
Guide pratique de gravure d'un CD RedHat

Adaptation française du *Burning a RedHat CD HOWTO*

Luigi Bitonti <uknadors CHEZ yahoo POINT com>

Morten Kjeldgaard

Peter von der Ahé

Vérification et traduction française: Guillaume Lelarge

Relecture de la version française: Guillaume Hatt, Jean-Philippe Guérard

Préparation de la publication de la v.f.: Jean-Philippe Guérard

Version : 2.1.fr.1.1

Copyright © 2000 Morten KjeldgaardPeter von der Ahé

Copyright © 2002, 2003 Luigi Bitonti

Copyright © 2003, 2004 Guillaume LelargeGuillaume HattJean-Philippe Guérard

2004-11-01

Historique des versions

Version 2.1.fr.1.1	2004-11-01	GL, GH, JPG
Version 2.1.fr.1.0	2003-12-07	GL, GH
Version 2.1	2003-10-17	LB

Ajout de la Redhat 9. Correction de quelques bogues mineurs. Merci à tous ceux qui ont envoyé des commentaires et des correctifs. (*Added RedHat 9. Fixed some minor bugs.*)

Thanks to all the people who have sent in comments and patches.)

Version 2.03.fr.1.0	2003-08-18	GL, GH, JPG
Version 2.03	2003-03-10	LB

Ajout de quelques commentaires et corrections pour le guide pratique. Les mises à jour anaconda sont maintenant incluses y compris les versions 7.x et plus. (*Added some comments and fixes to the howto. The anaconda updates are now included correctly even for versions >= 7.x.*)

Version 2.02	2003-03-06	LB
--------------	------------	----

La vérification de la signature fonctionne maintenant pour les paquets ciblés pour d'autres versions de la distribution RedHat que celle utilisée pour lancer les scripts. Correction d'un bogue dans la section de comparaison de version des scripts. (*The signature checking now works for packages targeted to versions of RedHat different from the one used to run the scripts. Corrected a bug in the version comparison section of the scripts.*)

Version 2.01	2002-12-04	LB
--------------	------------	----

Deuxième édition de la nouvelle version de ce guide pratique. Tous les scripts ont été revus et nettoyés. Le script `updateDist.sh` vérifie maintenant que toutes les mises à jour ont été téléchargées avant de vérifier les signatures. (*Second release of the new version of the howto. All the scripts were reviewed and cleaned. The updateDist.sh script now checks that all the updates were downloaded before checking the signatures.*)

Version 2.0	2002-10-28	LB
-------------	------------	----

Première édition de la version 2 de ce guide pratique. (*First release of the new version (2.00) of the HOWTO*)

Ce document explique comment créer vos propres CD à partir des différentes versions de la distribution Linux RedHat (jusqu'à la version 9), équivalents à ceux que RedHat commercialise. La structure de la distribution est décrite, ainsi que la procédure d'inclusion de paquets RPM mis à jour. Il donne quelques conseils et quelques exemples de personnalisation de l'installation par défaut. Ce document contient également des scripts automatisant autant que possible la (re)génération des images CD. Créer vos propres CD ne demandera qu'une bonne connexion réseau et un graveur de CD (une connaissance des scripts shells peut aussi être

utile).

Table des matières

1. Introduction	2
1.1. Avertissement et licence	3
2. Anatomie du site FTP de Red Hat	3
2.1. Organisation des répertoires de la distribution Redhat 9	3
2.2. Le répertoire « RedHat », le cœur de la distribution	4
2.3. Le répertoire « updates »	5
2.4. Différences avec l'arborescence 8.0	6
2.5. Différences avec l'arborescence 7.x	6
2.6. Différences avec l'arborescence de la 6.x	6
3. Paquets RPM	7
3.1. Comparer deux versions d'un paquet RPM	8
4. Obtenir votre copie locale de la distribution	9
4.1. Utiliser wget et bash	9
4.2. Utiliser mirror	10
5. Inclure les mises à jour	11
5.1. Corriger les modes de protection des fichiers	12
5.2. Remplacer les RPM mis à jour	12
5.3. Reconstruire l'installateur	12
6. Graver le ou les CD	21
6.1. Essayer l'image ou les images	21
6.2. Graver le ou des disques	21
7. Le fichier comps	21
7.1. Format du fichier comps pour RedHat versions < 6.1	22
7.2. Format du fichier comps pour RedHat version 6.1	22
7.3. Format du fichier comps dans RedHat version 6.2	24
7.4. Format d'un fichier comps dans la RedHat version 7.3	24
7.5. Format du fichier comps à partir des versions 8.0 et 9 de RedHat	25
8. Installation à partir du CD	27
8.1. Démarrer d'un CD amorçable	27
9. Autres distributions Linux	27
10. Ce document...	28
10.1. Documents connexes	28
10.2. Remerciements	28
11. Adaptation française	29
11.1. Traduction	29
11.2. Relecture	29

1. Introduction

Il existe de nombreuses raisons de créer vos propres CD. Peut-être êtes-vous avare et voulez-vous économiser le prix d'une distribution Red Hat [<http://www.redhat.fr/>]. Ou peut-être souhaitez-vous graver des CD contenant la plus récente distribution et toutes les dernières mises à jour. C'est très pertinent car après chaque version majeure de la distribution RedHat, de nombreuses mises à jour sont publiées, dont un certain nombre relatives à la sécurité. Jetez juste un œil sur la page d'errata [<https://www.redhat.com/security/archives.html>]. Ou peut-être voulez-vous personnaliser l'installation par défaut en ajoutant quelques paquets absents et en retirant certains autres.

Voici ce que vous apprendrez dans les sections suivantes (je l'espère). Les exemples seront basés sur les versions 7.3, 8.0 et 9 de la distribution. Les notes relatives aux versions précédentes (<6.1) proviennent d'une version précédente et ont été rassemblées par les auteurs originaux. Les notes relatives à la version 6.2 sont basées sur des essais que je n'ai pas terminés (et je ne sais pas si je les terminerai) et sur quelques documents que vous trouverez dans la section documents connexes. La procédure donnée dans les sections suivantes pour Redhat 7.3 et 8.0 peut fonctionner sur toutes les

plates-formes compatibles avec cette distribution (Alpha, PPC et cætera), pour toutes les versions 7.x (et peut-être les versions 8.x et 9 dans un avenir pas si lointain) mais je l'ai seulement essayée sur la plate-forme i386 avec le Linux Redhat 7.3, 8.0 et 9 (je serais intéressé par plus d'informations).

Note

Les opérations décrites ont des implications légales, ce qui veut dire que vous ne pouvez pas redistribuer les CD en tant que RedHat Linux si vous les avez modifiés de façon non conforme à la politique de RedHat. Pour les rendre légalement redistribuables, vous devez d'abord appliquer les lignes de conduites indiquées sur le site web de RedHat [<http://www.redhat.com/about/corporate/trademark/>].

Note

Rappelez-vous de toujours mettre en place les variables dans `rhcd.conf` et d'*exporter* la variable d'environnement `RHCDPATH` avant de lancer les scripts que vous trouverez tout au long du reste de ce document et en relation avec les versions supérieures ou égales à la 6.2 de la RedHat Linux. Un fichier `rhcd.conf` [`outils/RedHat-CD-HOWTO/rhcd-scripts/rhcd.conf`] d'exemple, qui devrait être bien commenté, est donné avec les scripts.

1.1. Avertissement et licence

Ce document est distribué dans l'espoir qu'il sera utile, mais SANS AUCUNE GARANTIE ; sans même la garantie implicite de qualité loyale et marchande ou d'exactitude pour un usage particulier.

Ni l'auteur ni les distributeurs, ou tout autre contributeur de ce document, ne sont de quelque façon que ce soit responsables pour les dommages physiques, financiers, moraux ou de tout autre type, occasionnés en suivant les suggestions de ce texte.

2. Anatomie du site FTP de Red Hat

Dans l'esprit de la communauté Linux, la société RedHat a rendu disponible ses distributions Linux pour plusieurs plates-formes sur son site FTP. Elles sont toutes disponibles à partir de la racine du répertoire de distribution (`pub/redhat/linux/` [<ftp://ftp.redhat.com/pub/redhat/linux/>]). Jetez donc un œil sur l'arborescence de la distribution.

2.1. Organisation des répertoires de la distribution Redhat 9

La plus récente distribution n'est, à l'heure où j'écris ces lignes, disponible que pour la plate-forme i386. Le répertoire de premier niveau semble un peu inutile, étant donnée la présence d'une seule architecture (`/pub/redhat/linux/9/en/os/` [<ftp://ftp.redhat.com/pub/redhat/linux/9/en/os/>]).

```
i386/
```

Le répertoire de premier niveau des versions un peu antérieures à la version 9 contient les distributions destinées aux différentes plates-formes. Par exemple, le répertoire correspondant de la version 7.1 du Linux RedHat est structuré comme ceci :

```
alpha/   i386/   ia64/   ppc/    s390x/
```

Pour une distribution Redhat 9, la racine du répertoire i386 ressemble à ce qui suit :

```
-rwxr-xr-x  2 root  root    248 Mar 14  2003  autorun
```

drwxr-xr-x	7	root	root	4096	Mar	14	2003	dosutils
-rw-r--r--	3	root	root	6192	Mar	14	2003	EULA
-rw-r--r--	3	root	root	18385	Mar	14	2003	GPL
drwxr-xr-x	3	root	root	2048	Mar	14	2003	images
drwxr-xr-x	2	root	root	2048	Mar	14	2003	isolinux
-rw-r--r--	3	root	root	6127	Mar	14	2003	README
-rw-r--r--	2	root	root	13052	Mar	14	2003	README-Accessibility
-rw-r--r--	2	root	root	6686	Mar	14	2003	README.de
-rw-r--r--	2	root	root	6990	Mar	14	2003	README.es
-rw-r--r--	2	root	root	6492	Mar	14	2003	README.fr
-rw-r--r--	2	root	root	6805	Mar	14	2003	README.it
-rw-r--r--	2	root	root	7995	Mar	14	2003	README.ja
-rw-r--r--	2	root	root	7312	Mar	14	2003	README.ko
-rw-r--r--	2	root	root	5070	Mar	14	2003	README.pt
-rw-r--r--	2	root	root	6613	Mar	14	2003	README.pt_BR
-rw-r--r--	2	root	root	5879	Mar	14	2003	README.zh_CN
-rw-r--r--	2	root	root	5892	Mar	14	2003	README.zh_TW
drwxr-xr-x	4	root	root	2048	Mar	14	2003	RedHat
-rw-r--r--	2	root	root	25824	Mar	14	2003	RELEASE-NOTES
-rw-r--r--	2	root	root	29902	Mar	14	2003	RELEASE-NOTES-de.html
-rw-r--r--	2	root	root	30409	Mar	14	2003	RELEASE-NOTES-es.html
-rw-r--r--	2	root	root	32354	Mar	14	2003	RELEASE-NOTES-fr.html
-rw-r--r--	2	root	root	30064	Mar	14	2003	RELEASE-NOTES.html
-rw-r--r--	2	root	root	29925	Mar	14	2003	RELEASE-NOTES-it.html
-rw-r--r--	2	root	root	34666	Mar	14	2003	RELEASE-NOTES-ja.html
-rw-r--r--	2	root	root	33520	Mar	14	2003	RELEASE-NOTES-ko.html
-rw-r--r--	2	root	root	29496	Mar	14	2003	RELEASE-NOTES-pt_BR.html
-rw-r--r--	2	root	root	22747	Mar	14	2003	RELEASE-NOTES-pt.html
-rw-r--r--	2	root	root	25217	Mar	14	2003	RELEASE-NOTES-zh_CN.html
-rw-r--r--	2	root	root	26645	Mar	14	2003	RELEASE-NOTES-zh_TW.html
-rw-r--r--	3	root	root	1910	Mar	14	2003	RPM-GPG-KEY
-r--r--r--	1	root	root	1823	Mar	14	2003	TRANS.TBL

Le répertoire SRPMS contient les paquets RPMS en format source.

Le répertoire `images` contient les images des disquettes de démarrage et de pilotes. Ces images peuvent être copiés sur une disquette si nécessaire. Pour la version 9, il n'existe qu'une seule image de disque de démarrage. Cette image de démarrage est appelée `boot.img`. Si l'installation n'est pas exécutée depuis un CD-ROM ou un disque dur, il est nécessaire de préparer également une seconde disquette avec les pilotes. Un fichier `boot.iso` est maintenant également disponible. Il permet de démarrer une machine à partir du lecteur de CD-ROM afin de pouvoir plus facilement lancer une installation réseau (c'est-à-dire sans avoir besoin de manipuler une tonne de disquettes). Reportez-vous à la section installation et aux références qu'elle contient pour plus d'informations. Consultez le fichier `README` contenu dans ce répertoire pour obtenir une explication plus détaillée du rôle des différents fichiers.

Le répertoire `isolinux` contient les fichiers nécessaires au démarrage depuis le CD (et pour reconstruire des CD de démarrage qui fonctionnent de la même façon). Ce processus a été modifié pour passer d'une émulation de disquette à pas d'émulation du tout. Ce qui aide à éviter les contraintes d'espace et les problèmes de compatibilité.

Le répertoire `dosutils` contient différents programmes pour certains autres systèmes d'exploitation, qui sont parfois utiles pour le bon déroulement du processus d'installation. Il contient aussi un fichier `README` explicatif.

La liste est complétée par un grand nombre de fichiers et par le répertoire `RedHat`. Ce dernier est le sujet des sections qui suivent tandis que le contenu des précédents est clairement indiqué par leur nom (sauf peut-être le fichier `EULA` dont le nom est l'abréviation [en anglais] de « Accord de licence avec l'utilisateur final »).

2.2. Le répertoire « RedHat », le cœur de la distribution

La majeure partie de l'arborescence de la distribution se situe dans le répertoire RedHat :

```
drwxr-xr-x  2 root  root  53248 Jun 14 03:15 RPMS
drwxr-xr-x  2 root  root   4096 Jun 14 04:15 base
```

Le répertoire RPMS contient la majeure partie de la distribution RedHat, sous la forme d'un ensemble de fichiers RPM (abréviation de *Redhat Package Manager*, c'est-à-dire « gestionnaire de paquets RedHat »). Un paquet RPM contient en général des exécutables binaires, accompagnés de leurs fichiers de configuration et de leur documentation. Référez-vous à la section les paquets RPM pour plus d'informations.

Le répertoire base contient différents fichiers nécessaires lors de l'installation, comme le fichier `comps.xml`, qui définit les *composants* (groupes de paquets) utilisés durant la phase « Sélection des groupes de paquetages¹ ». Reportez-vous à la section le fichier `comps` pour plus d'informations sur ce fichier et son utilisation.

Le répertoire base contient deux autres fichiers d'importance : `hdlist` et `hdlist2`. Ils contiennent la plupart des informations incluses dans les champs d'en-têtes de l'ensemble des paquets RPM du répertoire RPMS. Il est donc possible de déterminer les interdépendances entre paquets RPM par une simple lecture de ces fichiers, sans avoir à lire chacun des paquets RPM, ce qui est très appréciable notamment lors des installations par FTP. Ces fichiers permettent également de déterminer les fichiers correspondant à un paquet donné (par exemple `perl` renvoie vers le paquet `perl-5.004-6.i386.rpm`). Ce qui veut dire que si vous ajoutez vos propres paquets ou des mises à jour RedHat (reportez-vous à la section inclure les mises à jour) dans le répertoire RPMS, vous aurez besoin de mettre à jour `hdlist` et `hdlist2`. La façon de réaliser cette mise à jour sera décrite dans reconstruire l'installateur. En dehors de ces fichiers, on trouve les images à partir desquelles l'environnement d'installation est lancé (c'est-à-dire le noyau, l'interpréteur python, anaconda, et cætera).

2.3. Le répertoire « updates »

Le répertoire `/pub/redhat/linux/updates` contient des mises à jour destinées à toutes les versions de la distribution RedHat depuis la 3.0.3. C'est l'endroit où vous trouverez les paquets qui ont été mis à jour pour une raison ou une autre. Vous devez tout particulièrement faire attention aux mises à jour de sécurité. Elles sont affichées sur la page des erreurs de RedHat dès qu'une correction est disponible. Les fichiers les plus importants trouvés dans le répertoire `updates` sont :

```
drwxrwsr-x  3 root  root   4096 Jul 13 10:13 5.2
drwxrwsr-x  3 root  root   4096 Jul 13 10:13 6.0
drwxrwsr-x  3 root  root   4096 Jul 13 10:13 6.1
drwxrwsr-x  4 root  root   4096 Jul 13 10:14 6.2
drwxrwsr-x  4 root  root   4096 Jul 13 10:14 7.0
drwxrwsr-x  4 root  root   4096 Jul 13 10:14 7.1
drwxrwsr-x  4 root  root   4096 Jul 13 10:13 7.2
drwxrwsr-x  3 root  root   4096 Jul 13 10:14 7.3
drwxrwsr-x  3 root  root   4096 Jul 13 10:14 8.0
drwxrwsr-x  3 root  root   4096 Jul 13 10:14 9
```

La structure de chacun de ces sous-répertoires est similaire à ce qui est décrit dans la section l'organisation de la Redhat 9. Donc, pour chaque version, vous trouverez dans le sous-répertoire `en/os/` une série de sous-répertoires représentant les nombreuses architectures ainsi que les sous-répertoires `noarch` et `SRPMS`, respectivement pour les paquets qui fonctionnent sur toutes les architectures et pour ceux qui sont sous forme de source.

```
drwxrwsr-x  2 root  root   4096 Sep 23 05:28 SRPMS
```

¹ « Paquetages » est le terme utilisé par RedHat pour parler des « paquets », c'est-à-dire des formats d'archive utilisés pour distribuer des applications et incluant tout ce qui est nécessaire à l'installation, à la désinstallation et au fonctionnement de cette application.

```
drwxrwsr-x 2 root root 4096 Aug 28 18:25 athlon
drwxrwsr-x 2 root root 8192 Sep 23 05:28 i386
drwxrwsr-x 2 root root 4096 Jul 13 10:14 i486
drwxrwsr-x 2 root root 4096 Aug 28 18:26 i586
drwxrwsr-x 2 root root 4096 Aug 28 18:26 i686
drwxrwsr-x 2 root root 4096 Jul 13 10:14 noarch
```

2.4. Différences avec l'arborescence 8.0

La disposition de la distribution 8.0 est pratiquement identique à celle que nous venons de décrire. Les seules différences majeures, à cet égard, se trouvent dans le répertoire `images`.

Le répertoire `images` contient les images de disquettes de démarrage et de pilotes, pouvant être copiées sur une disquette si nécessaire. La première image de démarrage est appelée `boot.img` et est nécessaire lorsque l'installation est exécutée directement depuis le CD-ROM. Si vous envisagez une installation via un disque monté par NFS ou par FTP, l'image disque `bootnet.img` sera nécessaire. Les installations via l'adaptateur PCMCIA nécessitent la disquette `pcmcia.img`. Reportez-vous à la section installation et aux références qui s'y trouvent pour plus d'informations. Consultez le fichier `README` contenu dans ce répertoire pour obtenir une explication plus détaillée du rôle des différents fichiers.

2.5. Différences avec l'arborescence 7.x

Les deux distributions sont pratiquement similaires sous cet aspect. Les seuls changements intéressants pour nous (et faciles à remarquer en regardant l'arborescence de la distribution) sont la disparition du répertoire `isolinux` et quelques modifications affectant le répertoire `RedHat/base`. Le premier changement est dû à la façon dont les CD d'installation sont rendus amorçables dans les versions antérieures à la 8.0 (le mode « émulation disquette » a été changé en mode « pas d'émulation » pour la version 8.0), alors que le second est un effet du passage en XML du fichier `comps` pour les distributions Redhat postérieures à la 8.0 (ce qui explique pourquoi il a été renommé `comps.xml`). Dans les distributions Redhat 7.3 et précédentes, le fichier `Redhat/base/comps` est un simple fichier texte dont la syntaxe n'est pas très souple.

2.6. Différences avec l'arborescence de la 6.x

Pour la version 6.2 (<ftp://ftp.redhat.com/pub/redhat/linux/6.2/en/os/>), la dernière de la série des 6, l'organisation est la suivante (celle des précédentes versions est à peu près similaire, mais pas complètement) :

```
alpha/   i386/   sparc/
```

La racine du répertoire `i386`, quant à elle, ressemble à ceci :

```
-rw-r--r-- 1 root root 18385 Sep 7 1999 COPYING
-rw-r--r-- 1 root root 3400 Mar 8 2000 README
-rw-r--r-- 1 root root 16300 Mar 8 2000 RELEASE-NOTES
-rw-r--r-- 1 root root 1908 Sep 25 1999 RPM-GPG-KEY
drwxr-xr-x 1 root root 512 Sep 27 15:22 RedHat
drwxr-xr-x 1 root root 17408 Sep 27 15:22 SRPMS
-rwxr-xr-x 1 root root 538 Sep 26 1999 autorun
-rwxr--r-- 1 root root 2048 Mar 9 2000 boot.cat
drwxr-xr-x 1 root root 512 Sep 27 15:22 doc
drwxr-xr-x 1 root root 512 Sep 27 15:22 dosutils
drwxr-xr-x 1 root root 512 Sep 27 15:22 images
drwxr-xr-x 1 root root 512 Sep 27 15:22 misc
```

Dans les paragraphes suivants, j'indiquerai uniquement les différences avec les versions plus récentes ; ce qui ne sera pas explicitement mentionné ici est resté (ou est supposé être resté) inchangé.

Le répertoire `doc` contient une foule d'informations. En premier lieu, ce répertoire contient le manuel d'installation RedHat au format HTML (qui est aussi disponible sur le site de RedHat : Guide d'installation Redhat 6.2 [<http://www.europe.redhat.com/documentation/rhl6.2/install-guide-fr/>]). Il contient également le Guide de référence et le Guide de démarrage (*Getting started*). La documentation des versions 7.x, 8.0 et 9 est distribuée sur un CD séparé (dans une arborescence différente sur le site ftp).

Le répertoire `images` contient les images de disquettes de démarrage. Si nécessaire, ces images peuvent être recopiées sur disquette, comme pour les distributions Redhat 9, 8.0 et 7.3. Référez-vous à la section installation et aux références qu'elle contient pour plus d'information. Le répertoire `misc` contient les sources et les exécutables d'un certain nombre de programmes nécessaires à l'installation.

La plus importante partie de l'arborescence est (encore) située sous le répertoire RedHat :

drwxr-xr-x	2	root	root	28672	Oct 26 09:01	RPMS
drwxr-xr-x	2	root	root	4096	Oct 26 09:01	base
-rw-r--r--	1	root	root	0	Jan 19 1999	i386
drwxr-xr-x	6	root	root	4096	Oct 26 09:01	instimage

Vous devez déjà connaître Le répertoire `RPMS`. Référez-vous à la section les paquets RPM pour plus d'informations. Le répertoire `base` contient les différents fichiers nécessaires à l'installation, comme pour les Redhat 7.3, 8.0 et 9. Les seules différences visibles sont la présence d'un unique fichier `hdlist` et l'absence du fichier `stage2.img` dont les fonctionnalités devraient être assurées par les fichiers contenus dans le répertoire `instimage`. Ce répertoire contient, en fait, un vrai système de fichiers limité à l'essentiel et comportant les programmes et bibliothèques partagées nécessaires à l'installation.

Le répertoire `updates` est en fait similaire à celui qui est décrit pour la version 9, la seule différence étant qu'il comporte davantage de répertoires relatifs aux différentes architectures.

3. Paquets RPM

La majeure partie de la distribution RedHat consiste en un ensemble de fichiers RPM ((abréviation de *Redhat Package Manager*, c'est-à-dire « gestionnaire de paquets RedHat »). Un paquet RPM contient en général des exécutables binaires, accompagnés de leurs fichiers de configuration et de leur documentation. Le programme `rpm` [<http://www.rpm.org>] est un puissant gestionnaire de paquets, qui peut être utilisé pour installer, interroger, vérifier, mettre à jour, effacer et construire des paquets au format RPM. `Rpm` est très pratique car il gère une base de données de tous les paquets installés, ce qui permet de savoir à tout moment ce qui est installé.

Les fichiers binaires RPM inclus dans la distribution ont été construits sur un système utilisant lui-même la distribution. C'est important car la plupart des programmes des paquets dépendent de bibliothèques partagées. La nouvelle version 3 de la bibliothèque C standard GNU (compatible 64 bits) a été utilisée à partir de la distribution Redhat 5.0. Cette version de la bibliothèque est communément appelée *glibc* ou, sous Linux, *libc 6*. Tous les exécutables de cette distribution ont été liés à cette bibliothèque. Si vous tentez d'installer les fichiers binaires d'une distribution différente, il y a beaucoup de chances que cela ne fonctionne pas, sauf si vous installez le paquet `libc5` pour obtenir une compatibilité descendante. Il existe aussi des incompatibilités entre les nombreuses versions du RedHat Package Manager lui-même qui empêcheront l'installation de certains paquets même sur les machines où ils sont supposés fonctionner.

Les noms des paquets RPM contiennent le suffixe `.archi.rpm`, où *archi* est l'architecture. Celle-ci a habituellement pour valeur `i386` pour les binaires destinés à la plate-forme Intel. Les paquets que vous installez doivent correspondre aux versions des bibliothèques partagées installées sur la machine. Le programme `rpm` [<http://www.rpm.org>] est habituellement assez bon pour s'en assurer. Néanmoins, il existe des moyens de passer outre cette vérification. Si vous décidez de forcer l'installation d'un paquet de cette façon, soyez *vraiment* sûr de savoir ce que vous faites. Néanmoins, l'utilisation du disque de démarrage d'installation de RedHat vous garantit qu'un ensemble correct de paquets RPM sera installé sur la machine.

Si vous découvrez un paquet RPM qui n'a pas été installé sur votre système durant le processus d'installation, ne désespérez pas. À tout moment, vous pourrez (sous le compte root) installer des paquets RPM. Par exemple :

```
# rpm --install WindowMaker-0.18-1b.i386.rpm
```

Vous pouvez même installer un paquet RPM directement depuis Internet, si vous connaissez son URL :

```
# URL="ftp://rufus.w3.org/redhat-contrib/noarch/mirror-2.9-2.noarch.rpm"
# rpm --install "$URL"
```

Si vous voulez mettre à jour un paquet RPM (ou l'installer s'il n'est pas présent sur la machine), utilisez la commande :

```
# rpm --update WindowMaker-0.18-1b.i386.rpm
```

Si vous voulez mettre à jour un paquet RPM dont une version précédente est déjà installée, utilisez la commande :

```
# rpm --freshen WindowMaker-0.18-1b.i386.rpm
```

Il existe un autre type de paquets RPM qui contient les sources originales qui ont servi à construire les binaires. Ces paquets ont le suffixe `.src.rpm` et sont situés dans le répertoire SRPMS. La moitié du troisième CD et les deux derniers des cinq CD de la distribution Redhat 8.0 (ou la 7.3) contiennent des paquets source. Pour la 9, ils sont sur trois CD séparés. Pour la 6.2 (et les précédentes versions, pas trop anciennes), les choses changent un peu puisqu'il n'existe qu'un seul CD d'installation qui ne comporte pas les paquets SRPMS, que vous pouvez graver sur un disque différent si vous le voulez.

Pour obtenir plus d'informations sur le gestionnaire de paquet RedHat, je vous suggère de lire les pages de manuel et le très complet Maximum rpm [<http://www.rpm.org/max-rpm/>].

Dans la prochaine section, je présenterai un programme C qui sera utilisé dans divers scripts tout au long du reste de ce guide pratique. Il indique, entre deux versions du même paquet RPM, celui qui est le plus récent. Ce programme est basé sur le code utilisé dans le gestionnaire de paquets RedHat (version 4.1) et est utilisé quand l'option `--freshen` est ajoutée.

3.1. Comparer deux versions d'un paquet RPM

Le code C des trois fichiers Makefile [outils/RedHat-CD-HOWTO/rhcd-scripts/rpmvc/Makefile], rvc.h [outils/RedHat-CD-HOWTO/rhcd-scripts/rpmvc/rvc.h], rvc.c [outils/RedHat-CD-HOWTO/rhcd-scripts/rpmvc/rvc.c] a été extrait du code du gestionnaire de paquets RedHat et (légèrement) modifié pour répondre à nos besoins. Ils forment un programme C simple qui, avec deux versions A et B d'un paquet retourne 1, 0 ou -1 si A est respectivement plus récent, égal ou plus ancien que B et d'autres valeurs en cas d'erreur (vous pouvez lire les commentaires du code pour plus d'informations). Pour compiler le programme, vous aurez besoin du programme make et du compilateur C gcc. Copiez les fichiers dans le même répertoire et lancez la commande :

```
$ make
```

Ce programme est nécessaire à pratiquement tous les scripts utilisés dans les sections suivantes. Pour que les scripts puissent le retrouver, vous devez définir la variable RVC dans le fichier

rhcd.conf [outils/RedHat-CD-HOWTO/rhcd-scripts/rhcd.conf].

Vous trouverez une copie des sources et de la version précompilée dans l'archive rhcd-scripts.tar.gz [outils/RedHat-CD-HOWTO/rhcd-scripts.tar.gz] située dans le répertoire rpmvc.

Note

Ce programme était utilisé de façon incorrecte par les scripts *updateDist.sh* (ver. < 1.17) et *updateCD.sh* (ver. < 1.12). Je vous suggère fortement d'éviter les versions antérieures de ces scripts antérieures, même si ce problème n'est pas fréquent (du moins en apparence).

4. Obtenir votre copie locale de la distribution

Vous aurez besoin d'une copie de la distribution sur un disque où vous pouvez écrire et accessible à partir de l'ordinateur possédant le graveur (ouah !). Si vous souhaitez incorporer les dernières mises à jour, ce répertoire devra (aussi) être accessible à partir d'une machine Linux, soit à partir d'un disque local, soit à partir du disque distant monté via NFS, soit à partir d'un disque JAZ. Vous pouvez copier la distribution à partir des CD de RedHat (recommandé) ou vous pouvez l'obtenir par FTP. Si vous choisissez d'utiliser FTP, il existe deux moyens de le faire. Vous pouvez utiliser un script shell basé sur *wget*, script présenté dans la section suivante ou utiliser le paquet *mirror* comme suggéré jusqu'à la version 1.34 de ce guide pratique (consultez la section utiliser mirror).

4.1. Utiliser wget et bash

Ce n'est pas la plus simple des méthodes, même si elle est probablement la plus exacte. Je l'apprécie parce qu'elle permet de comparer les versions RPM des fichiers et non plus leur date et heure ou leur nom (comme les paquets classiques de synchronisation à distance) et qu'elle vérifie les signatures des mises à jour à chaque fois qu'elle en télécharge, si la variable `CHECKSIG` du fichier rhcd.conf [outils/RedHat-CD-HOWTO/rhcd-scripts/rhcd.conf] lui indique de le faire.

Créez un répertoire qui contiendra les fichiers d'installation et placez-vous à l'intérieur, puis lancez la commande suivante, qui téléchargera environ 3 Go de données sur votre disque dur :

```
$ wget -r -c -t0 -l0 --retr-symlinks -nH --cut-dirs=9 \
ftp://ftp.mirror.ac.uk/sites/ftp.redhat.com/pub/redhat/linux/updates/7.3/en/os/
```

Vous voudrez probablement changer le miroir FTP de téléchargement et, en conséquence, le paramètre indiqué à l'option `--cut-dirs`. Ce paramètre est utilisé conjointement avec l'option `-nH` pour éviter la re-création de la hiérarchie des répertoires du site FTP. Pour plus d'informations sur l'utilisation correcte de cette option, jetez un œil sur la documentation de *wget* [<http://www.gnu.org/manual/wget-1.8.1/wget.html>] et les pages de manuel correspondantes.

Si vous voulez exclure un ou plusieurs répertoires du téléchargement, vous pouvez utiliser l'option `-X liste`, où *liste* représente une liste de répertoires séparés par des virgules. Par exemple, pour exclure le répertoire SRPMS du précédent téléchargement, vous pouvez utiliser :

```
$ wget -r -c -t0 -l0 --retr-symlinks -nH --cut-dirs=9 \
-X /sites/ftp.redhat.com/pub/redhat/linux/updates/7.3/en/os/i386/SRPMS \
ftp://ftp.mirror.ac.uk/sites/ftp.redhat.com/pub/redhat/linux/updates/7.3/en/os/
```

Cela peut être utile si vous prenez en compte la taille du répertoire SRPMS (environ 1,2 Go) ; en tout cas, je le trouve utile.

Si vous voulez vérifier les signatures GPG pour vous assurer de l'authenticité des paquets (ce qui est quelque chose que je vous suggère), vous devrez installer le paquet *gnupg* (nécessaire uniquement pour la RedHat 7.3) et importer la clé publique *security@redhat.com* que vous trouverez dans le répertoire racine des CD (RPM-GPG-KEY) ou sur le site web RedHat [<http://www.redhat.com/solutions/security/news/publickey.html#key>]. La clé est importée en lançant la commande : `gpg --import nom_du_fichier` pour les versions jusqu'à la 7.3 in-

cluse, ce qui a été remplacé par `rpm --import nom_du_fichier` pour les versions 8.0 et 9 (pour plus d'informations sur ceci, jetez un œil aux sites web de GNU Privacy Guard [<http://www.gnupg.org/>] et de RPM [<http://www.rpm.org/>] — le Gestionnaire de paquets RedHat).

Si vous voulez vérifier les paquets RPM, vous pouvez le faire en utilisant la commande suivante (que je suppose lancée depuis le répertoire où vous avez réalisé les téléchargements) :

Pour les versions jusqu'à la 7.3 incluse :

```
$ find . -name "*.rpm" -exec rpm -K --nopgp {} \; |grep "NOT *OK"
```

Pour les versions 8.0 et 9 (ainsi que pour les versions futures, je suppose) :

```
$ find . -name "*.rpm" -exec rpm -K {} \; |grep "NOT *OK"
```

Si vous ne voulez pas vous « ennuyer » avec toutes ces étapes, j'espère que vous voudrez au moins vérifier l'intégrité des fichiers téléchargés (ce qui ne veut pas dire que personne ne les a modifiés), à l'aide des signatures md5. Ceci est fait avec :

Pour les versions jusqu'à 7.3 (incluse) :

```
$ find . -name "*.rpm" -exec rpm -K --nopgp --nogpg {} \; |grep "NOT *OK"
```

Pour les versions 8.0 et 9 (ainsi que pour les versions à venir, je suppose) :

```
$ find . -name "*.rpm" -exec rpm -K --nosignature {} \; |grep "NOT *OK"
```

Le contenu d'une distribution RedHat ne change pas entre les versions, donc vous aurez seulement besoin de télécharger ces paquets *UNE FOIS*. Toutes les modifications de la distribution sont contenues dans le répertoire `updates`. Donc, si vous voulez conserver un miroir à jour de la distribution RedHat, vous aurez seulement besoin de maintenir le répertoire `updates` à jour. Ceci se fait en utilisant le script `updateDist.sh` [outils/RedHat-CD-HOWTO/rhcd-scripts/updateDist.sh]. Avant d'utiliser ce script, vous devrez configurer le fichier `rhcd.conf` [outils/RedHat-CD-HOWTO/rhcd-scripts/rhcd.conf] et exporter la variable `RHCDPATH` pointant vers le répertoire où se trouve ce fichier.

```
$ export RHCDPATH=/home/luigi/tmp/rhcd-scripts
$ sh updateDist.sh
```

Ce script va télécharger les nouvelles mises à jour en excluant les sous-répertoires indiqués dans la variable `EXCLUDELIST`, en déplaçant les plus anciennes (c'est-à-dire celles remplacées par de nouvelles versions) dans le répertoire indiqué par la variable `OLDDIR` après avoir réussi deux tests. Le premier test compare les fichiers `.listing` générés par `wget` avec le contenu des répertoires locaux pour s'assurer que tous les fichiers ont été téléchargés. Le second test vérifie les signatures des paquets en fonction des valeurs de deux variables `CHECKSIG` et `USEGPG` (positionnez-les à « `yes` » si vous souhaitez que cette opération soit réalisée). En cas d'échec du processus de vérification de signature, le script déplacera le mauvais paquet dans `OLDDIR` en lui affectant l'extension « `.UPDcheckfail` » et abandonnera sans déplacer les anciennes mises à jour dans `OLDDIR`.

4.2. Utiliser mirror

Mirror est un script perl sophistiqué comparant le contenu d'un répertoire d'un site distant avec un répertoire local. Il utilisera FTP pour récupérer les fichiers qui sont sur le site distant mais pas sur le site local et supprimera sur le site local les fichiers qui ne sont pas sur le site distant. Le programme mirror est paramétré via un fichier de configuration. Le RPM du paquet est disponible à partir de ru-

fus.w3.org [<http://rufus.w3.org/linux/RPM/mirror.html>]. Créez une copie locale `mirror.redhat` du fichier de configuration de `mirror` et modifiez les champs appropriés en haut du fichier. Après la section des valeurs par défaut (*default*), définissez les paquets suivants :

```
package=updates
site=ftp.mirror.ac.uk
exclude_patt=(SRPMS/)
remote_dir=/sites/ftp.redhat.com/pub/redhat/linux/updates/7.3/en/os/i386
local_dir=/home/luigi/tmp/redhat-cd/redhat-7.3-updates

package=dist
site=ftp.mirror.ac.uk
exclude_patt=(SRPMS/)
remote_dir=/sites/ftp.redhat.com/pub/redhat/linux/7.3/en/os/i386
local_dir=/home/luigi/tmp/redhat-cd/redhat-7.3
```

La commande suivante va copier l'arborescence RedHat au complet sur votre disque local. ****Pensez****, avant de faire cela, que vous allez télécharger à peu près 1,5 Go de données (si vous avez exclu le répertoire SRPMS) !

```
$ mirror -pdist mirror.redhat
```

Ceci va créer une copie à l'identique du site FTP de RedHat sur votre disque local. Le contenu de la distribution RedHat ne change pas entre les versions, donc vous aurez seulement besoin de télécharger ces paquets *une fois*. Tout changement dans la distribution se trouvera dans le répertoire `updates`. Donc, si vous voulez maintenir votre miroir à jour, vous aurez seulement besoin d'actualiser le répertoire `updates`. Cela se fait avec la commande suivante :

```
$ mirror -pupdates mirror.redhat
```

Vous pouvez la lancer régulièrement, disons une fois par semaine, avec un script cron. La distribution RedHat est disponible sur un grand nombre de serveurs FTP dans le monde entier, mis à jour quotidiennement à partir du site de référence `ftp.redhat.com` [<ftp://ftp.redhat.com/pub>]. Vous devriez choisir un site FTP proche de vous, en consultant la liste des sites miroirs RedHat [<http://www.redhat.com/download/mirror.html>].

Note

Je n'ai pas testé personnellement cette procédure. C'était la seule procédure proposée sur les anciennes versions de ce guide pratique (jusqu'à la version 1.34, concernant RedHat < 6.1).

5. Inclure les mises à jour

Il y a trois étapes, les deux premières étant (pratiquement) identiques pour toutes les versions, alors que la dernière change un peu en raison des modifications de l'installateur `anaconda` :

- i. Corriger les modes de protection des fichiers
- ii. Remplacer les RPM mis à jour
- iii. Reconstruire l'installateur

Pour incorporer les mises à jour, vous aurez besoin d'avoir un accès en écriture au répertoire de la distribution à partir d'une machine Linux, avec une version fonctionnelle de `rpm` [<http://www.rpm.org>] installée, alors que *pour reconstruire l'installateur anaconda, vous aurez besoin d'utiliser une version du Linux RedHat égale à celle pour laquelle vous reconstruisez l'installateur (sinon la procédure échouera)*. Si vous maintenez à jour un miroir du répertoire `up-`

dates, vous pourrez à tout moment produire un CD incluant les dernières mises à jours en répétant ces étapes.

5.1. Corriger les modes de protection des fichiers

Durant le processus d'installation des versions jusqu'à la 6.2 incluse, certains programmes sont lancés directement depuis le CD. Malheureusement, le programme FTP ne préserve pas toujours les droits des fichiers et des répertoires copiés. Donc, il est nécessaire de s'assurer que les droits d'exécution sont bien donnés aux programmes, scripts shells et bibliothèques partagées, avant que le répertoire ne soit gravé sur le CD. Ceci est fait en lançant le script `updatePerm.sh` [outils/RedHat-CD-HOWTO/rhcd-scripts/updatePerm.sh] sur votre copie locale de la distribution. C'est réellement nécessaire pour les versions 6.2 et précédentes. La seule partie utile de cette procédure pour les versions 7.3, 8.0 et 9 est la mise à jour des droits des répertoires, le reste ne fera aucun mal et cela maintient les choses cohérentes. Ce script est quasi-identique au script `updatePerm` inclus dans la version précédente de ce guide pratique, seuls quelques changements mineurs ont été réalisés. Avant d'utiliser ce script, vous devez paramétrer le fichier `rhcd.conf` [outils/RedHat-CD-HOWTO/rhcd-scripts/rhcd.conf] et exporter la variable `RHCDPATH` pointant vers le répertoire où se trouve ce fichier.

```
$ export RHCDPATH=/home/luigi/tmp/rhcd-scripts
$ sh updatePerm.sh
```

5.2. Remplacer les RPM mis à jour

Le script `updateCD.sh` [outils/RedHat-CD-HOWTO/rhcd-scripts/updateCD.sh] copie tous les nouveaux fichiers du répertoire `update` vers le répertoire `RPMS` (et `SRPMS`). Le script utilise le programme `rvc` qui a été présenté dans la section comparer les versions des RPM pour déterminer quels sont les paquets du répertoire `update` qui sont plus récents. Les paquets plus anciens sont déplacés dans le répertoire `{OLDDIR}`. Si la variable `CHECKSIG` est positionnée à « `yes` », tous les paquets de l'arborescence principale verront leur signature vérifiée. Si la vérification de signature d'un paquet échoue (le type de vérification dépend de la variable `USEGPG` du fichier `rhcd.conf` [outils/RedHat-CD-HOWTO/rhcd-scripts/rhcd.conf]), celui-ci est déplacé dans le répertoire `OLDDIR` avec une extension ajoutée, « `CDcheckfail` ».

Avant d'utiliser ce script, vous devrez paramétrer le fichier de configuration `rhcd.conf` [outils/RedHat-CD-HOWTO/rhcd-scripts/rhcd.conf] et exporter la variable `RHCDPATH` pointant vers le répertoire où se trouve ce fichier.

```
$ export RHCDPATH=/home/luigi/tmp/rhcd-scripts
$ sh updateCD.sh
```

Note

Après avoir incorporé les mises à jour dans le répertoire principal RedHat /RPMS, votre copie de la distribution n'est plus un miroir du site de la distribution RedHat. En fait, elle est plus à jour ! Donc, si vous essayez de resynchroniser votre distribution en utilisant `mirror`, les anciennes versions des paquets RPM qui ont été mis à jour seront téléchargées une nouvelle fois et les mises à jour supprimées. La procédure basée sur `bash` et `wget` ne souffre pas de ce problème, mais laissera l'arborescence principale dans un état incohérent. Les anciens et les nouveaux paquets seront dans ce cas mélangés ensemble. Cependant, vous pourrez les trouver et les supprimer en intégrant l'exécutable `rvc` à un script shell simple (que je laisserai comme exercice pour le lecteur...).

5.3. Reconstruire l'installateur

Les choses ont bien changé dans cette section avec l'arrivée de l'installateur `anaconda` (version 6.1) et la considérable augmentation de taille (et... du nombre de CD) que les distributions 7.x et 8.0 ont connue. Jusqu'à la version 6.2, la seule étape composant cette section était la génération d'un nou-

veau fichier `hdlist`. Avec la version 6.2, cela ne reste vrai que jusqu'à un certain point, en raison des changements dans l'installateur `anaconda`, dans le logiciel `rpm` lui-même (qui est passé des versions 3.x aux versions 4.x) et de la migration des paquets mis à jour vers cette nouvelle version (les mises à jour de la version 6.2 sont en fait empaquetées pour les deux versions majeures du logiciel `rpm`). Nous envisagerons trois procédures différentes en essayant de couvrir toutes les versions.

5.3.1. RedHat # 6.1

5.3.1.1. Régénérer le fichier `hdlist`

Lors de l'installation à partir du CD, le programme d'installation du CD se base sur le fichier `RedHat/base/hdlist` qui décrit quels sont les paquets RPM disponibles sur le CD. Le fichier `hdlist` peut être généré par le programme `misc/src/install/genhdlist`. Ce programme doit être lancé avec comme seul argument le chemin absolu vers la racine de la distribution. Voici le script `updateHdlist` qui appelle ce programme (depuis la version 1.34 de ce guide pratique) :

```
#!/bin/bash
RHVERSION=6.1
ARCH=i386

echo "Génération de hdlist..."
RACINERH=/home/luigi/tmp/redhat-${RHVERSION}
GENHDDIR=${RACINERH}/${ARCH}/misc/src/anaconda/utils

chmod u+x ${GENHDDIR}/genhdlist
chmod 644 ${RACINERH}/${ARCH}/RedHat/base/hdlist
${GENHDDIR}/genhdlist ${RACINERH}/${ARCH} || echo "*** ÉCHEC DE GENHDLIST ***"

exit 0
```

Note importante pour les RedHat < 6.1

L'installation de la Redhat 6.1 est complètement différente de celle des versions précédentes car RedHat a introduit `anaconda`. Le programme `genhdlist` est maintenant situé à un autre endroit, donc dans le script ci-dessus, nous utiliserons :

```
GENHDDIR=${RHROOT}/${ARCH}/misc/src/anaconda/utils
```

alors que pour les versions jusqu'à 6.0 (comprise), cette ligne doit être :

```
GENHDDIR=${RHROOT}/${ARCH}/misc/src/install
```

Dans certains cas, `genhdlist` n'arrive pas à démarrer, car l'exécutable n'est pas lié statiquement. Dans un tel cas, vous pouvez ajouter la ligne :

```
${RHROOT}/${ARCH}/RedHat/instimage/usr/lib
```

dans le fichier `/etc/ld.so.conf` et lancer la commande `ldconfig -v`.

Une autre solution est de recompiler `genhdlist`. La modification suivante du script `updateHdlist` fonctionnait sous Redhat 5.2 :

```
#!/bin/bash
RHVERSION=6.1
```

```

ARCH=i386

RACINERH=/misc/redhat/redhat-${RHVERSION}
GENHDDIR=${RACINERH}/${ARCH}/misc/src/anaconda/utils

echo "Compilation de hdlist..."
sed -e 's/FD_t/int/' \
    -e 's/fdOpen/open/' \
    -e 's/fdClose/close/' \
    -e 's/fdFileno//' < ${GENHDDIR}/genhdlist.c > /tmp/genhdlist.c
cc -o /tmp/genhdlist -I/usr/include/rpm /tmp/genhdlist.c -lrpm -lz

echo "Génération de hdlist..."
chmod 644 ${RACINERH}/${ARCH}/RedHat/base/hdlist
/tmp/genhdlist ${RACINERH}/${ARCH} || echo "*** ÉCHEC DE GENHDLIST ***"

exit 0

```

Dans cette version du script, une copie du source C de `genhdlist.c`, compilable sous RedHat 5.2, est créée dans `/tmp`. Cette copie est réalisée en filtrant le source C original avec `sed` via un tube. Cette version de `genhdlist` est alors utilisée pour créer le fichier `hdlist`.

Note importante pour la Redhat 5.2

Tel qu'il est distribué avec les RedHat versions 5.2 et précédentes, `genhdlist` PLANTE si le répertoire `RedHat/RPMS` contient des fichiers qui ne sont *pas* des fichiers RPM ! Cela pose problème car dans la distribution 5.2, il y a des fichiers non-RPM nommés `ls-lR` et `ls-lR.gz` situés dans `RedHat/RPMS`. Donc, vous devez supprimer tous les fichiers non-RPM de ce répertoire. Sinon, vous pouvez appliquer le correctif `genhdlist.c.diff` [`outils/RedHat-CD-HOWTO/rhcd-scripts/oldversion/genhdlist.c.diff`] au fichier `misc/src/install/genhdlist.c` et faire un `make`. Ce correctif fait ignorer à `genhdlist` tout fichier non-RPM.

5.3.1.2. Créer l'image iso du CD

Vous aurez besoin de créer un fichier image qui sera écrit sur le CD. Ce fichier fera au moins 500 Mo, donc trouvez une partition avec assez d'espace libre. Vous aurez peut-être besoin d'utiliser le compte root pour lancer les commandes `mount` et `cdrecord`. Ici, vous allez préparer l'image iso du CD amorçable que vous allez graver. Il n'est pas strictement nécessaire de créer un CD amorçable, car vous pourriez utiliser une disquette de démarrage à la place, mais c'est une fonctionnalité vraiment sympathique (et elle rend le comportement de votre disque plus similaire à celui du disque original). Voici les commandes que j'utilise pour réaliser cette tâche :

```

$ mkdir /repertoire-destination-images
$ mkisofs -r -J -T -v -V "Red Hat 6.1 (Hedwig)" \
  -c boot.cat -b images/boot.img \
  -o /repertoire-destination-images/i386-disc.iso .

```

C'est nécessaire pour graver le disque (amorçable) et cela doit être lancé à partir du répertoire de niveau supérieur de la distribution. Le répertoire `/repertoire-destination-images` est le réceptacle de l'image iso que vous allez générer et il doit (obligatoirement) exister avant de lancer la procédure. Dans la table suivante, vous pourrez lire une brève explication des nombreuses options et de leur signification (la plupart ont été extraites de la page de manuel de `mkisofs`).

Tableau 1. Options et paramètres de `mkisofs`

-r	Extensions Rock Ridge, indiquant des valeurs utiles pour les droits.
-J	Extensions Joliet pour utiliser le CD sous quelques autres sys-

	tèmes d'exploitation.
-T	Génère un fichier TRANS.TBL dans chaque répertoire afin que, même sur des systèmes non compatibles avec les extensions Rock Ridge, la correspondance des noms de fichier soit correcte.
-v	Mode bavard
-V <i>identifiant</i>	Indique l'identifiant du volume (nom ou label du volume) qui devra être écrit dans le bloc maître.
-c <i>cata- logue_de_démarrage</i>	Indique le chemin et le nom du fichier contenant le catalogue de démarrage, qui sera utilisé lors de la création d'un CD amorçable « El Torito ». Le chemin doit être relatif au chemin source indiqué à mkisofs.
-b <i>image_eltorito</i>	Indique le chemin et le nom du fichier contenant l'image de démarrage à utiliser lors de la création du CD amorçable « El Torito ». Le chemin doit être relatif au chemin source indiqué à mkisofs. L'image doit être une image de disquette (ce qui explique pourquoi nous utilisons une des images de disquette trouvées sur le CD original). Vous pourrez choisir d'utiliser l'image <code>pcmcia.img</code> à la place si vous voulez réaliser une installation en utilisant des périphériques PCMCIA tels que des cartes réseau ou des lecteurs CDROM.
-o <i>nom_du_fichier</i>	Nom du fichier contenant l'image iso générée
.	Ceci est le répertoire racine de notre image iso (nous travaillons à partir du répertoire racine de chaque CD, donc un point est suffisant).

Vous trouverez plus d'informations sur la façon de graver une image sur un support dans graver le CD. Les étapes `mkisofs` et `cdrecord` peuvent être exécutées en utilisant une application graphique comme X-CD-Roast [<http://www.xcdrast.org/>] qui devrait actuellement permettre la création de CD amorçables (je ne l'ai jamais utilisé, donc ne vous attendez pas à ce que je vous donne des explications).

5.3.2. Redhat 6.2

Apparemment, c'est l'enfant difficile des distributions RedHat lorsque vient le moment de graver un CD à jour. L'arrivée de la version 4 du gestionnaire de paquets RedHat (RPM) a cassé la procédure de mise à jour de l'installateur anaconda. Les procédures que j'indique fonctionneront uniquement si les paquets mis à jour sont construits en utilisant une version du logiciel RPM postérieure ou égale à la version 3.0.4 (donc, en pratique, la version 3.0.4 ou 3.0.5).

Si vous utilisez les paquets originaux de RedHat, il faudra éviter d'utiliser les mises à jour publiées après le 28 mars 2001 (ce qui est un peu inutile selon moi). Une autre solution est de reconstruire les paquets en utilisant l'ancien format rpm. Vous trouverez des informations sur cette procédure et les outils nécessaires sur la page « rpmhack » [<http://www.tigress.co.uk/rmy/rh62/rpmhack.html>]. Je n'ai pas personnellement essayé cette procédure, mais, d'après les listes de discussion anaconda-devel et kickstart², elle semble fonctionner.

Si vous décidez de rester sur les anciens paquets originaux et de compléter la mise à jour (en utilisant les paquets rpm 4.0.2 après la fin de l'installation), il existe deux façons de le faire, en fonction du type de mise à jour que vous souhaitez effectuer. Si certaines des mises à jour dépendent directement du processus d'installation (c'est-à-dire le noyau, python, kudzu), vous devrez utiliser la procédure de reconstruction de l'installateur expliquée dans le document Construire un CDROM Red Hat Linux 6.2 [<http://web.archive.org/web/20030207104819/http://www.scyld.com/~pzb/rhcd.html>]. Sinon vous pourrez utiliser l'ancienne procédure (celle pour les versions précédentes jusqu'à la 6.1 incluse, telle qu'elle est expliquée dans la section précédente). Les deux dernières étapes, qui sont la création de l'image iso et la gravure du support, sont décrites respectivement dans créer des images iso et graver le CD.

² Vous pourrez trouver ces listes dans la section des listes de discussion [<https://listman.redhat.com/mailman/listinfo>] du site web de RedHat.

5.3.3. Redhat 9, 8.0 et 7.3

Une fois encore, beaucoup de choses ont été changées avec la sortie de la série 7.x. Il faut maintenant réaliser un plus grand nombre d'opérations pour obtenir une série de CD frais et à jour. En réalité, le CD a cessé d'être unique avec la version 7.0. L'arborescence doit maintenant être divisé pour tenir sur le support. Ce qui est fait en utilisant le script **splitdistro**, qui est écrit en python comme beaucoup d'éléments de l'installateur anaconda. Pour terminer cette partie, vous *devrez* utiliser une machine Linux utilisant la distribution Redhat 7.3, 8.0 ou 9 sur laquelle soit installé le paquet *anaconda-runtime* (il aura probablement la version 7.3.7, 8.0.4 ou 9.0.4), en fonction de la version que vous voulez reconstruire. La procédure est composée de sept étapes :

- i. Régénérer les fichiers `hdlist` et `hdlist2`
- ii. Mettre à jour le fichier `comps.xml` (ou `comps`)
- iii. Reconstruire l'installateur
- iv. Diviser la distribution en plusieurs parties de la taille d'un CD
- v. Régénérer (encore) les fichiers `hdlist` et `hdlist2`
- vi. Générer les images iso
- vii. Ajouter et vérifier les signatures md5 dans les images iso

Toutes ces étapes sont regroupées en un seul script qui sera présenté dans la section « le script `updateBuild.sh` ».

5.3.3.1. Opérations préliminaires sur l'arborescence principale

Quelques-uns des scripts inclus dans le paquet *anaconda-runtime* ont besoin de l'arborescence principale, qui doit être déplacée dans un sous-répertoire nommé comme l'architecture que nous allons construire (donc `i386` / chez moi). Nous déplacerons tout vers ce répertoire avant de lancer la procédure. Nous corrigerons également les appels des scripts qui n'ont pas besoin de cette modification.

Pour la Redhat 9 et 8.0 :

```
$ chmod -R u+w /chemin-absolu-de-la-racine
$ mkdir -p /chemin-absolu-de-la-racine/i386
$ cd /chemin-absolu-de-la-racine
$ /bin/mv * i386
```

Vous devrez remplacer `/chemin-absolu-de-la-racine` par le *chemin absolu* du répertoire où la racine de votre copie locale de la distribution est placée (peut-être quelque part sur l'un des disques durs). Vous obtiendrez une erreur, lors de l'exécution de la dernière commande, car le répertoire `i386/` ne peut être déplacé sous lui-même, mais vous ne devez pas en tenir compte.

Pour Redhat 7.3 :

```
$ chmod -R u+w /chemin-absolu-de-la-racine
$ mkdir -p /chemin-absolu-de-la-racine/i386
$ cd /chemin-absolu-de-la-racine
$ for i in `ls` ; do [ $i != "SRPMS" -a $i != i386 ] && \
/bin/mv $i i386 ; done
```

Cette fois-ci (je l'espère) la dernière commande ne devrait produire aucune erreur.

5.3.3.2. Régénérer les fichiers `hdlist` et `hdlist2`

Ceci est fait au moyen des deux commandes suivantes et avec l'aide du programme **genhdlst**.

```
$ /usr/lib/anaconda-runtime/genhdlst
/chemin-absolu-de-la-racine/i386
$ chmod 644
/chemin-absolu-de-la-racine/i386/RedHat/base/hdlist{,
2}
```

Une fois encore */chemin-absolu-de-la-racine* est le *chemin absolu* du répertoire où la racine de votre copie locale de la distribution est placée. La seconde commande est nécessaire pour vous assurer que les droits de ce fichier sont corrects. Vous devez déjà avoir une idée de ce que sont ces fichiers si vous avez lu la section « le répertoire RedHat ».

5.3.3.3. Mettre à jour le fichier `comps.xml`

Avec la distribution Linux RedHat 8.0, le format du fichier `comps` a complètement changé et il est maintenant basé sur XML. Ce nouveau format apporte une plus grande flexibilité et facilité de personnalisation. La section « le fichier `comps` » vous donnera plus d'informations sur le sujet. Si vous avez modifié ou si vous souhaitez modifier la liste des paquets installés, vous aurez besoin de réaliser cette étape. Ce qui implique alors d'avoir installé une version modifiée du paquet `comps-9.tar.gz` [[rhcd-scripts/comps-9.tar.gz](#)] (l'original ne fonctionne pas pour moi) ou `comps-8.0.tar.gz` [[http://rhlinux.redhat.com/anaconda/comps-8.0.tar.gz](#)] (suivant la version que vous construisez) qui contient le fichier maître `comps` trouvé sur le site web de RedHat, ainsi que le paquet `comps-extras`. Suivez alors ces étapes pour la Redhat 9 et 8.0 :

```
$ cd /répertoire-de-votre-choix
$ tar xzvf
/chemin-vers-comps-9.tar.gz/comps-9.tar.gz
$ cd comps
$ make
$ cat comps-milan.xml |sed 's!</comps>!!g' >comps-tmp.xml
$ /usr/share/comps-extras/getfullcomps.py comps.xml \
/chemin-absolu-de-la-racine i386 >>
comps-tmp.xml
$ echo '</comps>' >> comps-tmp.xml
$ cp comps-tmp.xml
/chemin-absolu-de-la-racine/i386/RedHat/base/comps.
xml
```

En plus de */chemin-absolu-de-la-racine*, vous devrez prendre soin d'indiquer des noms valides pour */répertoire-de-votre-choix* et */chemin-vers-comps-9.tar.gz*. Le reste des commandes pourra simplement être recopié. Et vous devrez évidemment changer 9 en 8.0 si vous construisez une version 8.0.

De nouveau, avant de lancer la commande **make**, vous devrez modifier le fichier `comps-milan.xml.in` en utilisant votre éditeur de texte favori et en suivant les lignes de conduite et suggestions de la section « le fichier `comps` » et de la page « `anaconda comps` » [[http://rhlinux.redhat.com/anaconda/comps.html](#)] du site web RedHat.

Toutes les étapes nécessaires après la commande **make** seront réalisées par le script de la section « le script `updateBuild.sh` ». Ce script utilise la variable `COMPSFILE`, pour trouver le fichier `comps-milan.xml` (il n'a pas besoin d'avoir ce nom, j'utilise juste le nom original, mais vous pouvez le changer si vous le voulez).

Si vous utilisez la distribution Redhat 7.3, le fichier `comps` (avez-vous remarqué la différence de nom ?) est un fichier de texte avec une syntaxe complètement différente. Cette syntaxe est décrite plus précisément dans le fichier `comps`. Pour cette distribution, les seules opérations nécessaires sont l'adaptation du fichier pour correspondre à vos besoins et la copie du fichier RedHat `/base/comps` dans l'arborescence principale en remplacement de l'original.

5.3.3.4. Reconstruire l'installateur

Cette étape consiste à reconstruire l'installateur anaconda dans votre copie locale de la distribution en utilisant les paquets mis à jour. Pour la Redhat 9, lancez :

```
$ /usr/lib/anaconda-runtime/buildinstall \
--pkgorder /chemin-absolu-de-la-racine/pkgorder.txt \
--comp dist-9 --product "Red Hat Linux" --version 9 \
--release "Redhat 9 (Shrike)" \
/chemin-absolu-de-la-racine/i386
```

Où, une fois encore, */chemin-absolu-de-la-racine* est le répertoire où est placée la racine de votre copie locale de la distribution.

Pour la distribution Redhat 8.0, la procédure est pratiquement identique (l'option `--product` en moins) :

```
$ /usr/lib/anaconda-runtime/buildinstall \
--pkgorder /chemin-absolu-de-la-racine/pkgorder.txt \
--comp dist-8.0 --version 8.0 --release "Redhat 8.0 (Psyche)" \
/chemin-absolu-de-la-racine/i386
```

Ou si, comme moi, vous utilisez toujours une Redhat 7.3 :

```
$ /usr/lib/anaconda-runtime/buildinstall \
--pkgorder \
/chemin-absolu-de-la-racine/pkgorder.txt \
--comp dist-7.3 --version 7.3 \
/chemin-absolu-de-la-racine/i386
```

L'absence de l'option (obligatoire pour la 8.0) `--release` est la seule différence notable.

5.3.3.5. Diviser la distribution

Dans cette étape, nous allons créer cinq répertoires, chacun correspondant à un CD différent et y placer des liens physiques vers les fichiers réels contenus dans votre copie locale de la distribution.

Note

Cette étape ne marchera pas du tout avec la distribution RedHat 7.3 si vous n'utilisez pas la version modifiée du script `splitdistro` présentée dans le prochain paragraphe. Pour les distributions RedHat 8.0 et 9, une version modifiée de `splitdistro` est proposée principalement parce que, même si les problèmes du script précédent ont été corrigés, l'exécution échouait systématiquement s'il n'existait pas suffisamment de paquets pour remplir tous les CD (les quatre premiers complètement et le dernier même seulement partiellement).

```
$ /usr/lib/anaconda-runtime/splitdistro \
--fileorder \
/chemin-absolu-de-la-racine/pkgorder.txt --release \
"Redhat 9.0 (Shrike)" /chemin-absolu-de-la-racine \
i386
```

La seule chose que vous ayez besoin de changer pour les versions 8.0 et 7.3 est le texte indiqué à l'option `--release` (qui doit être « Redhat 8.0 (Psyche) » ou « Redhat 7.3 (Valhalla) »).

Pour la distribution Redhat 7.3, la version du script (python) `splitdistro7.3` [outils/RedHat-CD-HOWTO/rhcd-scripts/splitdistro7.3] utilisée a été extraite du paquet *anaconda-runtime 7.3.7* et modifiée par moi. Vous pouvez le substituer à l'original, nommé /

`/usr/lib/anaconda-runtime/splitdistro`, après avoir éventuellement sauvegardé ce dernier.

La seule modification (en dehors de quelques petites corrections) que ce script aie subie est un changement de son comportement si le répertoire `SRPMS` n'est pas trouvé (il n'échoue pas, mais génère les CD sans paquets source).

Pour la distribution Redhat 8.0, la version du script (python) `splitdistro8.0` [outils/RedHat-CD-HOWTO/rhcd-scripts/splitdistro8.0] utilisée a été extraite du paquet *anaconda-runtime 8.0.4* et modifiée une nouvelle fois par moi pour obtenir quelques améliorations dont je ressentais le besoin. Vous devez le substituer au fichier original (peut-être après avoir sauvegardé ce dernier) nommé `/usr/lib/anaconda-runtime/splitdistro`. Néanmoins, l'original fonctionnera bien pour construire une distribution qui comprend tous les paquets *SRPMS* (et ainsi remplir les cinq CD, car sinon le script échouera).

La seule modification apportée au script est un changement dans son comportement s'il ne trouve pas le répertoire `SRPMS` (il n'échoue plus, mais génère les CD sans paquets source) ou s'il y a un CD sans paquet (le script générera un répertoire vide un lieu d'échouer).

Pour la distribution Redhat 9, vous trouverez une copie du script incluant les mêmes modifications que le script de la 8.0 ici : `splitdistro9` [rhcd-scripts/splitdistro9]. Tout ce qui a été dit dans le paragraphe précédent pour la distribution Redhat 8.0 s'applique à la version 9.

5.3.3.6. Régénérer les fichiers `hdlist` et `hdlist2`

Il est nécessaire de recréer les fichiers `hdlist` et `hdlist2`, en utilisant certaines des informations obtenues dans les étapes précédentes. Il n'y a pas de différences entre 7.3, 8.0 et 9 pour l'exécution de ce programme. La commande à utiliser est la suivante :

```
$ /usr/lib/anaconda-runtime/genhdlist \
  --fileorder
  /chemin-absolu-de-la-racine/pkgorder.txt
  --withnumbers \
  /chemin-absolu-de-la-racine/i386-disc[1-3]
```

Comme vous pouvez le voir, comparé à la première utilisation de `genhdlist`, deux nouvelles options sont passées au programme. La première, `--fileorder`, indique à `genhdlist` d'utiliser le fichier `pkgorder.txt` que nous avons généré à la seconde étape (reconstruire l'installateur). Ce fichier contient la répartition des paquets sur les différents CD et est utilisé par l'installateur pour déterminer dans quel ordre les paquets doivent être installés. De manière simple, si vous ne l'utilisez pas, vous finirez probablement par changer très souvent de CD lors de l'installation. L'option `--withnumbers` est nécessaire pour associer un numéro de CD à chaque paquet (comme vous le voyez, un joker indiquant les trois premières images iso est utilisé).

5.3.3.7. Générer les images iso

Dans cette étape, vous allez préparer les images iso à graver sur les CD réels. Il y a deux commandes distinctes à utiliser, l'une pour le premier disque et l'autre pour le reste des CD. Ceci est dû au besoin de rendre amorçable le premier CD. Ce n'est pas strictement nécessaire car vous pouvez utiliser à la place une disquette de démarrage mais il s'agit d'une fonctionnalité intéressante (et elle rend le comportement de vos disques plus similaire aux originaux). Voici les commandes que j'utilise pour réaliser cette tâche :

```
$ mkdir /repertoire-destination-images
$ mkisofs -r -J -T -v -V "Red Hat 9 (Shrike) disque 1" \
  -c isolinux/boot.cat -b isolinux/isolinux.bin -no-emul-boot \
  -boot-load-size 4 -boot-info-table \
  -o /repertoire-destination-images/i386-discl.iso .
```

Cela est nécessaire pour graver le premier disque (amorçable) pour les distributions Redhat 8.0 et 9 (sans émulation de disquette) et c'est exécuté à partir du répertoire racine de la distribution. Le ré-

pertoire `/repertoire-destination-images` est le contenant pour les cinq images iso que vous avez générées et il doit exister avant de pouvoir lancer la procédure. La seule modification à effectuer pour la distribution Redhat 8 est le nom du volume, qui devrait être « Red Hat 8.0 (Psyche) disque 1 ».

```
$ mkdir /repertoire-destination-images
$ mkisofs -r -J -T -v -V "Red Hat 7.3 (Valhalla) disc 1" \
  -c boot.cat -b dosutils/autoboot/boot.img \
  -o /repertoire-destination-images/i386-disc1.iso .
```

C'est nécessaire pour graver le premier disque (amorçable) sur la 7.3. Cette commande doit être exécutée à partir du répertoire racine de la distribution (cette fois-ci nous utilisons l'émulation de disquette).

Le reste des images peut être écrit au moyen d'une boucle « for » :

```
$ for i in `echo 2 3 4 5` ; do mkisofs -r -J -T -v \
  -V "Red Hat 9 (Shrike) disc ${i}" \
  -o /repertoire-destination-images/i386-disc${i}.iso . ; done
```

La boucle présentée va préparer les quatre dernières images en leur donnant les bons numéros. Comme vous pouvez le voir, il y a deux options manquantes par rapport à la commande précédente et, comme vous pouvez le deviner, ces commandes ne sont utiles que pour créer un CD amorçable. Dans créer des images iso, vous pourrez lire une brève explication sur des différentes options et de leurs significations (la plupart ont été extraites de la page de manuel). De nouveau, si vous construisez une distribution Redhat 8.0, vous devriez changer le nom du volume par « Red Hat 8.0 (Psyche) disc1 ».

5.3.3.8. Implanter et vérifier les signatures md5 dans les images iso

C'est une étape optionnelle mais elle permet l'utilisation de l'option « checkmedia » pour vérifier les signatures des CD avant de les installer et donc de garantir leur intégrité.

Les commandes suivantes permettent d'injecter ou de vérifier une signature md5 sur une image iso :

```
$ /usr/lib/anaconda-runtime/implantisomd5 image-iso
$ /usr/lib/anaconda-runtime/checkisomd5 image-iso
```

Après avoir fini toutes ces étapes, nous aurons obtenu les cinq images CD à graver. Considérant que saisir tout ceci est un peu long, la prochaine section présentera un script qui réalise toutes ces opérations en une seule fois (n'oubliez pas de configurer correctement les paramètres).

5.3.3.9. Réunir toutes ces étapes

Le script `updateBuild.sh` [outils/RedHat-CD-HOWTO/rhcd-scripts/updateBuild.sh] exécutera toutes les étapes nécessaires à la reconstruction des CD des distributions Redhat 7.3, 8.0 et 9 en un seul lancement (sous le compte `root`). Avant d'utiliser ce script, vous devrez définir une variable `RHCDPATH` pointant vers le répertoire où se trouve le fichier `rhcd.conf` [outils/RedHat-CD-HOWTO/rhcd-scripts/rhcd.conf] et paramétrer ce fichier. Si vous voulez inclure un fichier `comps.xml` modifié (ou `comps`) dans vos CD, comme expliqué dans le fichier `comps`, vous devrez le copier à l'emplacement défini par la variable `COMPSFILE` avant d'exécuter le script. Si vous en avez besoin, n'oubliez pas d'ajouter le script modifié `splitdistro` dans le répertoire `/usr/lib/anaconda-runtime`.

```
# export RHCDPATH=/home/luigi/tmp/rhcd-scripts
# sh updateBuild.sh
```

6. Graver le ou les CD

Cette étape est composée d'une partie optionnelle et d'une partie obligatoire. Rappelez-vous que vous aurez probablement besoin d'être connecté sous le compte « root » de votre machine pour lancer **cdrecord**.

6.1. Essayer l'image ou les images

Si vous êtes paranoïaque, vous pouvez tester votre nouvelle image disque en la montant. Si vous oubliez de corriger les droits des fichiers ou de mettre en place les extensions rock ridge, alors l'erreur sera évidente puisque les noms de fichier et la structure des répertoires seront erronés. Le test (optionnel) peut être réalisé en entrant la commande suivante :

```
# mount -t iso9660 -o ro,loop=/dev/loop0 image-iso /mnt/cdrom
```

Où *image-iso* est le nom que vous aurez donné à l'image iso à monter (qui doit être seule pour les versions supérieures ou égales à la 6.2). Quand vous avez terminé, n'oubliez pas de la démonter :

```
# umount /mnt/cdrom
```

6.2. Graver le ou des disques

Assurez-vous que vous avez défini les bons paramètres pour votre périphérique. Par exemple, la commande suivante est destinée à un graveur 4x, ce qui est assez lent. De plus, on suppose que le graveur de CD est ici sur le bus SCSI numéro 0, avec l'ID 0 et le LUN 0. Vous pourrez obtenir ces valeurs en lançant un **cdrecord -scanbus** et en les assignant au paramètre « -dev= ».

```
# cdrecord -v speed=4 dev=0,0,0 /repertoire-destination-images/disc1.img
```

7. Le fichier comps

Le fichier `comps` définit la façon dont les paquets seront regroupés durant l'installation. Dans la distribution RedHat, ceci est fait selon les fonctionnalités qu'ils offrent, par exemple :

- Utilisation d'imprimantes
- Système X Window
- GNOME
- KDE
- Outils courrier, web et forums
- ...
- Développement du noyau
- Documentation supplémentaire

Quelquefois durant le processus d'installation, l'utilisateur se trouve face à une fenêtre appelée « Composants à installer ». Quelques-uns des composants ont été présélectionnés, d'autres non. Le dernier point sur la liste des composants est appelé « Tout ». Sur la fenêtre, il existe aussi une option de personnalisation qui permet à l'utilisateur de choisir très précisément la liste des paquets qui se-

ront installés. Personnaliser l'installation ou sélectionner « Tout » dans la liste des composants est le seul moyen d'obtenir que votre propre sélection de paquets soit installée, sauf si vous modifiez le fichier `RedHat/base/comps`.

7.1. Format du fichier `comps` pour RedHat versions < 6.1

Le fichier `comps` commence avec un en-tête indiquant la version du format du fichier `comps`, suivie d'une ligne vide.

```
0.1
<empty line>
```

Puis, il dresse la liste des composants, séparés par des lignes vides :

```
<composant 1>
<ligne vide>
<composant 2>
<ligne vide>
...
<composant n>
<ligne vide>
EOF
```

Chaque composant est défini comme suit :

```
(0|1) (--hide)? <name>
<RPM 1>
<RPM 2>
...
<RPM n>
end
```

Avant le nom de chaque composant est placé un 0 ou un 1. Une valeur de 1 à cet endroit indique que le composant est choisi par défaut alors qu'un 0 signifie le contraire. L'option `--hide` veut dire que l'entrée ne sera pas affichée, sauf si vous choisissez l'installation « en mode expert ». Le premier composant est appelé « Base » et il est spécial, dans le sens où il *doit* être présent et qu'il n'apparaîtra pas dans le dialogue (vous ne pouvez pas désélectionner l'installation de la base, ce qui est sensé). Suit une liste de paquets RPM appartenant à ce composant. Notez qu'il s'agit du nom du paquet stocké *dans le fichier rpm* et *non* d'une partie du nom de fichier du paquet (bien qu'il soient sensés, par convention, être identiques).

En ajoutant vos paquets au fichier `comps`, vous pourrez personnaliser votre propre distribution et vous assurer que vos paquets seront installés par défaut. Une chose à laquelle vous devez faire attention est l'interdépendance entre vos paquets, mais ici c'est à vous de jouer ;-) Un mot pour vous prévenir : faites attention de ne pas ajouter ou supprimer des espaces blancs supplémentaires dans le fichier. Examinez le fichier `comps` existant (faites une copie de l'original) pour voir comment il est fait (ou vérifiez `i386/misc/src/install/pkgs.c` si vous voulez comprendre comment le fichier est analysé).

7.2. Format du fichier `comps` pour RedHat version 6.1

Avec la version 6.1 de la distribution RedHat, le format du fichier `comps` a changé. Le décodage s'effectue dans `/${RHROOT}/${ARCH}/misc/src/anaconda/comps.py`. Je n'ai pas encore analysé ce script python et les règles suivantes ont été obtenues seulement en étudiant le fichier `comps` et en essayant quelques paramètres.

Dans la version 6.1, la définition du *composant* est étendue pour inclure quelques éléments optionnels supplémentaires avant les *<RPM>*. Ces éléments sont :

```
<RPM-dépendant-de-l-architecture 1>
...
<RPM-dépendant-de-l-architecture n>
<composant-requis 1>
...
<composant-requis n>
<RPM-dépendant-d-un-composant 1>
...
<RPM-dépendant-d-un-composant n>
```

Un *<RPM-dépendant-de-l-architecture>* définit une dépendance entre un paquet et une architecture spécifique et a pour définition :

```
(!)?arch: <RPM>
```

Donc, il peut, par exemple, se présenter dans le monde réel comme :

```
!alpha: kernelcfg
```

ce qui veut dire : si l'architecture n'est pas alpha alors il faut installer le paquet *kernelcfg*.

Ou comme :

```
i386: mkbootdisk
```

Ce qui veut dire : si l'architecture est i386 alors il faut installer le paquet *mkbootdisk*.

Un *<composant-requis>* renforce la dépendance avec un autre composant et il est défini comme :

```
@ <composant>
```

Donc, par exemple, si à l'intérieur de la définition d'un composant, vous trouvez la ligne suivante :

```
@ Station réseau
```

cela veut dire que le composant lui-même a besoin de l'installation d'un autre composant nommé *Station réseau*.

Un *<RPM-dépendant-d-un-composant>* est utilisé pour sélectionner l'installation de quelques paquets additionnels pour un composant, étant donné la présence d'un autre composant. Sa définition est la suivante :

```
? <composant> {
  <RPM 1>
  ...
  <RPM n>
}
```

Donc si, par exemple, dans la définition d'un composant, il vous arrive de lire les lignes suivantes :

```
? KDE {
  kpppload
}
```

alors si le composant *KDE* est installé, le paquet *kpppload* sera installé en même temps que les paquets inclus dans le composant où se trouve cette définition.

7.3. Format du fichier comps dans RedHat version 6.2

Avec RedHat version 6.2, le format du fichier `comps` a apparemment très légèrement changé. Le décodage se fait toujours dans `#{RHROOT}/#{ARCH}/misc/src/anaconda/comps.py`. Encore une fois, je n'ai pas analysé ce script python et les règles suivantes ont été obtenues uniquement en étudiant le fichier et en essayant quelques paramètres.

Dans la version 6.2, la définition du composant est étendue pour inclure deux éléments optionnels supplémentaires :

```
<RPM-dépendant-de-la-langue 1>
...
<RPM-dépendant-de-la-langue n>
<composant-dépendant-de-l-architecture 1>
...
<composant-dépendant-de-l-architecture n>
```

Un `<RPM-dépendant-de-la-langue>` est nécessaire pour demander l'installation d'un paquet lorsqu'une langue spécifique a été sélectionnée. C'est défini ainsi :

```
(lang <langue>): <RPM>
```

Par exemple, la ligne suivante

```
(lang ja_JP): locale-ja
```

veut dire : si la langue japonaise est sélectionnée, alors il faudra installer le paquet *locale-ja* en plus des autres paquets installés pour ce composant.

Un `<composant-dépendant-de-l-architecture>` étend le concept du `<RPM-dépendant-de-l-architecture>`, introduit pour la version 6.1, au composant entier, comme vous pouvez le voir à la lecture de sa définition :

```
(!)?arch: <composant>
```

7.4. Format d'un fichier comps dans la RedHat version 7.3

Avec la version 7.3 de la distribution RedHat, le format du fichier `comps` possède une syntaxe plus puissante. Le décodage prend place (encore) dans le script `comps.py`, que vous trouverez dans le répertoire `/usr/lib/anaconda/` si vous avez installé le paquet *anaconda*. Les dépendances d'un composant ou d'un paquet envers une langue ou une architecture données peuvent maintenant être liées par l'opérateur logique `and` (et logique). Par exemple :

```
(arch !s390 and arch !s390x and arch !ia64): readline2.2.1
```

ce qui veut dire que si l'architecture n'est ni s390, ni s390x, ni ia64, alors il faudra installer le paquet `readline2.2.1`. Ceci peut être fait avec des composants au lieu des paquets et avec des langues à la place des architectures. Tout ceci est sans aucun doute plus qu'assez pour les exemples simples de personnalisation de l'installation par défaut qui seront présentés dans la section suivante.

7.4.1. Personnaliser l'installation par défaut de la RedHat version 7.3

L'exemple que nous allons suivre dans cette section implique de modifier le fichier `comps` pour changer les valeurs par défaut concernant l'installation des paquets. Je préfère habituellement (en fait spécialement dans certaines situations) une installation par défaut incluant seulement les paquets de base, avec quelques légères modifications pour certains d'entre eux. Dans le premier des exemples présentés, nous construirons une installation par défaut qui ajoute *libsafe* au composant « Base », dont la plupart des paquets, qui sont généralement installés par défaut, seront désélectionnés dans le but de construire une installation minimale. Dans le second des exemples, nous modifierons quelques-uns des composants pour construire une autre installation minimale qui réponde à nos besoins (cette fois, pratiquement parfaitement ; ce sont, en fait, mes besoins, les vôtres pouvant varier). Si vous voulez inclure un fichier `comps` modifié dans vos CD, vous devrez le copier dans l'arborescence principale juste avant de lancer les opérations décrites dans reconstruire l'installateur 7.3 ou 8.0.

7.4.1.1. Ajouter des RPMS et désélectionner les composants par défaut

Pour personnaliser votre installation de cette façon, vous devrez éditer le fichier `comps` avec votre éditeur de texte favori (faites attention à ne pas laisser d'espaces ou de tabulations dans ce fichier) et le déplacer dans le répertoire `Redhat/base` en écrasant l'original.

Dans le premier fichier `comps` [`outils/RedHat-CD-HOWTO/rhcd-scripts/comps/comps.1`] inclus, le paquet *libsafe* est ajouté au composant « Base » (système de base) et presque tous les composants sont désélectionnés pour obtenir une installation par défaut comportant seulement 200 paquets (je sais qu'ils peuvent être encore trop nombreux).

7.4.1.2. Modifier quelques-uns des composants standards

Nous avons construit le deuxième fichier `comps` [`outils/RedHat-CD-HOWTO/rhcd-scripts/comps/comps.2`] ci-joint à partir de la configuration précédente en réduisant un peu plus l'installation par défaut (cette fois, elle ne contient plus que 154 paquets). Quelques-uns des groupes ont été divisés pour donner à l'installation une meilleure granularité. Toutes les modifications que vous faites doivent prendre en compte les interdépendances entre paquets et les applications utilisées durant les phases d'installation (par exemple, vous ne pouvez pas supprimer *kudzu* du composant *Base*, bien que vous puissiez le faire après installation). Je dois aussi vous préciser que des résultats similaires peuvent être obtenus en utilisant *kickstart*. Pour plus d'informations à ce propos, vous pourrez lire le Guide de Personnalisation du Linux RedHat [<http://www.europe.redhat.com/documentation/rhl7.3/rhl-cg-fr-7.3/ch-kickstart2.php3>].

7.5. Format du fichier `comps` à partir des versions 8.0 et 9 de RedHat

Avec les versions 8.0 et 9, le format du fichier `comps` a complètement changé et on utilise maintenant un fichier XML, nommé `comps.xml`. Les détails de la syntaxe de ce fichier sont présentés dans la section Les `comps` d'Anaconda [<http://rhlinux.redhat.com/anaconda/comps.html>] du site web de RedHat.

7.5.1. Personnaliser l'installation par défaut de la RedHat version 8.0

Nous allons maintenant reprendre les exemples présentés pour la version 7.3 en prenant en compte

les modifications qu'ont subi les différents groupes. Le groupe le plus important (le groupe « Base ») est ici divisé en deux groupes nommés « Base » et « Core ». Le groupe « Base » devrait représenter la plus petite installation possible.

7.5.1.1. Notre premier exemple revisité pour Redhat 8.0

Cette fois, pour personnaliser votre installation, vous devrez éditer le fichier `comps-milan.xml.in` avec votre éditeur de texte favori. Ce fichier se trouve dans l'archive `comps-8.0.tar.gz` [<http://rhlinux.redhat.com/anaconda/comps-8.0.tar.gz>] disponible sur le site web de RedHat. Pour ajouter les informations relatives aux paquets au fichier que vous créez, vous aurez besoin d'avoir installé le paquet `rpm comps-extras`. Les commandes à lancer pour terminer les opérations sont indiquées dans `mettre à jour comps.xml` et dans la documentation [<http://rhlinux.redhat.com/anaconda/comps.html>]. Après avoir créé le fichier, vous devrez le copier dans le répertoire `Redhat/base` en écrasant l'original. Si vous utilisez le script `updateBuild.sh`, vous devrez seulement copier `comps-milan.xml` (après avoir modifié le `comps-milan.xml.in` qui se trouve dans le paquet `tar/gzip comps-8.0.tar.gz` et lancer la commande `make`), à l'emplacement que vous avez déjà indiqué dans la variable `COMPFILE` (dans `rhcd.conf` [outils/RedHat-CD-HOWTO/rhcd-scripts/rhcd.conf]).

Dans le premier fichier `comps` [outils/RedHat-CD-HOWTO/rhcd-scripts/comps/comps-milan.xml.in.1] ci-joint, le paquet `libsafe` a été ajouté au groupe (composant) « Base » et pratiquement tous les groupes (composants) ont été désélectionnés, sauf « Base » et « Core », pour avoir une installation par défaut de seulement 220 paquets environ (probablement trop nombreux, encore une fois).

7.5.1.2. Notre deuxième exemple revisité pour Redhat 8.0

Nous avons construit le deuxième fichier `comps` [outils/RedHat-CD-HOWTO/rhcd-scripts/comps/comps-milan.xml.in.2] ci-joint en partant de la configuration précédente et en réduisant un peu plus l'installation par défaut (cette fois, il ne restera que 158 paquets dans l'installation par défaut). Encore une fois, des résultats similaires peuvent être obtenus en utilisant `kickstart`, pour plus d'informations à ce propos, vous pourrez lire le Guide de personnalisation du Linux RedHat [<http://www.redhat.com/docs/manuals/linux/RHL-8.0-Manual/custom-guide/ch-kickstart2.html>]. Dans cet exemple, je n'ai pas complètement désélectionné l'installation du groupe « Base », parce qu'il contient trop de paquets que j'utilise d'habitude. J'ai donc juste désélectionné l'installation par défaut pour ces paquets en les rendant optionnels. Comme vous pouvez le voir, même le paquet `redhat-logos` du groupe « Core » a été rendu optionnel. Sachant que tous les paquets de ce groupe doivent représenter, ensembles, la *plus petite installation possible*, vous ne voudrez sans doute pas le faire (mes CD fonctionnent même même comme cela ; cependant, il doit y avoir quelques problèmes que je n'ai pas encore détectés). Le paquet `tripwire` a aussi été ajouté au groupe « Base ». La dernière modification visible a été faite au groupe « dialup », qui sera installé même s'il est désélectionné, parce que le groupe « Base » en dépend (ce qui est indiqué dans la définition du groupe lui-même). J'ai seulement sélectionné pour installation certains paquets de ce groupe dont j'ai habituellement besoin et laissé le reste désélectionné.

7.5.2. Personnaliser l'installation par défaut de la RedHat version 9

Nous allons de nouveau reproduire les exemples présentés pour les versions 7.3 et 8 en prenant en compte les modifications qu'ont subies les différents groupes.

7.5.2.1. Notre premier exemple revisité pour la Redhat 9

Comme dans le cas de la 8.0, pour personnaliser votre installation, vous devrez modifier le fichier `comps-milan.xml.in` avec votre éditeur de texte favori. Ce fichier est disponible dans le fichier `comps-9.tar.gz` [rhcd-scripts/comps-9.tar.gz] avec les scripts (*comme je l'ai déjà dit, vous ne trouverez pas la même chose sur le site web de Redhat*). Pour ajouter les informations relatives aux paquets au fichier que vous créez, vous aurez besoin d'avoir installé le paquet `rpm comps-extras`. Les commandes à utiliser sont indiquées dans `mettre à jour comps.xml` et dans la documentation [<http://rhlinux.redhat.com/anaconda/comps.html>]. Après avoir créé le fichier, vous

devrez le copier dans le répertoire `Redhat/base` en écrasant l'original. Si vous utilisez le script `updateBuild.sh`, vous devrez seulement copier `comps-milan.xml`, (après avoir modifié `comps-milan.xml.in` trouvé dans le paquet `tar/zip comps-9.tar.gz` et avoir lancé la commande `make`), vers la destination que vous avez déjà indiquée dans la variable `COMPSFILE` (`rhcd.conf` [`rhcd-scripts/rhcd.conf`]).

Dans le premier fichier `comps` [`rhcd-scripts/comps/comps-milan.xml.in.1-9`] inclus, le paquet *libsafe* a été ajouté au groupe (composant) « Base » et pratiquement chaque groupe (composant) a été désélectionné, mis à part « Base » et « Core », de façon à avoir une installation par défaut comprenant seulement environ 240 paquets (hummm, la complexité augmente beaucoup...).

7.5.2.2. Notre second exemple revisité pour Redhat 9

Dans le second fichier `comps` [`rhcd-scripts/comps/comps-milan.xml.in.2-9`] ci-joint, nous partons de la configuration précédente et nous réduisons un peu plus l'installation par défaut (cette fois, il n'y aura qu'environ 175 paquets dans l'installation par défaut). Ceci ressemble beaucoup à l'exemple présenté pour la Redhat 8.0, donc je vais éviter de vous ennuyer avec les mêmes explications. Encore une fois, des résultats identiques peuvent être obtenus en utilisant *kickstart*. Vous trouverez plus d'informations sur le sujet en lisant le Guide de personnalisation du Linux RedHat [<http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/custom-guide/ch-kickstart2.html>].

8. Installation à partir du CD

Lors de l'installation à partir du nouveau CD, vous pourriez avoir tout d'abord besoin de créer une disquette d'installation amorçable. **IMPORTANT** : *utilisez une disquette neuve, tout juste formatée en format MS-DOS !*. Utiliser une vieille disquette peut entraîner des problèmes étranges lors de l'installation ! Sur un système Linux, vous pourrez créer la disquette en utilisant la commande `dd` :

```
$ dd if=/mnt/cdrom/images/boot.img of=/dev/fd0 bs=1440k
```

Sur un système tournant sous DOS ou Windows-9x, vous aurez besoin d'utiliser le programme `rawrite.exe`, disponible dans le répertoire `dosutils` du CD. Sur une machine sous Windows-9x/Me/NT/2k, vous pourrez utiliser `rawritewin.exe` situé dans le répertoire `dosutils/rawritewin`.

Arrêtez la machine sur laquelle vous voulez installer votre CD (ou faire une mise à jour du système), insérez la disquette de démarrage et votre CD fraîchement gravé, et laissez la machine démarrer à partir de la disquette. Pour plus d'informations sur le processus d'installation, voir les documents ainsi que le Guide pratique Installation

[<http://www.traduc.org/docs/howto/lecture/Installation-HOWTO.html>] ou le Guide pratique des disques d'amorçage [<http://www.traduc.org/docs/howto/lecture/Bootdisk-HOWTO.html>].

8.1. Démarrer d'un CD amorçable

La plupart des machines modernes sont capables de démarrer à partir d'un CD, à condition qu'il soit rendu amorçable par procédure indiquée dans la section créer des images iso. Souvent, néanmoins, vous aurez besoin de changer le paramétrage du BIOS pour permettre le démarrage à partir du lecteur de CD. Référez-vous à la documentation de votre carte-mère pour savoir comment le faire.

9. Autres distributions Linux

Les informations concernant les versions inférieures ou égales à la version 6.1, présentes dans les précédentes versions de ce guide pratique (# 1.34) et reprises dans le présent document, sont supposées s'appliquer à toutes les distributions clones de la RedHat, telle que la distribution Mandrake [<http://www.linux-mandrake.com/fr/>]. Malgré tout, cette procédure n'a pas été vérifiée [pour d'autres distributions] (comme vous pouvez le lire dans le guide pratique lui-même).

C'est en gros la même chose pour la distribution LinuxPPC [<http://www.linuxppc.org>] pour les PowerMac d'Apple. Pour créer une distribution pour la plate-forme PowerMac, vous devrez utiliser

mkhybrid [<http://rufus.w3.org/linux/RPM/mkhybrid.html>] à la place de `mkisofs` et ce devrait être la seule différence.

Les informations concernant les nouvelles versions de RedHat (> 6.1) ne devraient pas fonctionner avec la distribution Mandrake, qui dispose maintenant d'un installateur assez différent de celui de RedHat. Je ne sais absolument pas s'il est possible de mettre à jour les CD d'autres distributions clones de la distribution RedHat de la même façon, mais je serais heureux d'en être informé.

10. Ce document...

Le code source SGML de la plus récente version originale de ce document est disponible ici [<http://www.tldp.org/HOWTO/RedHat-CD-HOWTO/RedHat-CD-HOWTO>].

Le code source XML de la dernière version française de ce document est disponible sur le site du projet Traduc.org [<http://www.traduc.org>] :
<ftp://ftp.traduc.org/pub/traduc.org/doc-vf/HOWTO/telechargement/sgml/>.

10.1. Documents connexes

10.1.1. Documentation relatifs à la version actuelle

Les documents suivants ont été utiles pour la création de ce guide pratique :

Le « Petit guide (non officiel) de personnalisation de l'installateur de la Redhat 7 » de Tony Nugent. Ce document est très intéressant et utile. Donc, si vous pensez sérieusement construire des CD personnalisés, je vous suggère fortement de le lire. Vous le trouverez sur www.linuxworks.com.au [<http://www.linuxworks.com.au/redhat-install-howto.html>].

Miguel Freitas a écrit le « Petit guide des CD RedHat 7 », que vous pourrez lire sur ce site web [<http://cambuca.ldhs.cetuc.puc-rio.br/RedHat7-CDs-HowTo.html>].

Ron Yorston a écrit le document `rpmhack` [<http://www.tigress.co.uk/rmy/rh62/rpmhack.html>], qui est intéressant pour la version 6.2 du Linux RedHat.

Quelqu'un (je n'ai pas trouvé son nom) a écrit le document Construire un CDROM Linux Red Hat 6.2 [<http://www.scyld.com/~pzb/rhcd.html>], utile pour la version 6.2.

10.1.2. Documentation relative à l'édition précédente

Avec le sens des bonnes choses de la vie, Jussi Torhonen de Finlande <`jussi POINT torhonen CHEZ tietosavo POINT fi`> nous dit comment créer un CD-ROM RedHat Linux 5.2 amorçable fait-à-la-maison [<http://www.iwn.fi/~jt/cd/>].

Consultez aussi le « Guide pratique de la gravure de CD » [<http://www.traduc.org/docs/howto/lecture/CD-Writing-HOWTO.html>] du projet de documentation Linux (LDP).

10.2. Remerciements

À part les personnes mentionnées ci-dessus, je remercie les personnes suivantes pour leurs informations précieuses, commentaires, discussions et caetera :

10.2.1. Remerciement pour la version actuelle

- Morten Kjeldgaard, <`mok CHEZ imsb POINT au POINT dk`>
- Peter von der Ahé, <`pahe+rhcd CHEZ daimi POINT au POINT dk`>
- Giulia Tomaselli

- Jacinta Conneely
- Filippo Carcaci
- Guillaume Lelarge <gleu CHEZ wanadoo POINT fr>
- Alain Portal <aportal CHEZ univ-montp2 POINT fr>
- Toutes les personnes des listes de discussion anaconda-devel et kickstart

10.2.2. Remerciements pour les versions précédentes

- Lars Christensen <larsch CHEZ cs POINT auc POINT dk>
- Thomas Duffy <tbd CHEZ cs POINT brown POINT edu>
- Dawn Endico <dawn CHEZ math POINT wayne POINT edu>
- Seva <seva CHEZ null POINT cc POINT uic POINT edu>
- Michael Thomas Cope <mcope CHEZ orion POINT ac POINT hmc POINT edu>
- Charles J. Fisher <charles_fisher CHEZ bigfoot POINT com>
- Eric Thomas <eric POINT thomas CHEZ ericsson POINT com>
- Gordon Yuen <gdccyuen CHEZ yahoo POINT com>
- Dave Morse <morse CHEZ nichimen POINT com>

11. Adaptation française

11.1. Traduction

La traduction française de ce document a été réalisée par Guillaume Lelarge <gleu CHEZ wanadoo POINT fr>.

11.2. Relecture

La relecture de ce document a été réalisée par Guillaume Hatt <ghatt CHEZ netcourrier POINT com>.

Une relecture complémentaire de ce document a été réalisée par Jean-Philippe Guérard <fevrier CHEZ tigreraye POINT org>.