даних не встановлюючи програму, або ж до програми, котра б запускалась без даних, роблячи їх необов'язковими.

Подібні ситуації виникають і з бібліотеками: зазвичай вони встановлюються, якщо від них залежать програми з вибраного пакунка. Коли такий пакунок видаляється, пакунки з бібліотеками залишаються в системі. Або у випадку, коли пакунок з програмою перестає бути залежним від пакунка з бібліотекою.

В цьому випадку `foo-data` не залежить від `foo`, отож, якщо ви видаляєте пакунок `foo`, `foo-data` не буде автоматично видалено більшістю інструментів для керування пакунками. Це саме відбувається при використанні пакунків з бібліотеками. Це необхідно ще й для того, щоб уникнути циклічних залежностей. Якщо ви використовуєте `aptitude` (див. 'aptitude' на стор. 42) у якості керівника пакунками, він, окрім іншого, може відслідковувати автоматично встановлені пакунки та видаляти їх, якщо в системі не залишилось пакунків, що їх потребують.





## Розділ 8

# Підтримування вашої системи Debian в сучасному стані

Метою Debian є забезпечення постійних оновлень та безупинне підвищення безпеки. Ми робимо все можливе, щоб оновлення до нової версії було максимально простою процедурою. У тому випадку, якщо під час процесу оновлення щось сталося, система попереджає користувача про це та пропонує шляхи вирішення можливої проблеми.

Вам також потрібно прочитати примітки до нової версії (Release Notes), документ, що описує специфічні деталі оновлення, котрий доступний як на компакт-дисках усіх версій Debian, так і в тенетах за адресою <http://www.debian.org/releases/stable/releasenotes>.

### 8.1 Як я можу оновити свою збірку Debian 1.3.1 (чи давнішу), що базується на libc5, до 2.0 (чи новішої), котрі базуються на libc6?

Є декілька можливостей оновлення:

- За допомогою простого сценарію оболонки, що називається `autoup.sh` ви можете оновити всі найважливіші пакунки. Після того, як сценарій завершить свою роботу, ви можете встановити всі інші пакунки за допомогою `dselect`. Це рекомендований метод, хоча й не єдиний.

Найновіші версії `autoup.sh` на даний момент можна знайти за наступними адресами:

- <http://www.debian.org/releases/2.0/autoup/>
- <http://www.taz.net.au/autoup/>
- <http://debian.vicnet.net.au/autoup/>
- Дотримуйтесь вказівок, наведених в Debian libc5 to libc6 Mini-HOWTO (<http://debian.vicnet.net.au/autoup/HOWTO/libc5-libc6-Mini-HOWTO.html>) та оновіть найважливіші пакунки вручну. Сценарій `autoup.sh` базується на цьому документі, отож цей метод працюватиме приблизно так само, як і попередній.

- За допомогою apt, що базується на libc5. АРТ в дечому може замінити dselect. На даний момент він працює лише в режимі командного рядка або як метод доступу dselect. libc5 ви можете знайти в теці dists/slink/main/upgrade-older-i386 архівів Debian.
- Використовуючи лише dselect, без попереднього оновлення пакунків вручну. По можливості, використовувати цей метод НЕ рекомендується, оскільки dselect не визначає оптимальний порядок встановлення пакунків. АРТ в цьому випадку працює значно краще та безпечніше.

## 8.2 Як я можу підтримувати систему Debian у найновішому стані?

Найпростіше здійснити анонімний вхід до ftp-серверу з архівом Debian, знайти необхідний файл, стягнути його та встановити за допомогою dpkg. Зауважте, що dpkg встановить оновлені файли тут же, на працюючій системі. Іноді отриманий пакунок може вимагати встановлення іншого пакунку; в цьому випадку завершити операцію не вдасться поки той пакунок не буде встановлено.

Багато людей вважають, що оновлення в такий спосіб займає надто багато часу, оскільки Debian розвивається надто швидко — щотижня оновлюється сотня, а то й більше пакунків. Це число таке велике в основному через незначні оновлення. Щоб залишатись „на передовій“, багато людей використовують більш автоматизовані методи. Для цього існують декілька пакунків:

### 8.2.1 aptitude

АРТ — це досконалий інтерфейс для системи керування пакунками Debian. Його можливості включають повне впорядкування встановлення, можливість доступу до багатьох джерел та деякі інші унікальні функції, описані в посібнику користувача, що знаходиться в /usr/share/doc/apt-doc/guide.html/index.html (у вас повинен бути встановлений пакунок apt-doc).

aptitude — це рекомендований керівник пакунками для системи Debian GNU/Linux. Це псевдографічний інтерфейс до АРТ, що використовує бібліотеку curses. З його допомогою задачі керування пакунками вирішуються швидкими та легкими методами.

Перед тим, як ви зможете використовувати aptitude, вам доведеться відредагувати файл /etc/apt/sources.list, змінивши його відповідним чином. Якщо ви бажаєте оновитись до останньої стабільної версії Debian, ви, найімовірніше, виберете за джерело щось на кшталт

<http://http.us.debian.org/debian stable main contrib non-free>

Ви можете замінити [http.us.debian.org](http://http.us.debian.org) назвою найшвидшого для вас дзеркала Debian. Переглянувши список дзеркал за адресою <http://www.debian.org/misc/README.mirrors> можна отримати більше інформації.

Більше детально про це розповість сторінка довідки `sources.list(8)`.

Щоб оновити вашу систему, запустіть спочатку

```
aptitude update
```

, після чого

```
aptitude dist-upgrade
```

Дайте відповіді на всі необхідні запитання і вашу систему буде модернізовано. Перегляньте також ‘aptitude’ на стор. 42.

## 8.2.2 apt-get, dselect та apt-cdrom

apt-get — це інструмент командного рядка для обробки пакунків; його метод в dselect — це інтерфейс до АРТ через програму dselect. Обидва вони пропонують простіший та безпечніший шлях для встановлення та оновлення пакунків.

Щоб використовувати apt-get, встановіть пакунок apt та відредагуйте /etc/apt/sources.list, змінивши його відповідним чином, наприклад так, як описано в ‘aptitude’ на стор. 50.

Далі запустіть

```
apt-get update
```

та

```
apt-get dist-upgrade
```

Дайте відповіді на всі необхідні запитання і вашу систему буде модернізовано. Перегляньте також сторінку довідки apt-get(8), а також ‘АРТ’ на стор. 40.

Щоб використати АРТ разом з dselect, виберіть метод доступу АРТ на екрані вибору методу доступу dselect (опція 0) та вкажіть, який список джерел слід використовувати. Конфігураційним файлом є /etc/apt/sources.list. Див. також ‘dselect’ на стор. 42.

Якщо ви хочете встановлювати пакунки з КД, ви можете використовувати apt-cdrom. Деталі описуються у Release Notes, глава Setting up for an upgrade from a local mirror („Налаштування для оновлення з локального дзеркала“).

Зверніть, будь ласка, увагу, що коли ви отримуєте та встановлюєте пакунки, їх копії зберігаються в теці /var. Щоб вберегти ваш розділ від переповнення не забувайте видаляти надлишкові файли за допомогою apt-get clean та apt-get autoclean, або ж пересувати їх де-інде (підказка: за допомогою apt-move).

### 8.2.3 dpkg-ftp

Це більш давній метод доступу для програми `dselect`. Його можна викликати з `dselect`, що дасть користувачеві можливість завантажувати файли та встановлювати їх за один крок. Щоб зробити це, виберіть тип доступу `ftp` та визначте назву й теку віддаленого вузла. `dpkg-ftp` автоматично завантажить вказані файли (вибрані як в поточному сеансі роботи з `dselect` так і раніше).

Зауважте, що на відміну від програми `mirror`, `dpkg-ftp` завантажує не цілий сайт, а лише ті файли, котрі ви вказали та ті, що потребують оновлення.

`dpkg-ftp` є дещо застарілим. Натомість вам краще використовувати метод доступу АРТ з рядками `ftp://URL` у файлі `sources.list`.

### 8.2.4 mirror

Щоб вибрати визначені користувачем частини дерева каталогів з заданого користувачем вузла за допомогою анонімного `ftp`-з'єднання можна використати сценарій, написаний на Perl та його додаткову керуючу програму, що називається `mirror-master`.

`mirror` є надзвичайно зручним для завантаження великих масивів програмного забезпечення. Після того, як файли були вперше завантажені з сайту, на локальному комп'ютері створюється файл під назвою `.mirrorinfo`. Всі зміни у віддаленій файлової системі автоматично відслідковуються сценарієм, котрий порівнює цей файл з аналогічним на віддаленій системі та завантажує лише змінені файли.

Загалом програмою `mirror` зручно користуватись для оновлення локальних копій віддалених дерев тек. Отримані файли не обов'язково повинні бути архівами Debian. Оскільки `mirror` написана на Perl, вона може бути запущена також і на інших системах окрім Unix. В програму включено механізм виділення назв файлів, що відповідають заданим користувачем ланцюжкам. Програма зручніша для завантаження цілих дерев каталогів, а не окремих файлів.

### 8.2.5 dpkg-mountable

`dpkg-mountable` додає метод доступу `mountable` до списку методів `dselect`, котрий дозволяє вам встановлювати пакунки з будь-якої файлової системи, визначеної у файлі `/etc/fstab`. Наприклад, архів може знаходитись на звичайному дисковому розділі, чи на NFS-сервері, що при потребі автоматично підмонтовується та відмонтовується.

Він також має деякі додаткові можливості, котрих немає у стандартних методах доступу `dselect`, як наприклад підтримка локального дерева тек (як паралельно до головної збірки, так і повністю окремого), та отримання лише потрібних пакунків на відміну від тривалого рекурсивного сканування каталогів, а також журналювання всіх дій `dpkg` під час встановлення.

### 8.3 Чи повинен я переходити у однокористувацький режим при оновленні пакунків?

Ні, пакунки можуть оновлюватись на місці, навіть у працюючій системі. В Debian є програма `start-stop-daemon`, котра при необхідності в процесі оновлення перезапускає запущений процес.

### 8.4 Чи потрібно мені зберігати на диску всі ці `.deb`-файли?

Ні. Якщо ви завантажили файли на ваш диск (що не завжди є необхідним, дивись вище опис `dpkg-ftp`), ви можете видалити їх після встановлення.

### 8.5 Як я можу вести список всіх пакунків, котрі я встановив у системі? Я хотів би знати коли і які оновлення або видалення відбувались!

Виклик `dpkg` з параметром `—log` змушує `dpkg` реєструвати оновлення та дії. В при цьому реєструються не лише виклики `dpkg` (наприклад,

```
2005-12-30 18:10:33 install hello 1.3.18 2.1.1-4
```

), але й результати цих викликів (наприклад,

```
2005-12-30 18:10:35 status installed hello 2.1.1-4
```

) Якщо ви хотіли б реєструвати всі виклики `dpkg` (навіть ті, що здійснюються програмами-інтерфейсами на кшталт `aptitude`), додайте рядок

```
log /var/log/dpkg.log
```

до файлу `/etc/dpkg/dpkg.cfg`. Впевніться, що створений файл журналу періодично зазнає ротації. Якщо ви використовуєте `logrotate`, цього можна досягти створивши файл `/etc/logrotate.d/dpkg` з таким вмістом:

```
/var/log/dpkg {  
    missingok  
    notifempty  
}
```

Більше детально про журналювання в `dpkg` можна дізнатись зі сторінки довідки `dpkg(1)` `aptitude` реєструє всі встановлення, видалення та оновлення, котрі збирається виконати в файлі `/var/log/aptitude`. Однак зауважте, що результати цих дій в вищенаведений файл не записуються!

Іншим методом для запису ваших дій є запуск сесії управління пакунками за допомогою програми `script(1)`.

## Розділ 9

# Debian і ядро

### 9.1 Чи можу я встановити та скомпілювати ядро без специфічних для Debian вдосконалень?

Так.

Є лише одна поширена пастка: С-бібліотеки Debian компонуються з найновішими стабільними версіями заголовків ядра. Якщо у вас виникне потреба скомпілювати програму з новішими заголовками, вам буде потрібно або оновити пакунок, що містить заголовки (`libc6-dev`), або використовувати нові заголовки з розпакованого найновішого ядра. Тому, якщо джерельні коди ядра розташовані в `/usr/src/linux`, то ви повинні додати параметр `-I/usr/src/linux/include/` до вашого командного рядка під час компіляції.

### 9.2 Які інструменти пропонує Debian для компонування нестандартних ядер?

Користувачі, котрі хочуть (або повинні) збирати нестандартні ядра, можуть завантажити пакунок `kernel-package`. Він містить скрипт для створення пакунка з ядром та забезпечує можливість створити Debian-пакунок `kernel-image`-версія простим запуском команди

```
make-kpkg kernel_image
```

у верхній теці джерельних кодів ядра. Довідка є доступною шляхом виконання команди

```
make-kpkg --help
```

та на сторінці підручника `make-kpkg(1)`.

Користувачі повинні окремо завантажити джерельні коди найсвіжішого ядра (або іншого, за їхнім вибором) з їхнього улюбленого сайту Linux-архіву, якщо пакунок `kernel-source-version` чомусь недоступний (де `version` означає версію ядра).

Детальні інструкції щодо використання пакунку `kernel-package` подано у файлі `/usr/share/doc/kernel-package/README.gz`. Якщо коротко, то вам потрібно:

- Розпакувати джерельні коди ядра та перейти у новостворену теку.
- Налаштувати конфігурацію ядра за допомогою однієї з команд:
  - `make config` (для чисто текстового інтерфейсу „один-рядок-за-раз“).
  - `make menuconfig` (для псевдографічного інтерфейсу, що базується на бібліотеці `ncurses`). Зауважте, що при цьому у вас має бути встановлений пакунок `libncurses5-dev`.
  - `make xconfig` (для інтерфейсу X11). Ця опція відповідно вимагає встановлених паунків X, а також Tcl/Tk.

Будь-який з цих кроків створює новий файл `.config` у верхній теці джерельних кодів ядра.

- Виконати команду: `make-kpkg -rev Custom.N kernel_image`, де N — це номер редакції, що вказується користувачем. Цим буде сформовано новий архівний файл Debian з версією перегляду Custom.N, наприклад, `kernel-image-2.2.14_Custom.1_i386.deb` для Linux-ядра 2.2.14 і архітектури i386.
- Встановити створений пакунок.
  - Запустити `dpkg --install ../kernel-image-VVV_Custom.N_i386.deb`, щоб встановити ядро. Встановлювальний сценарій:
    - \* при необхідності запустить завантажувач (`grub`, `LILO` чи якийсь інший),
    - \* встановить нестандартне ядро в `/boot/vmlinuz_VVV-Custom.N`, та створить необхідні символічні відсилачі до попередніх версій ядра.
    - \* запитає користувача, чи створювати завантажувальну дискету. Ця дискета міститиме лише ядро (див. ‘Як мені створити нестандартну завантажувальну дискету?’ на стор. 56).
  - Щоб використати другий завантажувач, `loadlin`, скопіюйте цей образ в інше місце (наприклад, розділ MS-DOS).

### 9.3 Як мені створити нестандартну завантажувальну дискету?

В розв’язанні цієї задачі вам дуже допоможе пакунок `boot-floppies`, що знаходиться в секції `admin` FTP-архіву Debian. Сценарії оболонки, що знаходяться у цьому пакунку, формують завантажувальні дискети у форматі SYSLINUX. Це дискети, відформатовані під MS-DOS, чиї завантажувальні сектори були модифіковані таким чином, що вони можуть завантажити Linux (чи будь-яку іншу операційну систему, що описана у файлі `syslinux.cfg`). Інші сценарії в цьому пакунку створюють відновлювальні диски, або навіть можуть перестворювати базові.

Більше інформації ви зможете знайти у файлі `/usr/doc/boot-floppies/README` після того, як встановите пакунок `boot-floppies`.



## 9.4 Які особливі можливості пропонує Debian для роботи з модулями ядра?

Пакунок `modconf` забезпечує сценарій (`/usr/sbin/modconf`), котрий може використовуватись для налагодження конфігурації модулів. Цей сценарій пропонує меню-подібний інтерфейс, запитуючи користувача про деталі завантажуваних драйверів пристроїв у системі. Відповіді користувача використовуються для налаштування файлу `/etc/modules.conf` (в ньому перелічені псевдоніми та інші аргументи, що повинні використовуватись разом з різноманітними модулями) через файли в `/etc/modutils/` та `/etc/modules/` (де перераховуються модулі, що мають завантажитись при ввімкненні системи).

Пакунок `modconf`, як і (нові) файли `Configure.help`, котрі зараз доступні для підтримки в конструюванні власних ядер, поставляється з набором файлів довідки (в `/usr/lib/modules_help/`), де подано детальну інформацію про аргументи, котрі можна передати будь-якому з модулів.

## 9.5 Чи можу я безпечно видалити старий пакунок ядра, і якщо можу, то як?

Так. Скрипт `kernel-image-NNN.prerm` перевіряє, чи ядро, котре у вас зараз запущено, не є тим самим, яке ви збираєтесь видалити і намагається видалити вказане ядро. Таким чином ви можете видалити небажані пакунки образів ядра за допомогою такої команди:

```
dpkg --purge --force-remove-essential kernel-image-NNN
```

(NNN, звісно, потрібно замінити версією та номером редакції вашого ядра).



## Розділ 10

# Налаштування вашої Debian GNU/Linux

### 10.1 Як мені переконатись, що всі програми використовують однаковий розмір паперу?

Встановіть пакунок `libpaper1` і він встановить однаковий розмір паперу для всієї системи. Ця настройка буде збережена у файлі `/etc/papersize`.

Користувачі можуть перепризначити розмір паперу за допомогою змінної середовища `PAPERSIZE`. Деталі описано на сторінці довідки `papersize(5)`.

### 10.2 Як мені надати доступ до периферійних пристроїв не ставлячи під загрозу безпеку системи?

Багато пристроїв у теці `/dev` належать деяким наперед визначеним групам. Наприклад, `/dev/fd0` належить групі `floppy`, а `/dev/dsp` — групі `audio`.

Якщо ви хочете дати якомусь користувачеві доступ до цих пристроїв, просто включіть його у відповідну групу, наприклад ось таким чином:

```
adduser user group
```

В цьому випадку вам не потрібно буде змінювати права доступу до пристрою.

### 10.3 Яким чином в Debian задати консольний шрифт при завантаженні?

Цю можливість підтримують пакунки `kbd` та `console-tools`, відредагуйте конфігураційні файли `/etc/kbd/config` або ж `/etc/console-tools/config`.

## 10.4 Як я можу змінити стандартні настройки додатків для X11?

Графічні програми для Debian записують стандартні налаштування додатків у теку `/etc/X11/app-defaults/`. Якщо ви хочете змінити їх глобально, запишіть ваші зміни саме у ці файли. Вони позначаються як конфігураційні, отож не будуть замінені під час оновлень.

## 10.5 Схоже, що кожна збірка завантажується по-своєму. Розкажіть, як це робить Debian.

Як і всі інші Юнікси, Debian завантажується запускаючи програму `init`. Її конфігураційний файл (`/etc/inittab`) вказує, що першим має виконуватись сценарій `/etc/init.d/rcS`. Він запускає всі інші сценарії з теки `/etc/rcS.d/`, виконуючи їх безпосередньо, або ж відбрунковуючи, в залежності від їх розширення, щоб провести ініціалізацію системи (наприклад, перевірити та змонтувати файлові системи, завантажити модулі, запустити мережеві служби, налаштувати годинник та ін.). Далі, щоб дотримуватись сумісності, він запускає файли (за винятком тих, що містять крапки в назві) з теки `/etc/rc.boot/`. Ця тека, зазвичай, вважається зарезервованою для адміністраторських цілей і використання його пакунками є ознакою поганого тону.

Після завершення завантажувального процесу `init` запускає всі стартові сценарії з теки, котра визначається типовим рівнем запуску (який задається відповідним записом `id` у `/etc/inittab`). Як і більшість System V-сумісних Юніксів, Linux має 7 рівнів запуску:

- 0 (зупинка системи),
- 1 (однокористувацький режим),
- від 2 до 5 (різні багатокористувацькі режими), та
- 6 (перезавантаження системи).

Системи Debian як правило постачаються з `id=2`. Це означає, що типовим рівнем запуску буде '2', з багатокористувацьким режимом, при котрому будуть виконані сценарії з теки `/etc/rc2.d/`.

Фактично в теках `/etc/rcN.d/` розташовуються не сценарії, а символічні відсилачі на сценарії в теці `/etc/init.d/`. Їх назви підбираються таким чином, щоб вказувати, в якій послідовності будуть запускатись відповідні сценарії з теки `/etc/init.d/`. Взагалі, перед переходом на певний рівень запуску, спершу виконуються сценарії, що починаються на літеру 'K'; вони зупиняють (англ. kill) непотрібні служби. Далі запускаються всі сценарії, що починаються на літеру 'S'; вони власне запускають необхідні служби. Двоцифрове число, що йде після 'K' або 'S' вказує на порядок запуску сценарію. Сценарії з меншими числами запускаються першими.

Такий підхід працює тому, що сценарії в теці `/etc/init.d/` приймають аргументи з набору `start`, `stop`, `reload`, `restart` та `force-reload` і виконують відповідні завдання (англ. `start` — запустити, `stop` — зупинити, `reload` — перевантажити, `restart` — перезавпустити та `force-reload` — перевантажити примусово). Ці сценарії можуть виконуватись при завантаженні системи для контролю різноманітних процесів.

Наприклад, з аргументом `reload` команда

```
/etc/init.d/sendmail reload
```

відправить демону `sendmail` сигнал перечитати заново його конфігураційний файл. (Debian постачає `invoke-rc.d` в якості оболонки для запуску сценаріїв з `/etc/init.d/`.)

## 10.6 Виходить, Debian не використовує `rc.local` для налаштування процесу завантаження; що ж в такому випадку використовується?

Припустимо, при завантаженні чи при зміні рівня запуску нам потрібно запускати сценарій `foo`. Системний адміністратор в такому випадку повинен:

- Помістити сценарій `foo` до теки `/etc/init.d/`.
- Запустити команду `update-rc.d` з необхідними аргументами, щоб створити відповідні відсилачі між теками `rc?.d` та `/etc/init.d/foo` (де '?' — число від 1 до 6, що означає рівень запуску System V).
- Перезавантажити систему.

Команда `update-rc.d` створює відсилачі на сценарії теки `/etc/init.d/` у теці `rc?.d`. Кожен відсилач починається з літери 'S' або 'K', за якими йде номер та власне назва сценарію. Сценарії, що починаються на 'S' з теки `/etc/rcN.d/` виконуються при завантаженні рівня запуску N; ті, що починаються на 'K' — при його зупинці.

Отож сценарій `foo` можна запустити при завантаженні системи, помістивши його в `/etc/init.d/` та встановивши відсилачі за допомогою команди `update-rc.d foo defaults 19`. Аргумент `'defaults'` позначає типові рівні запуску — від 2 до 5. Аргумент `'19'` вказує, що сценарій запуститься перед будь-яким іншим, котрий починається з номера 20 або більшого.

## 10.7 Що система керування робить з пакунками, котрі містять конфігураційні файли інших пакунків?

Хтось з користувачів може захотіти створити, наприклад, нову службу, встановивши групу пакунків Debian та створивши локальний архів, з їх конфігураційними файлами. Взагалі-то це не є хорошою ідеєю, тому що `dpkg` не знатиме про конфігураційні файли з такого

архіву і може перезаписати суперечливі конфігурації при оновленні одного з пакунків такої групи.

Створіть натомість локальний пакунок, що змінює конфігураційні файли такої групи пакунків. Після цього `dpkg` та інші системи керування пакунками бачитимуть, що ці файли були модифіковані локальним адміністратором та не старатимуться перезаписувати їх при оновленні.

## 10.8 Як я можу перезаписати файл, встановлений пакунком, так щоб використовувалась інша його версія?

Припустимо, адміністратор, або локальний користувач хочуть використовувати програму `login-local` замість `login`, котра поставляється з пакунком `login`.

НЕ:

- перезаписуйте файл `/bin/login` програмою `login-local`.

Система керування пакунками не знатиме про цю зміну і просто перезапише ваш модифікований `/bin/login` при будь-якому встановленні чи оновленні пакунку `login` (чи будь-якого іншого, котрий постачає програму `/bin/login`).

Натомість

- виконайте:

```
dpkg-divert --divert /bin/login.debian /bin/login
```

для того, щоб змусити всі наступні встановлення пакунку `login` записувати файл `/bin/login` у `/bin/login.debian`.

- після цього запустіть:

```
cp login-local /bin/login
```

щоб помістити вашу власну програму на потрібне місце.

За детальнішою інформацією зверніться до сторінки підручника `dpkg-divert(8)`.

## 10.9 Як мені додати свій пакунок до списку відомих системі керування пакунками?

Запустіть команду:

```
dpkg-scanpackages BIN_DIR OVERRIDE_FILE [PATHPREFIX] > my_Packages
```

де:

- BIN-DIR — це тека, в котрій зберігаються файли архівів Debian (що як правило мають розширення „.deb“)
- OVERRIDE\_FILE — це файл, що редагується супроводжуваними збірками та зберігається як правило на FTP-сайтах архівів Debian як `indices/override.main.gz` для пакунків зі збірки `main`. Для локальних пакунків ви можете ігнорувати цю опцію.
- PATHPREFIX — це необов'язковий ланцюжок, що може бути доданий до файлу `my_Packages` при його створенні.

Після того, як ви сформуєте файл `my_Packages`, накажіть системі керування пакунками використовувати його за допомогою команди:

```
dpkg --merge-avail my_Packages
```

Якщо ви використовуєте АРТ, ви також можете додати локальне сховище до вашого файлу `sources.list(5)`.

## 10.10 Деяким користувачам подобається `mawk`, іншим `gawk`; одні користуються `vim`, інші `elvis`; дехто уподобав `trn`, а дехто `tin`; як Debian підтримує такі різноманітності?

Є декілька випадків, коли два пакунки забезпечують дві різні варіації програм, кожна з котрих володіє однаковою базовою функціональністю. Користувачі по якихось причинах можуть надавати перевагу одній перед іншою. В той же час інші користувачі на тій самій системі можуть мати інші вподобання.

Debian використовує „віртуальну“ систему пакунків, щоб дозволити адміністраторам (або дати можливість користувачам) обирати улюблені інструменти за наявності двох чи більше програм, котрі пропонують однакову базову функціональність, підтримуючи при цьому залежності без вказування конкретного пакунку.

Наприклад, у системі є дві різні версії програм для читання новин. Паунок серверу новин може рекомендувати котрусь із них, але кінцевий вибір між `tin` та `trn` робить кожен користувач особисто. Це забезпечується тим, що обидва ці пакунки забезпечують віртуальний паунок `news-reader`. Котру з програм використовувати вказує відсилач `/etc/alternatives/news-reader` на вибраний файл, наприклад `/usr/bin/trn`.

Сам по собі відсилач не здатен підтримувати повноцінне використання альтернативних програм; зазвичай сторінки довідки та можливо інші використовувані файли також повинні підтримуватись. Сценарій `Perl update-alternatives` дає можливість переконатись, що всі файли, пов'язані з певним паунком, вибрані як типові для системи.

Наприклад, щоб перевірити, які файли забезпечує `x-window-manager`, запустіть:

```
update-alternatives --display x-window-manager
```

Якщо ви хочете щось змінити, виконайте:

```
update-alternatives --config x-window-manager
```

та дотримуйтесь інструкцій на екрані (як правило, натискайте номер елемента, котрий вам більше подобається).

Якщо з якоїсь причини паунок не зареєстрував себе, як віконний менеджер, або ж ви хочете використовувати менеджер з теки `/usr/local`, то екранне меню не буде містити потрібного вам елемента. Ви можете оновити конфігурацію за допомогою опцій командного рядка, як наприклад:

```
update-alternatives --install /usr/bin/x-window-manager \  
x-window-manager /usr/local/bin/wmaker-cvs 50
```

Перший аргумент опції `'--install'` є символічним відсилачем, що вказує на `/etc/alternatives/ІМ'Я`, де `ІМ'Я` — другий аргумент. Третій аргумент — це програма, на котру повинен вказувати `/etc/alternatives/ІМ'Я`, четвертий — пріоритет (чим більше значення, тим вища ймовірність автоматичного вибору даної програми).

Щоб видалити додану альтернативу, просто запустіть:

```
update-alternatives --remove x-window-manager /usr/local/bin/wmaker-cvs
```



## Розділ 11

# Отримання підтримки для Debian GNU/Linux

### 11.1 Яка ще документація існує для системи Debian?

- Інструкції по встановленню для поточного випуску: перегляньте <http://www.debian.org/releases/stable/installmanual>.
- Довідник Debian GNU/Linux торкається багатьох аспектів системного адміністрування за допомогою прикладів команд оболонки. Базові навчальні програми, підказки та інша інформація з багатьох різноманітних тем від системного адміністрування до програмування постачається у пакунку `debian-reference`, також доступна за адресою <http://www.debian.org/doc/user-manuals#quick-reference>.
- Підручник політики (Policy manual) документує стратегічні вимоги збірки, як наприклад структуру та вміст архіву Debian; деякі проблеми дизайну операційної системи і т.п. Він також включає технічні вимоги до пакунку, котрих потрібно дотримуватись для включення його в збірку, та документує основні технічні аспекти двійкових та джерельних пакунків Debian.  
Його ви можете отримати з пакунку `debian-policy` або за адресою <http://www.debian.org/doc/devel-manuals#policy>.
- Документація розроблена в рамках Проекту документування Debian (Debian Documentation Project). Вона доступна за адресою <http://www.debian.org/doc/> та включає посібники користувача, адміністратора та убезпечення операційної системи Debian GNU/Linux.
- Документація на встановлені пакунки Debian: переважна більшість пакунків мають файли документації, що встановлюються до `/usr/doc/ПАКУНОК`.
- Проект документування Linux: пакунок `doc-linux` встановлює всі найсвіжіші версії документів HOWTOs та mini-HOWTOs від Linux Documentation Project (<http://www.tldp.org/>).

- Сторінки підручника у стилі Unix: Переважна більшість команд мають сторінки довідки в стилі оригінальних man-файлів Unix. Вони поділені на секції у каталозі man: наприклад, `foo(3)` вказує на сторінку підручника, котра знаходиться у `/usr/share/man/man3/`, і може бути викликана за допомогою команди: `man 3 foo`, або просто `man foo`, якщо третя секція є першою, що містить сторінку `foo`.

Які з підтек `/usr/share/man/` містять задану сторінку можна дізнатись, виконавши команду `man -w foo`.

Новачки Debian повинні пам'ятати, що сторінки підручника багатьох загальносистемних команд будуть недоступними, аж поки вони не встановлять наступні пакунки:

- `man-db`, котрий містить саму програму `man`, та інші програми для маніпуляцій сторінками підручника;
  - `manpages`, що містить системні сторінки підручника (див. 'Як справи в Debian з підтримкою інших мов, окрім англійської?' на стор. 19).
- Інформаційні сторінки у стилі GNU: Користувачка документація для багатьох команд, особливо інструментів GNU доступна не лише у man-сторінках, але й у файлах `info`, котрі можуть бути прочитані інструментом GNU `info`, запуском `M-x info` у середовищі GNU Emacs, чи будь-яким іншим переглядачем сторінок Info.
- Їхньою головною перевагою над сторінками підручника є те, що вони базуються на гіпертекстовій системі. Проте вони не потребують тенет; сторінки `info` можуть бути прочитані з текстової консолі. Ця система була спроектована Річардом Столлменом та передувала WWW.

Зауважте, що ви можете знайти чимало документації у своїй системі за допомогою веб-переглядача та команд `dwww`, `dhelp` або `doccentral` з відповідних пакунків.

## 11.2 Чи є якісь мережеві ресурси для обговорення Debian?

Так. Фактично, основним методом підтримки, що надає Debian своїм користувачам є підтримка за допомогою електронної пошти.

### 11.2.1 Списки розсилки

Існує велика кількість пов'язаних з Debian списків розсилки (<http://www.debian.org/MailingLists/>).

В системі з встановленим пакунком `doc-debian` повний їх список можна знайти у файлі `/usr/share/doc/debian/mailling-lists.txt`.

Списки розсилки Debian отримують назви за таким зразком: `debian-тема-списку`. Наприклад, `debian-announce`, `debian-user`, `debian-news`. Щоб підписатись до будь-якого з них, відправте листа за адресою `debian-тема-списку-request@lists.debian.org` з словом `subscribe` у полі Subject заголовку листа. Не забувайте додавати `-request` до адреси електронної пошти,

якщо ви використовуєте такий метод щоб підписатись чи відписатись. В іншому випадку ваш лист потрапить до власне розсилки, що може роздратувати, або ж розгубити вас, в залежності від вашої точки зору.

Якщо у вас є переглядач Тенет, що підтримує форми, ви можете з їх допомогою підписатись (<http://www.debian.org/MailingLists/subscribe>) на списки розсилки або відписатись (<http://www.debian.org/MailingLists/unsubscribe>) від них.

Список керівників e-mail-адреса — <listmaster@lists.debian.org>, сюди можна звертатись у випадку виявлення будь-яких проблем.

Списки розсилки є публічними форумами. Всі електронні листи, відправлені до списків, копіюються до загальнодоступних архівів, щоб будь-хто (навіть люди не підписані на них) міг переглянути їх або здійснити пошук по їх вмісту. Будь ласка, впевніться що ви не відсилаєте жодних конфіденційних чи неліцензійних матеріалів до списку розсилки. Це стосується і таких речей, як, наприклад, електронні адреси. Зверніть особливу увагу і на той факт, що електронні адреси, з яких ви відсилаєте листи до наших списків розсилки, можуть стати відомими спамерам та використані ними для зловживань. Перегляньте політику щодо приватності в списках розсилки Debian (<http://www.debian.org/MailingLists/index.en.html#disclaimer>) щоб отримати більше інформації.

Архіви списків розсилки Debian доступні через WWW за адресою <http://lists.debian.org/>.

Як поводитись у списках розсилки?

Спілкуючись у списках розсилки Debian, дотримуйтесь, будь ласка, наступних правил:

- Не надсилайте спаму. Перегляньте Політику Debian щодо реклами в списках розсилки (<http://www.debian.org/MailingLists/#ads>).
- Не флейміть — це неввічливо. Всі розробники Debian — добровольці, що жертвують проекту свій час, енергію та кошти, стараючись спільно принести користь.
- Не використовуйте нецензурну лексику; окрім усього іншого деякі люди отримують листи за допомогою пакетного радіо, де лайливі вислови є неприпустимими.
- Переконайтесь, що ви вказуєте правильну назву списку. Ніколи не відправляйте ваші запити про підписку на список розсилки або відмову під подальшої участі в ньому безпосередньо у списки<sup>1</sup>.
- Див. ‘Як мені відправити в Debian звіт про помилку?’ на стор. 69 щоб дізнатись про відсилення повідомлень про помилки.

### 11.2.2 Супроводжуючі

Користувачі можуть адресувати свої запитання до конкретних супроводжуючих пакунків за допомогою електронної пошти. Щоб зв'язатись з супроводжуючим пакунком хуз, відішліть листа за адресою [xuz@packages.debian.org](mailto:xuz@packages.debian.org).

<sup>1</sup>Для цього необхідно вказувати адресу [debian-назва-списку-REQUEST@lists.debian.org](mailto:debian-назва-списку-REQUEST@lists.debian.org).

### 11.2.3 Групи новин

Користувачі мають публікувати не специфічні для Debian запитання до однієї з груп новин, присвячених Linux, котрі називаються `comp.os.linux.*` та `linux.*`. Також є кілька груп новин у тенетах, наприклад на сайтах Linux Online (<http://www.linux.org/docs/usenet.html>) та LinuxJournal (<http://www.linuxjournal.com/helpdesk.php>).

## 11.3 Чи є якийсь швидкий спосіб пошуку інформації про Debian GNU/Linux?

Є набір пошукових систем, котрі обслуговують документацію, присвячену Debian:

- Пошуковий веб-сайт Debian (<http://search.debian.org/>).
- Google Groups (<http://groups.google.com/>): пошукова система для груп новин.

Наприклад, щоб дізнатись, про успіхи людей у пошуку драйверів під Debian для контролерів Promise, спробуйте здійснити пошук по фразі `Promise Linux driver`. Вам буде показано всі повідомлення, що містять цей рядок, тобто все, що люди обговорювали на цю тему. Якщо до цього ланцюжка додати слово `Debian`, ви побачите лише ті повідомлення, що стосуються саме Debian.

- Чим краще ви задаєте ключову фразу, тим точніші результати видають стандартні пошукові системи, як наприклад AltaVista (<http://www.altavista.com/>) чи Google (<http://www.google.com/>).

Наприклад, пошук по ланцюжку `cgi-perl` дасть значно детальніші пояснення про пакунок, аніж коротке поле опису у його файлі `control`.

## 11.4 Чи ведуться записи про відомі помилки?

Звіти про невіршені (та закриті) питання є загальнодоступними: Debian пообіцяв робити так, заявляючи, що „Ми не будемо приховувати проблеми“ в Суспільному договорі Debian ([http://www.debian.org/social\\_contract](http://www.debian.org/social_contract)).

Збірка Debian GNU/Linux має систему відслідковування помилок (bug tracking system — BTS), котра зберігає деталі звітів про помилки, надісланих користувачами та розробниками. Кожній помилці присвоюється число і звіт про неї зберігається до тих пір, поки проблема не буде вирішеною.

Копія цієї інформації доступна за адресою <http://www.debian.org/Bugs/>.

Поштовий сервер надає доступ до бази даних системи відслідковування помилок за допомогою електронної пошти. Щоб отримати детальніші інструкції, відправте листа на адресу [request@bugs.debian.org](mailto:request@bugs.debian.org) зі словом `help`.

## 11.5 Як мені відправити в Debian звіт про помилку?

Якщо ви знайшли помилку, прочитайте, будь ласка, інструкції щодо звітування про помилки в Debian. Ці інструкції можна отримати кількома шляхами:

- Через анонімний FTP. Дзеркальні сайти Debian містять інструкції у файлі `doc/bug-reporting.txt`.
- З тенет. Копія інструкцій доступна за адресою <http://www.debian.org/Bugs/Reporting>.
- На будь-якій системі Debian зі встановленим пакунком `doc-debian`. Інструкції знаходяться у файлі `/usr/doc/debian/bug-reporting.txt`.

Ви також можете відіслати звіт про помилку за допомогою пакунку `reportbug`, котрий проведе вас через процес створення звіту та надішле листа електронною поштою за правильною адресою, автоматично додавши деякі відомості про вашу систему. Він також покаже вам список помилок, знайдених у пакунку, про котрий ви звітуєте з тим щоб ви могли надати додаткову інформацію до існуючих звітів.

Якщо ви хочете відіслати звіт електронною поштою, надсилайте його на `<submit@bugs.debian.org>`. Перший рядок звіту має бути подібний до

```
Package: package-name
```

(замініть `package-name` назвою пакунка). Наступний рядок повинен подібним чином вказувати на версію програми:

```
Version: version-number
```

Номер версії будь-якого встановленого у вашій системі пакунку можна дізнатись за допомогою команди

```
dpkg -s package-name
```

Ця частина відноситься до так званого псевдо-заголовку. Решта повідомлення повинна містити опис помилки (робіть його помірковано детальним, будь ласка), версію вашої збірки Debian та версії інших пакунків, що мають відношення до помилки. Версію Debian можна дізнатись за допомогою команди

```
cat /etc/debian_version
```

Ви маєте отримати автоматичне підтвердження про отримання вашого звіту. Йому також буде автоматично надано номер, додано до журналу помилок та перенаправлено до списку розсилки `debian-bugs-dist`.



## Розділ 12

# Співробітництво в рамках проекту Debian

Жертвування (<http://www.debian.org/donations>) вами часу (для розробки нових пакунків, супроводу існуючих чи надання підтримки користувачам), ресурсів (для віддзеркалення FTP та WWW архівів) та коштів (для оплати нових випробовувальних стендів а також апаратного забезпечення архівів) може допомогти проекту.

### 12.1 Як я можу стати розробником програмного забезпечення Debian?

Розробка Debian відкрита для всіх і для супроводу існуючих пакунків, котрі були „покинуті“ попередніми супроводжувачами, для розробки нових та для забезпечення підтримки користувачів потрібні нові розробники з відповідним досвідом та/або бажанням вчитися.

Опис того, як можна отримати статус розробника Debian можна знайти на веб-сайті Debian в куточку нового супроводжувачого (<http://www.debian.org/devel/join/newmaint>).

### 12.2 Як я можу допомогти ресурсами проекту Debian?

Оскільки проект має на меті формування стабільного кістяка для швидкого доступу до програмного забезпечення по всій земній кулі, то нам нагально потрібні дзеркала. Бажано, але все ж не абсолютно необхідно, віддзеркалювати весь архів. За інформацією щодо необхідного дискового простору зверніться, будь ласка, до сторінки розмір дзеркала Debian mirror size (<http://www.debian.org/mirror/size>).

Значна частина віддзеркалення відбувається повністю автоматично, за допомогою сценаріїв, без будь-якої взаємодії з користувачем. Проте, несподівані перебої в роботі вашої системи можуть потребувати людського втручання.

Якщо ви маєте швидкісний канал в Internet, ресурси для віддзеркалення всієї збірки чи її частини та бажаєте затратити час (або ж знайти когось, хто може забезпечити регулярну підтримку системи), будь ласка, зверніться до <[debian-admin@lists.debian.org](mailto:debian-admin@lists.debian.org)>.

## 12.3 Як я можу допомогти фінансово проекту Debian?

Ви можете допомогти одній з двох організацій, діяльність котрих є критичною для проекту.

### 12.3.1 Software in the Public Interest (Програми для загалу)

Software in the Public Interest (SPI) це некомерційна IRS 501(c)(3) організація, сформована після того, як Фонд вільного програмного забезпечення (FSF) припинив спонсорувати Debian. Метою організації є розробка та поширення вільного програмного забезпечення.

Наші цілі дуже подібні до цілей FSF, і ми заохочуємо програмістів до застосування Загальної публічної ліцензії GNU General Public License до їхніх програм. Проте, існує деяка різниця в тому, що ми розробляємо й поширюємо збірку системи Linux, котра в багатьох технічних деталях відрізняється від системи GNU, планованої Фондом ВПЗ. Проте ми досі взаємодіємо і співпрацюємо з Фондом ВПЗ, надсилаючи їм зміни у програмному забезпеченні GNU і запрошуючи користувачів допомагати проектам FSF та GNU.

Детальніше про проект SPI можна прочитати на <http://www.spi-inc.org/>.

### 12.3.2 Free Software Foundation (Фонд вільного програмного забезпечення)

На даний момент формального зв'язку між проектом Debian та Фондом ВПЗ немає. Проте Фонд ВПЗ відповідальний за деякі найважливіші частини Debian, включаючи компілятор GNU C, GNU Emacs, та багато інших бібліотек, котрі використовуються всіма програмами у системі. Фонд ВПЗ є засновником переважної більшості вільного програмного забезпечення на сьогоднішній день: вони написали Загальну публічну ліцензію, котра застосовується до значної частини програмного забезпечення Debian і придумали проект GNU щоб створити повністю вільну Unix-систему. Debian потрібно розглядати як нащадок системи GNU.

Детальнішу інформацію про Фонд ВПЗ можна отримати на <http://www.fsf.org/>.



## Розділ 13

# Перепозищення Debian GNU/Linux як комерційного продукту

### 13.1 Чи можу я створювати та продавати компакт-диски Debian?

Будь ласка. Вам не потрібно ніякого дозволу на поширення будь-чого, випущеного нами, отож ви можете створювати ваші КД як тільки завершилось тестування. Вам не вимагаємо ніякої плати. Звісно, що всі виробники КД повинні поважати ліцензії програм, включених в Debian. Зокрема, багато програм поширюються під ліцензією GPL, котра вимагає розповсюджувати також їх джерельний код.

Ми також публікуємо список виробників КД, котрі допомагають грошами, програмами та часом проекту Debian і заохочуємо користувачів купувати диски саме в них, ця реклама є непоганою компенсацією за ваші пожертвування.

### 13.2 Чи може бути Debian запакований разом з закритим програмним забезпеченням?

Так. В той час як основні компоненти Debian є відкритим програмним забезпеченням, ми надаємо теку non-free для програм, котрі не можна вільно перепозищувати.

Виробники КД можуть поширювати програми, котрі ми помістили у цю теку, в залежності від умов ліцензії та їх особистих домовленостей з авторами відповідного програмного забезпечення. Вони також можуть поширювати закрите програмне забезпечення, котре вони отримали з інших джерел, на тому ж диску. В цьому немає нічого нового: вільне та комерційне програмне забезпечення зараз поширюється на одному диску багатьма виробниками. Звичайно, ми досі заохочуємо авторів програм випускати їхні продукти як вільне програмне забезпечення.

### 13.3 Я створюю особливу збірку Linux для „вертикального маркетингу“. Чи можу я використовувати Debian GNU/Linux як базову систему та додати мої програми поверх неї?

Так. Debian-базовані дистрибутиви створюються при тісному співробітництві як власне з проектом Debian, так і з зовнішніми учасниками. Можна використовувати структуру Custom Debian Distributions (<http://cdd.alioth.debian.org/>) для співпраці з Debian; Skolelinux (<http://www.skolelinux.org/>) є одним з таких проектів.

Одна людина створює збірку „Linux for Hams“ з спеціалізованими програмами для радіоаматорів. Він використав Debian як базову систему та додав програми для контролю передавача, відслідковування сателітів та ін. Всі додані ним програми запаковані системою керування пакунками Debian отож його користувачі можуть легко оновитись коли він випускає чергові диски.

На даний момент на ринку є кілька інших збірок, породжених Debian, як наприклад Progeny Debian, Linspire, Knoppix та Ubuntu, з дещо іншими цілями, аніж в оригінального Debian GNU/Linux, але з більшістю наших компонент у їхніх продуктах.

Debian також надає механізм, котрий дозволяє розробникам та системним адміністраторам встановлювати локальні версії обраних файлів таким чином, щоб вони не перезаписувались при оновленні відповідних пакунків. Щоб переглянути дискусію з цього приводу, див. ‘Як я можу перезаписати файл, встановлений пакунком, так щоб використовувалась інша його версія?’ на стор. 62.

### 13.4 Чи можу я запакувати комерційну програму у пакунок Debian щоб її можна було легко встановити на будь-якій Debian-системі?

Будь ласка. Система керування пакунками є вільним програмним забезпеченням; пакунки ж можуть бути відкриті або закритими — вона може встановити і ті, і інші.

## Розділ 14

# Зміни, що очікуються у новій версії Debian

### 14.1 Покращена безпека

Debian включає підтримку тіньових паролів починаючи з версії 1.3. На додачу, доступна бібліотека Приєднаних модулів аутентифікації (відома також як `libpam` (<http://www.kernel.org/pub/linux/libs/pam/>)), котра дозволяє системним адміністраторам вибирати тип авторизації в залежності від використовуваних програм, і типово налаштована автентифікувати за допомогою тіньових паролів.

Ведеться робота над включенням повної підтримки вдосконалених методів аутентифікації, таких як Kerberos, та додатковим покращенням обов'язкових механізмів контролю безпеки на кшталт SELinux, RSBAC та захисту від переповнення буфера Exec-shield чи PaX.

### 14.2 Вдосконалена підтримка неангломовних користувачів

В Debian вже є хороша підтримка неангломовних користувачів, див. ‘Як справи в Debian з підтримкою інших мов, окрім англійської?’ на стор. 19.

Ми сподіваємось знайти людей, котрі забезпечуватимуть підтримкою та перекладом ще більшу кількість мов. Деякі програми вже підтримують інтернаціоналізацію, отож нам потрібні перекладачі каталогів повідомлень. Багато програм все ще потребують правильного перекладу.

Над інтернаціоналізацією GNU-програм працює GNU Translation Project <ftp://ftp.gnu.org/pub/gnu/ABOUT-NLS>.

### 14.3 Більше архітектур

Незабаром очікується повна система Debian на інших архітектурах, як наприклад AMD64 чи SuperH. Зауважте, що розробницький архів AMD64, доступний за адресою <http://amd64>.

[debian.net/](http://debian.net/), надає встановлювальні образи та двійкові пакунки для значної частини збірки. Повна підтримка цієї архітектури буде напевно додана в etch.

## 14.4 Більше ядер

На додаток до Debian GNU/Hurd, Debian було портовано також на ядра BSD, а саме на NetBSD (<http://www.debian.org/ports/netbsd>).

## Розділ 15

# Загальна інформація про FAQ

### 15.1 Автори

Перша версія цього документу була розроблена та підтримувалась J.H.M. Dassen (Ray) та Chuck Stickelman. Авторами переписаного Debian GNU/Linux FAQ є Susan G. Kleinmann та Sven Rudolph. Після них FAQ підтримувався Santiago Vila та, пізніше, Josip Rodin. Зараз супроводжуючим є Javier Fernandez-Sanguino.

Частини інформації прийшли з/від:

- Оголошення про версію Debian-1.1, автор Bruce Perens (<http://www.perens.com/>).
- Linux FAQ, автор Ian Jackson (<http://www.chiark.greenend.org.uk/~ijackson/>).
- Архівів списків розсилки Debian (<http://lists.debian.org/>),
- довідника програмістів dpkg та підручника політики Debian (див. ‘Яка ще документація існує для системи Debian?’ на стор. 65)
- багатьох розробників, ентузіастів та бета-тестерів
- пошматованої пам’яті авторів :-)

Автором українського перекладу Debian FAQ є Mykhaylo Masik, <miwaonline@gmail.com>.

Автори хочуть висловити вдячність всім, хто допоміг у створенні цього документу.

Не надається жодних гарантій. Всі торгові марки та права належать їхнім законним власникам.

### 15.2 Зворотній зв’язок

Коментарі та доповнення до цього документу завжди радо вітаються. Будь ласка, надсилайте листи за адресою <doc-debian@packages.debian.org>, або ж відправте звіт про помил-

ку з побажаннями, котрий буде стосуватись пакунка doc-debian (<http://bugs.debian.org/doc-debian>).

### 15.3 Доступність

Останню версію цього документу можна знайти у веб-сторінках Debian за адресою <http://www.debian.org/doc/FAQ/>.

Вона також доступна для завантаження у звичайному текстовому, HTML, PostScript та PDF форматах з <http://www.debian.org/doc/user-manuals#faq>. Там також доступні кілька перекладів.

Оригінальні SGML файли, що використовувались при створенні цього документу, доступні у джерельному пакунку doc-debian, та через CVS: `pserver:anonymous@cvs.debian.org:/cvs/debian-doc/ddp/manuals.sgml/faq`

### 15.4 Формат документу

Цей документ був написаний за допомогою DebianDoc SGML DTD (переписаний з LinuxDoc SGML). Система DebianDoc SGML дозволяє нам створювати багато форматів документу з одного джерела, таким чином цей документ можна переглянути у форматах HTML, звичайний текст, TeX DVI, PostScript, PDF, або GNU info.

Інструменти для перетворення DebianDoc SGML можна знайти у пакунку debiandoc-sgml.