



GNU

LINUX

PRATIQUE

# GNU/LINUX ET LES LOGICIELS LIBRES

# EN 80 QUESTIONS

**LE LIBRE**

**CONFIGURATION**

**BUREAUTIQUE**

**DÉFINITIONS**

**SÉCURITÉ**

**GNU/LINUX**

**MULTIMÉDIA**

**LINUX/WINDOWS**

**PROBLÈMES COURANTS**

RECEIVING SIGNAL STRENGTH 100%

FRANCE MÉTRO : 6,40 € - DOM 6,95 €  
BEL : 7,30 € - LUX : 7,30 € - PORT. CONT. : 7,30 €  
CH : 13 FS - CAN : 12 \$ - MAR : 65 DH

L 12225 - 8 H - F: 6,40 € - RD



Bonjour à vous ami(e)s lecteur(trice)s !

Nous avons le plaisir de vous offrir ce numéro afin que vous puissiez profiter d'un aperçu de nos publications en version électronique. Ce numéro 8 de Linux Pratique Hors-série fournit des bases utiles aussi bien aux débutants qu'aux utilisateurs plus avancés de systèmes GNU/Linux. Nous espérons qu'il se révélera toujours aussi pratique dans votre parcours avec les logiciels libres. N'hésitez pas à en faire profiter votre entourage !

Si vous souhaitez parfaire vos connaissances, vous pouvez vous rendre sur notre site UnixGarden ([www.unixgarden.com](http://www.unixgarden.com)) qui vous proposera une sélection d'articles (plus de 1700 à ce jour) précédemment parus au sein de nos publications. Si certains articles retiennent particulièrement votre attention, vous pourrez acquérir les anciens numéros dont ils sont issus, soit en version « papier » sur notre site [www.ed-diamond.com](http://www.ed-diamond.com), soit en version PDF sur [diamond.izibookstore.com](http://diamond.izibookstore.com).

Nous vous remercions de l'intérêt que vous portez à nos publications et nous vous souhaitons une bonne lecture !

Arnaud Metzler  
Directeur de Publication  
Éditions DIAMOND

## Les Éditions Diamond

Site institutionnel : <http://www.editions-diamond.fr/>

Boutique : <http://www.ed-diamond.com/>

Site rédactionnel : <http://www.unixgarden.com/>

Vous pouvez également nous suivre sur Twitter : <https://twitter.com/#!/unixgarden>  
et sur Google + : <https://plus.google.com/103278016435676632910/posts>.

## sommaire

D é c e m b r e  
2 0 0 6  
J a n v i e r  
2 0 0 7

<b>LE LIBRE</b>	<b>04</b>
C'est quoi « GNU » ? / Qu'est-ce que la FSF ?	04
Qu'est-ce qu'un Logiciel libre ?	05
Pourquoi des sources ? / Qui utilise GNU/Linux ?	06/07
Quelles sont les différentes licences libres ?	08
Certaines distributions sont payantes, est-ce normal ?	10
Puis-je aider des projets de Logiciels libres ?	11
Est-ce que GNU/Linux est le seul système d'exploitation libre ?	12
Qui est Richard M. Stallman ?	13
Pourquoi utiliser les Logiciels libres dans l'éducation ?	14
Quels logiciels libres puis-je utiliser avec mes enfants ?	15
<b>GNU/LINUX</b>	<b>16</b>
Doit-on dire Linux ou GNU/Linux ? Où trouver de l'actualité ?	16
Où trouver des logiciels supplémentaires ?	17
Où trouver de l'aide ?	18
Quelle est l'origine de Tux, la mascotte de Linux ?	19
Quel est l'historique de Linux ?	20
Boot, chargeur, noyau, services, X...	21
Serveur X, gestionnaire de fenêtres, bureau...	22
GTK+, QT, programmes Gnome ou KDE ? Peut-on les mélanger ?	23
Faut-il une configuration puissante pour utiliser GNU/Linux ?	24
Qu'est-ce que je ne pourrai pas faire facilement sous Linux ?	25
Pourquoi n'y a-t-il pas d'outils de défragmentation ?	26
Est-il indispensable de savoir utiliser la console ?	27
Quelle distribution choisir ?	28
Pourquoi n'y a-t-il pas de nom de volume sous GNU/Linux ?	29
À quoi correspondent les numéros des distributions ?	30
<b>DÉFINITIONS</b>	<b>32</b>
Qu'est-ce qu'un newbie/noob ? / Ça veut dire quoi « RTFM » ?	32
C'est quoi un geek ? / C'est quoi un troll ?	32/33
C'est quoi l'OSI ? / C'est quoi la LSB ? / C'est quoi l'EFF ?	34/35
Qu'est-ce qu'une distribution Linux ?	36
Qu'est-ce qu'un live-CD ? / Qu'est-ce que la (mémoire) swap ?	37
Qu'est-ce qu'une bibliothèque ? Qu'est-ce que le shell ?	39/40
C'est quoi un Fork ? C'est quoi une commande ?	41/42
Qu'est-ce que UTF-8 et LATIN1 ? / Qu'est-ce qu'un démon ?	42/43
Qu'est-ce qu'un processus ?	45
C'est qui root ? / Qu'est-ce qu'une ABI ?	46/47
<b>CONFIGURATION</b>	<b>48</b>
Comment connaître son matériel ?	48
Comment connaître l'espace disque disponible ?	48
De quelle quantité de mémoire je dispose ?	49
Comment est partitionné mon disque ?	49
Peut-on tout faire en lignes de commandes ?	50
Comment créer un script de commandes ?	51
Quelles sont les commandes Unix de base ?	52
À quoi ça sert un cronjob ?	53
Comment mettre à jour ses paquets de logiciels ?	54
Est-ce bien ou mal la compilation « à la hache » ?	55
Comment créer/désactiver un compte utilisateur ?	56
<b>LINUX / WINDOWS</b>	<b>58</b>
Comment lire mes fichiers MS Excel, MS Word, ...?	58
Existe-t-il des clients de messagerie identiques à Outlook ?	58
Accéder à ses données avec Linux et Windows	59
Faire cohabiter Linux et Windows sur la même machine	60
Peut-on utiliser un logiciel créé pour Windows sous Linux ?	61
Logiciels libres / Logiciels propriétaires	62

Ce numéro a été conçu de façon à aborder tous les points qui peuvent paraître obscurs aux yeux d'un débutant sous Linux, toutes les questions que, peut-être, vous n'avez jamais osé poser sur un forum spécialisé de peur d'être ridicule, tous les concepts qu'il est utile de maîtriser lorsque l'on prend en main son nouveau système, etc. Plus simplement, il s'agit de balayer ici toutes les appréhensions et les réticences qui pourraient encore vous empêcher d'essayer GNU/Linux.

Ainsi, ce hors-série est divisé en 9 rubriques thématiques, qui vous permettront de retrouver plus facilement l'information que vous recherchez. Dans la rubrique **Le Libre**, vous pourrez en apprendre davantage sur toute la communauté : qui sont les différents protagonistes ? qui sont les utilisateurs ? que sont les licences libres ? comment contribuer ? etc. La rubrique **Définitions** renferme, quant à elle, les significations de divers acronymes, termes techniques et autres mots du jargon informatique. Les rubriques **GNU/Linux** et **Configuration** vont vous guider dans la découverte concrète de votre système GNU/Linux : présentations, explications, conseils et avertissements en tout genre vous permettront de vous familiariser rapidement avec Linux. Dans la rubrique **GNU/Linux et Windows**, vous trouverez les réponses aux questions que vous vous posez sûrement sur la cohabitation de ces deux systèmes sur votre disque dur et les similitudes/différences qu'on peut relever entre eux. Les rubriques **Bureautique** et **Multimédia** sont surtout destinées à faire le lien entre les applications que vous aviez l'habitude d'utiliser sur votre système Microsoft Windows et les outils libres qui sont aujourd'hui à votre disposition pour les mêmes usages. Enfin, les rubriques **Sécurité** et **Problèmes courants** renferment de précieux conseils quant à la gestion optimale de votre système Linux, la prise en charge des périphériques et la résolution de petits problèmes fréquemment rencontrés par tout utilisateur débutant.

Tous les auteurs ont tâché d'être le plus clair et le plus didactique possible, de façon à vous apporter un maximum d'informations, que nous avons jugées importantes, voire indispensables, pour quiconque souhaite découvrir Linux.

Bien entendu, le numéro que vous avez entre les mains n'a pas la prétention de passer pour un guide exhaustif de l'utilisation de Linux. En effet, seule une utilisation régulière de votre système, votre investissement personnel et votre détermination sont garants d'un apprentissage réussi. On ne devient pas « Linuxien » du jour au lendemain ! Toutefois, ce numéro vous servira à établir de solides fondations dans votre apprentissage de ce système d'exploitation.

Bonne lecture...

... et joyeuses fêtes de fin d'année à tous !

Fleur Brosseau

<b>BUREAUTIQUE</b>	<b>64</b>
Que sont les fichiers d'extension .gz, .tar.gz, .tar.bz2, ...?	64
Comment faire une capture d'écran sous Linux ?	65
Comment faire ma comptabilité sous Linux ?	66
Comment lire mes fichiers PDF sous Linux ?	66
<b>MULTIMÉDIA</b>	<b>67</b>
Comment lire mes CD audio/DVD/DIVX sous Linux ?	67
Comment graver un CD/DVD sous Linux ?	68
Lire et créer des fichiers MP3	69
<b>SÉCURITÉ</b>	<b>70</b>
Et si je faisais tout en tant que root ?	70
J'ai oublié mon mot de passe, comment faire ?	71
Est-ce que Linux est sécurisé ?	72
Pourquoi n'y a-t-il pas de virus ou de spyware sous Linux ?	73
C'est quoi un port IP ? Je bloque quoi et comment ?	74
<b>PROBLÈMES COURANTS</b>	<b>75</b>
Comment tuer une application qui a planté ?	75
Mémoire vive / Lire des fichiers bureautiques	76/77
Support des périphériques	77
Qu'est-ce qu'un « segmentation fault » ?	78
Je n'arrive pas à booter sur le CD, comment faire ?	79
<b>QUIZ</b>	<b>31/80</b>
<b>abonnements / commandes</b>	<b>81/82</b>

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>



## C'est quoi « GNU » ?

GNU est le premier projet de développement d'un système d'exploitation libre. Il a été motivé et démarré par Richard M. Stallman en 1983 (voir page 13). Le développement de ce système d'exploitation démarra par la programmation d'un compilateur C libre, point de départ essentiel pour le développement des autres outils nécessaires au futur système.

Le sigle GNU en lui-même est un acronyme récursif signifiant *Gnu is Not Unix*, afin de préciser que ce système d'exploitation mimant Unix n'en contenait pas une ligne de code.

En 1985, devant l'ampleur du projet, RMS mit en place la FSF, *Free Software Foundation*, afin de recueillir des fonds et de pourvoir le projet d'une structure juridique.

En 1989, ayant substantiellement avancé son projet, avec déjà l'aide de nombreux collaborateurs, RMS travailla sur la licence libre GPL, *General Public Licence*, afin de protéger de façon concrète les productions libres. Cette licence protège de nos jours de 60 à 70% des Logiciels libres de par le monde.

Enfin, en 1991, alors que le nouveau système d'exploitation était relativement abouti (interfaces, système de fichiers, outils...), il y manquait encore une pièce maîtresse : le noyau. Un noyau GNU, actuellement nommé *GNU Hurd*, avait été planifié par les membres du projet, mais son développement était lent et chaotique. Alors, quand un jeune étudiant finlandais du nom de Linus Torvalds annonça être en train de développer un petit noyau compatible Unix pour occuper son temps libre, il suscita bien plus d'intérêt qu'il ne l'avait prévu. En effet, les premiers prototypes de systèmes GNU fonctionnant avec le noyau de Linus firent leur apparition dès l'année suivante, et, comme l'ensemble était satisfaisant, les premiers systèmes GNU/Linux virent le jour, après que l'« Unix-ification » du prénom de Torvalds eût été choisie pour nommer son noyau.

Après environ une dizaine d'années de très grande tranquillité, le développement du noyau Hurd prévu originellement pour le système GNU connaît quelques soubresauts, mais un système GNU/Hurd n'est toujours pas envisageable pour le grand public. À l'inverse, des hybridations telles que GNU/kFreeBSD, GNU/NetBSD, GNU/Solaris sont, elles, viables, et assureraient la continuité des systèmes d'exploitation libres si jamais l'actuel noyau Linux connaissait des problèmes de licence ou autre.

Simon Descarpentries

## Qu'est-ce que la FSF ?



La FSF, pour *Free Software Foundation* – littéralement Fondation pour le Logiciel libre – est une organisation américaine à but non lucratif, fondée par Richard Stallman en 1985. Basée à Cambridge, dans le Massachusetts, cette organisation a pour vocation de soutenir financièrement le projet GNU et la communauté du Logiciel libre dans son ensemble. Elle tâche ainsi de réunir des fonds en vendant des distributions GNU/Linux, des T-shirts et autres *goodies*... Bien entendu, la FSF produit elle-même des programmes systématiquement diffusés sous licence GPL et dont les principaux représentants forment le projet GNU. Des centaines de programmeurs, parmi les meilleurs, ont travaillé pour elle.

À ce jour, la FSF (<http://www.fsf.org/> [en]) est dirigée par son fondateur, Richard Stallman, et par Peter T. Brown. Elle est gérée en outre par un conseil d'administration constitué de 6 membres, ingénieurs en informatique et professeurs pour la plupart.

Depuis, quelques FSF « locales » ont vu le jour et entreprennent les mêmes actions que leur grande sœur : la *FSF Europe* (FSFE), la *FSF France*, la *FSF India* et la *FSF Latin America*. Toutefois, ces organisations sont toutes quatre distinctes et indépendantes de la FSF. La FSF Europe (<http://www.france.fsf-europe.org/index.fr.html> [fr/en]), dont le siège est basé à Düsseldorf en Allemagne, est née en mars 2001. Organisation à but non lucratif, elle vise non seulement à soutenir le projet GNU, mais aussi à fournir un pôle de compétences aux politiciens, juristes et journalistes, dans le but d'assurer l'avenir légal, politique et social du Logiciel libre.

La FSF France (<http://www.fsf-france.org/> [fr]), quant à elle, est une association à but non lucratif (régie par la loi 1901) dont le rôle est de promouvoir et d'encourager l'utilisation du Logiciel libre en France. Elle a vu le jour en avril 2001 comme partie intégrante de la FSF Europe, mais est devenue totalement indépendante de cette dernière en 2003. Son président, Frédéric Couchet, ainsi que l'ensemble des membres de l'équipe sont particulièrement impliqués dans la promotion et la défense des Logiciels libres. Ainsi, fin 2002, la FSF France a créé le site <http://euclid.info>, dédié à la lutte contre la directive européenne EUCD et sa transposition en droit français la loi DADVSI (Droit d'Auteur et Droits Voisins dans la Société de l'Information), cette dernière menaçant fortement l'avenir du Logiciel libre.

Vous pouvez soutenir chacune de ces associations de diverses manières : dons pécuniaires, développement, traduction, etc. Vous trouverez la marche à suivre en consultant le site officiel.

 FREE SOFTWARE  
FOUNDATION

Fleur Brosseau

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Qu'est-ce qu'un Logiciel libre ?

Si vous avez lu les définitions, en particulier, l'article de présentation de l'OSI, vous savez donc qu'il y a des variations possibles sur la notion de « Logiciel libre » en fonction des licences que vous utilisez. Cependant, tous les Logiciels libres répondent aux critères suivants :

- > vous avez le droit de diffuser le logiciel dans l'état dans lequel il vous a été remis ;
- > vous avez le droit de disposer des sources du logiciel ;
- > vous avez le droit de modifier les sources du logiciel pour votre usage ou celui de quiconque ;
- > vous avez le droit de diffuser le logiciel modifié du moment que vous respectez les droits d'auteurs et que vous maintenez le logiciel dans les mêmes droits (ou encore la même licence) que les auteurs précédents.

Ensuite, en fonction des groupes de développement, des distributions, des associations ou des sociétés et des personnes, d'autres droits peuvent être ajoutés. Ainsi, la définition de l'OSI (voir page 34) compte dix points. Un logiciel libre doit alors :

- > ne pas être discriminatoire envers les personnes ou les groupes ;
- > ne pas interdire d'usage ou de lieux d'usage (pas d'interdiction de l'usage par le gouvernement chinois, sinon ce n'est plus un Logiciel libre. Cela visait alors plutôt les licences des années 1980 qui interdisaient au gouvernement sud-africain l'usage de certains logiciels...);
- > disposer d'une licence qui soit complète en termes de droit, c'est-à-dire que les droits et les devoirs qui s'y attachent ne demandent pas l'exécution d'une autre licence, qui serait alors complémentaire ;
- > disposer d'une licence qui soit complète en termes de logiciel (on ne peut pas diffuser un logiciel qui serait libre, mais obligé de fonctionner dans un ensemble plus vaste pour être valable) ;
- > disposer d'une licence qui ne soit pas exclusive envers d'autres logiciels (on ne peut pas interdire le fonctionnement du logiciel vis-à-vis d'un autre logiciel) ;
- > disposer d'une licence dont l'acceptation ne doit pas nécessiter de technologies pouvant bloquer l'usager du logiciel.

Cette définition est dérivée des travaux de Bruce Perens pour la charte du groupe Debian.

Si vous voulez le point de vue de la *Free Software Foundation*, je vous invite à lire ce document en ligne : <http://www.gnu.org/philosophy/free-sw.fr.html> (en français). Vous verrez qu'il y a quelques différences philosophiques, mais pas de différences pratiques...

Sidoine Pierrel



## CONTRIBUEZ À LINUX PRATIQUE

La rédaction du magazine est à la recherche permanente de nouvelles idées et de nouvelles contributions.

Si vous utilisez régulièrement des applications libres sous GNU/Linux et que vous souhaitez partager vos expériences et vos choix, contactez-nous.

La promotion des Logiciels libres passe principalement par la publicité faite par ses utilisateurs. Si vous vous sentez capable de décrire votre logiciel préféré, ses avantages, ses problèmes et en quoi il vous permet d'améliorer votre quotidien, n'hésitez pas à proposer un article.

Une idée, un sujet, un thème que vous aimeriez voir traité en détail dans un numéro hors-série de Linux Pratique ? N'hésitez pas à nous en faire part !

Pour nous contacter à ce propos, envoyez simplement un courrier électronique à l'adresse ci-dessous :

[contrib@linux-pratique.com](mailto:contrib@linux-pratique.com)

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Pourquoi des sources ? Qu'est-ce qu'on peut en faire ?



Si vous avez lu la définition d'un Logiciel libre (voir page précédente), vous avez remarqué que la mise à disposition des sources du logiciel est un droit fondamental. Pourquoi ? Parce que les sources sont la recette de fabrication du logiciel. C'est grâce à elles que vous pouvez faire évoluer le logiciel, lui ajouter des fonctions, le faire passer d'une plate-forme matérielle à une autre, d'une plate-forme logicielle à une autre. Grâce à elles, vous pouvez si nécessaire recréer le logiciel au moment où vous en avez besoin. Vous n'êtes plus limité à une plate-forme ou un système d'exploitation ou encore un format de données, mais vous pouvez (avec le savoir adéquat et les moyens) reprendre vos travaux où et quand bon vous semble et si nécessaire faire les ajouts adéquats.

Cependant, il y a de fortes chances que vous ne soyez pas développeur, que vous n'ayez pas besoin d'autres choses que d'un traitement de texte, un tableur, un gestionnaire d'images et de vidéos et accessoirement d'une ou deux bricoles en plus. Alors à quoi cela va-t-il vous servir ?

Eh bien, c'est comme la liberté d'expression. Ce n'est pas parce que vous ne vous en servez pas aujourd'hui qu'elle vous est inutile pour demain... Il en va de même pour les sources. Tant qu'un logiciel fonctionne bien, tant qu'il remplit la mission pour laquelle vous le possédez, tout va bien. Vous n'avez besoin de rien ! Mais s'il faut changer de machine (un peu de puissance en plus) ou encore s'il y a un plantage, vous allez découvrir toute l'horreur des systèmes fermés. Je vous propose pour cela trois exemples simples et concrets :

- Vous disposez de musique achetée pour un baladeur numérique avec un système fermé de droits. Vous en êtes très content et vous ne voyez pas en quoi un système ouvert vous servirait. Sauf que ce système de droits n'est pas transposable ! C'est-à-dire que même si vous l'avez sauvegardé, vous ne pourrez pas réutiliser la musique et les droits associés, ceux-ci étant irrémédiablement liés à votre machine et l'installation d'un ou de plusieurs logiciels spécifiquement sur cette machine. A moins d'être hors-la-loi, la perte de votre baladeur numérique ou de votre disque dur signifie la perte de vos achats de musique. Je vous prie de me croire que vous n'allez pas apprécier de perdre votre argent, le plaisir d'écouter votre musique en plus de la perte sèche du lecteur ou du disque dur et du temps perdu que cela occasionne.
- Imaginez maintenant que le logiciel avec lequel vous produisez vos courriers ne soit plus maintenu et plus compatible avec ses suivants, y compris le système d'exploitation de votre machine. Vous devez alors conserver une machine fonctionnelle pour lire vos anciens courriers et une pour les nouveaux ou bien renoncer à vos anciens courriers au format numérique.
- De même, le logiciel que vous utilisez pour contrôler votre imprimante n'est pas compatible avec la dernière version de votre système d'exploitation. Si c'est un logiciel fermé et que votre imprimante ne dispose pas des outils adéquats pour le nouveau système, il ne vous reste qu'à changer d'imprimante !

Au travers de ces trois petits exemples, je veux vous sensibiliser au fait qu'un système fermé est dangereux pour votre production personnelle, mais aussi et surtout pour votre portefeuille ! Vous me demanderez alors en quoi les sources protègent vos données et vos finances ? Eh bien, elles renferment les méthodes de lecture de vos données en même temps qu'elles renferment les secrets de fabrication du logiciel. Et ces méthodes de lecture, tout comme ces secrets de fabrications sont lisibles par un grand nombre de gens, plus ou moins spécialistes, mais capables le cas échéant d'éviter des situations de pertes sèches de données ou de passage obligatoire par une case financière forcément douloureuse. Elles garantissent une pérennité de moyens sur une durée nettement supérieure à la demande boursière, puisqu'elles ne sont pas indexées sur la réussite ou l'échec d'une entité financière.

Ainsi, vous n'aurez sûrement jamais en temps qu'utilisateur lambda à compiler un logiciel. Il y a les paquets de votre distribution pour ça. Mais cela peut vous être utile de disposer ou de savoir que quelqu'un quelque part dispose des sources pour vous aider le cas échéant. En utilisant des Logiciels libres et en protégeant le droit de chacun à disposer des informations techniques sur les logiciels qui le servent, vous protégez vos droits d'utilisateur autant que de consommateur !

Et rien ne dit qu'un jour vous ne passerez pas du stade d'utilisateur à celui de producteur. Car, au fond, les sources sont à compiler ! C'est-à-dire transformer le code écrit par l'homme en code lisible par la machine. Cette compilation se fait via un ou des programmes spécifiques. Ce n'est généralement pas une tâche compliquée, mais je la déconseille sauf si nécessaire. Cela peut être le cas de temps à autre, mais cela reste exceptionnel voire rarissime...

Sidoine Pierrel

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Qui utilise GNU/Linux et les Logiciels libres ? Qui est assez fou (ou humaniste) pour écrire des programmes gratuits de qualité ?

Cette question, qui est fort pertinente, ouvre le champ à l'élaboration de listes à peu près infinies ou qui pourraient occuper toutes les pages du présent magazine à elles seules, en petite police... Alors je vous propose d'en faire un rapide tour d'horizon en tâchant de rester succinct.

### Des administrations utilisatrices

En France :

- > En janvier 2005, la Gendarmerie nationale généralise l'usage d'OpenOffice à 70 000 postes d'utilisateurs.
- > Le ministère de la Culture convertit 1 700 postes utilisateurs.
- > Dès 2007, plus de 400 000 postes informatiques de l'administration centrale française passeront de Microsoft Office à la suite bureautique OpenOffice.

Mais également bon nombre de mairies, comme celle de Tours (pour parler de ce dont je suis sûr), et quasiment toutes les universités de France. Un grand nombre de PME également, de manière spontanée, mais elles sont particulièrement difficiles à répertorier.

Et il est vrai qu'OpenOffice (suite bureautique), Firefox (navigateur web), Thunderbird (client de courriel), The Gimp (retouche d'images), VideoLan (lecteur vidéo), Gaim (messagerie instantanée), 7-zip (archivage/compression) constituent une offre bureautique fiable et complète. Ajoutez à cela des compilations intégrées sur CD permettant d'installer d'un clic toute cette collection sur n'importe quel poste de travail sous système d'exploitation propriétaire répandu, et vous comprendrez vite le choix des administrateurs système dans ce domaine (pour plus d'info, voir <http://free-eos.org/>).

À l'international, si l'on scrute rapidement la presse des six derniers mois :

- > Le Conseil des Ministres belge adoptait le format Open Document.
- > Le ministère de la Défense de Singapour choisissait OpenOffice.org.
- > La ville de Munich compte migrer 80% de son parc informatique vers GNU/Linux entre fin 2008 et mi 2009.
- > Juillet 2006 : le gouvernement de Croatie adopte une stratégie d'usage des Logiciels libres et souhaite développer l'usage des Logiciels libres au sein des institutions du gouvernement et des administrations.

Et nous reparlerons plus loin des universités américaines, dans la partie contribution.

### Des entreprises utilisatrices

En France, on trouve des sociétés comme Free ou Bull, qui se comportent en consommateurs passifs de Logiciels libres.

À l'international, de nombreuses sociétés arborent des étiquettes « *powered by Linux* » ou Apache ou BSD... Les lister ici ne serait pas particulièrement pertinent... Alors, passons tout de suite aux créateurs de Logiciels libres !

### Les institutions qui contribuent...

En France :

- > le CEA (Commissariat à l'Énergie Atomique) ;
- > le CNRS (Centre National de Recherche Scientifique) ;
- > l'INRIA (Institut National de Recherche en Informatique et Automatique) ;
- > l'École Centrale de Paris avec le projet VidéoLan Client ;
- > l'École des mines de Nancy (héberge des images ISO) ;
- > les régions Bretagne et PACA ;
- > les universités de Bretagne Occidentale, de Versailles, de Rennes (participation au projet Bureau Libre).

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Quelles sont les différentes licences libres ?

### L'horreur des licences propriétaires

Il existe une quantité phénoménale de licences logiciel de par le monde. Chaque éditeur de logiciel dispose généralement de la sienne, qu'il décline d'ailleurs souvent en plusieurs versions... Pourtant, on retrouve rapidement des constantes dans cette masse de contrats synonymes de perte de liberté.

Par exemple, les licences n'accordent à chaque fois qu'un vague droit d'utilisation des logiciels cités, sur un nombre limité de machines, dans un espace limité dans le temps et pour un nombre limité de personnes ; le plus souvent une seule. Ainsi, les éditeurs de logiciels propriétaires s'assurent que leurs utilisateurs payeront abondamment et régulièrement pour l'utilisation de leurs logiciels, tout en imaginant se prémunir contre la concurrence...

Les éditeurs intègrent d'ailleurs parfois des clauses plus singulières, comme le droit de publication par leurs soins des informations que vous traitez avec les logiciels pour lesquels vous achetez une licence (notamment en matière de messagerie instantanée), ou l'interdiction d'installer les logiciels concernés dans un pays sous blocus américain... Et on retrouve tout ceci d'un bloc et agrémenté de quantité d'autres aberrations dans le fameux MS-CLUF, le Contrat de Licence d'Utilisateur Final du leader des ventes de logiciels.

### La force d'une licence libre

Alors, après avoir vu les abus de ce système de licence, pourquoi donc vouloir entrer dans ce jeu ? Pour principalement deux raisons : la première fut de ne pas faire du monde des Logiciels libres une zone de non-droit pillable à souhait ; la seconde fut de retourner le mécanisme à l'avantage des développeurs libres.

À l'international :

- > la NASA ;
- > les universités de Berkeley (avec les systèmes d'exploitation BSD), Illinois, MIT ;
- > les fondations telles que la *Free Software Foundation*, *Jabber Software Foundation*, *Apache Foundation*, *Mozilla Foundation*...

### Les entreprises qui contribuent...

En France :

- > Mandriva (distribution GNU/Linux éponyme) ;
- > AdaCore (compilateur Ada 2005 GNAT intégré à GCC, environnement de développement multi-langages GPS et contributions actives à GCC lui-même) ;
- > les membres du réseau libre entreprise : <http://www.libre-entreprise.org/> ;
- > les membres du réseau Gitoyen : <http://www.gitoyen.net/> ;
- > et un nombre important et grandissant de SSSL (Société de Services en Logiciels Libres) plus modestes telles que : Néréide, Quantic et Nativ'Object, pour ne citer que les trois entreprises situées à moins de 15 minutes en voiture de chez moi...

À l'international, pour ne citer que les plus connues :

- > Red-Hat ;
- > LinSpire ;
- > Novell ;
- > IBM ;
- > Sun ;
- > HP ;
- > Intel ;
- > Apple ;
- > NetGear ;
- > RealNetworks ;
- > Trolltech.



Petite précision : bien sûr, toutes ces listes sont largement incomplètes, arbitrairement choisies et organisées ; elles ont pour seule prétention de donner une idée de l'importance et du volume de l'éco-système libre, et de son degré d'intégration dans les mœurs...

Toutefois, on constate que les Logiciels libres ne sont pas tous développés par des programmeurs du dimanche, seuls dans leur salon. Les Logiciels libres sont le fruit d'efforts importants, consentis par de nombreuses structures (associations géographiques comme les LUG, philosophiques comme l'APRIL ou l'AFUL, technologiques comme les Mongueurs de Perl ou Ada-France, entreprises, entités nationales, régionales...) et leur fiabilité ainsi que leur adaptabilité n'est plus à prouver.

Heureusement, il reste encore une grande place pour l'esprit communautaire dans tout ça, et de simples projets d'hier grandissent encore aujourd'hui pour peut-être devenir les logiciels phares de demain.

Simon Descarpentries



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

Ainsi, en 1989, lorsque le besoin de disposer d'une licence garantissant les libertés voulues pour les logiciels développés par la FSF se fit sentir, Richard M. Stallman, initiateur de projet GNU, plancha avec toute une équipe de juristes et de co-développeurs sur la réalisation d'une licence permettant de protéger les premiers logiciels annoncés comme « libres ». Cette licence devait finalement garantir 4 libertés fondamentales aux logiciels qui lui seraient confiés :

- la liberté d'exécuter le logiciel, pour tous les usages ;
- la liberté d'étudier le programme et de l'adapter à ses besoins, ce qui implique l'accès au code source du logiciel ;
- la liberté de redistribuer autant de copies que souhaité du logiciel ;
- la liberté de rendre publiques ses améliorations et ainsi d'en faire profiter l'ensemble de la communauté.

La licence publique générale (ou GPL) était née. Cette licence eut un franc succès dans la communauté naissante du Logiciel libre, et gagna ses lettres de noblesse en permettant la victoire de développeurs libres dans plusieurs procès les opposant à des entreprises peu scrupuleuses. (Plus d'info sur <http://www.gpl-violations.org/>) Et c'est pourquoi la GPL protège aujourd'hui 60 à 70% des Logiciels libres.



### La variété des licences libres

Toutefois, et comme il y a toujours autant de points de vue sur une question que de personnes pour y réfléchir, d'autres licences se réclamant libres ou seulement « *open source* » (à sources ouvertes) virent le jour. Or, malgré leurs appellations, elles ne garantissaient pas toutes les quatre libertés citées ci-dessus... C'est pourquoi, l'OSI (*Open Source Initiative*) décida de faire un classement des licences et de publier la liste des licences qu'elle reconnaissait être libres, au sens GPL du terme (voir page 34).

Cette liste compte aujourd'hui près de soixante licences ! Et on retrouve parmi les plus connues : la GNU/GPL (et ses dérivés : LGPL, GFDL...), mais également la CeCILL (conçue par le CEA, le CNRS et l'INRIA pour s'intégrer au droit français avant que la GPL ne soit traduite), la licence *Creative Common Share Alike* (sorte de gadget graphique ayant le mérite d'être compréhensible par le plus grand nombre), la licence Apache (initiée par la fondation du même nom), la licence publique Mozilla, la licence PHP, la licence Python, les licences BSD, la licence Nokia, Sun, RealNetworks...

Toutes ces licences étant compatibles avec la définition du libre reconnue par l'OSI, leurs raisons d'être est d'intégrer des spécificités souhaitées par leur principal développeur, jugées utiles dans le domaine particulier où il opère, et également d'introduire des degrés plus ou moins prononcés de « *copyleft attitude* », c'est-à-dire de contamination par le libre des logiciels dérivés ou utilisant des fragments de code protégé par licence libre. Ces degrés vont d'une contamination totale dans le cas de la licence GPL (c'est-à-dire que pour pouvoir utiliser une ligne de code protégé par la GPL, votre future production se devra d'être, elle aussi, entièrement protégée par la GPL), à une contamination nulle dans le cas des licences de type BSD, qui permettent à n'importe quelle entreprise d'intégrer un logiciel comme OpenSSH dans leurs productions propriétaires. Les deux approches se défendent à la fois d'un point de vue philosophique et économique...

Simon Descarpentries

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Certaines distributions sont payantes, est-ce normal ? Qu'ont-elles de plus ?

Si vous avez lu la définition d'un Logiciel libre (p. 5) et la définition d'une distribution (p. 36), vous avez dû remarquer que l'on ne parle jamais d'argent. Et pour cause ! Ce n'est pas un critère dans le cadre du Logiciel libre. Si vous préférez, tant que vous respectez le droit d'une personne à diffuser, modifier, etc., le logiciel que vous vendez, c'est toujours un Logiciel libre même s'il est payant. D'ailleurs, des sociétés comme InfoSIAL en Espagne développent des Logiciels libres et vendent certains modules (qui sont sous GPL d'ailleurs).

Sauf que dans le cas d'une distribution, d'autres éléments interviennent. Ainsi, il existe des distributions payantes dans trois domaines : les serveurs, l'embarqué et les stations de travail. Vous ne pouvez pas les télécharger sur Internet légalement sans avoir payé un abonnement. Pour certaines, il existe des versions d'évaluation (limitées de 30 à 90 jours). Comment se fait-il ? Ce sont bien des Logiciels libres, donc normalement, on devrait pouvoir y accéder quoi qu'il advienne... Oui et non. Pour les serveurs et les stations, dans ce cas précis, les auteurs de cette (ces) distribution(s) ont joué non seulement sur les mots, mais aussi sur les moyens : vous n'achetez pas une *distribution*, mais un *service* (la mise à jour et parfois l'assistance téléphonique, mais c'est souvent une option). Et la distribution n'est disponible que pour les clients du service. De fait, la distribution n'est pas en libre-service...

En revanche, les sources de la distribution sont toujours librement fournis. Et donc, il est possible de disposer d'une distribution entièrement libre qui soit compatible avec la distribution payante (merci la GPL !). C'est la cas de *CentOS* (<http://www.centos.org>) ou *WhiteBox* (<http://www.whitebox.org>) avec *Red Hat Advanced Server* (<http://www.redhat.com>) dans le domaine des serveurs. La RHAS n'est pas disponible, mais sa concurrente, *Novell SUSE*, propose une version d'évaluation (c'est-à-dire que la mise à jour est limitée dans le temps) disponible sur leur site (<http://www.novell.com/products/server/>). De même, en termes de station, une distribution emblématique de ce modèle est *Xandros* (<http://www.xandros.com>). *Mandriva* (<http://www.mandriva.com>) est restée sur un modèle un peu différent, puisque seul le club des utilisateurs est payant ; or, le club n'est qu'une avant-première des développements et des paquets GPL (ainsi que quelques paquets non libres).

Concernant l'embarqué, c'est quelque peu différent. Ici, de nouveau, on ne paie pas à proprement dire la distribution, mais des portions de codes, spécifiques à l'embarqué et aux cartes qui le composent et qui ne sont pas disponibles librement. Un particulier lambda n'aura pas besoin de ça. Par contre, les férus d'électronique sont régulièrement frustrés !

Alors, vous me direz, quel intérêt ? Concernant les serveurs, ces distributions sont rares et elles visent un public très ciblé : les grandes et moyennes entreprises. Si vous n'utilisez pas Oracle ou DB2, si vous n'êtes pas client SAP, Tivoli ou Lotus, probablement aucun intérêt pour vous. Par contre, si vous utilisez ces logiciels, il y a de fortes chances que vous utilisiez une distribution payante. De même, si vous avez dans vos bureaux ou votre salon, un iSeries ou un pSeries d'IBM, il y a aussi de fortes chances que vous utilisiez une distribution de ce type. Pourquoi ? Parce que les distributions payantes ont été certifiées par les constructeurs et les vendeurs de logiciels. C'est-à-dire que ces derniers « garantissent » la qualité et le fonctionnement (théorique) de leurs produits avec cette version de Linux et des outils GNU associés. De plus, ils fournissent les paquets et les informations d'installation exclusivement pour ces distributions. Dans ce cas précis, ce système permet de disposer d'un suivi industriel du logiciel et d'une assistance garantie (y compris juridiquement) de la part de tous les acteurs de la chaîne de production. Même si tout cela est fortement théorique (l'expérience prouve que ce n'est pas le contrat qui fait la qualité), il y a là néanmoins des garanties importantes pour une société commerciale ou des utilisateurs professionnels en milieu de production. Ce sont eux les cibles des distributions payantes.

Concernant l'embarqué, il est parfois impossible de faire tourner le matériel sur Linux sans une distribution spécifique et payante. On est ici directement lié à la politique du fabricant et à ses accords commerciaux. De plus, on dispose alors d'une assistance et d'un savoir-faire réel qui permet de réduire les coûts de développement de l'application embarquée. L'investissement est cependant à ne pas sous-estimer. Une distribution d'exemple est *Montavista* (<http://www.mvista.com>).

Au sujet des stations, c'est peut être là que c'est le plus surprenant de rencontrer des distributions payantes tant l'offre libre est devenue foisonnante et de qualité. Mais ici, comme pour les serveurs, on a rendu payant les mises à jour et le téléchargement d'applications. Force est de constater qu'on paie aussi une qualité des outils graphiques de gestion des paquets et de la distribution. On s'adresse sans détour aux personnes qui ne font qu'un usage non technique de Linux. Cependant, on peut douter de la pertinence d'une telle offre, le grand public à la différence du public commercial cité ci-dessus n'étant captif ni à la qualité des interfaces et des logiciels, ni au matériel. Ensuite, c'est une question de goût et de soutien financier (voir le modèle de club de Mandriva) autant que de rapport qualité/prix (Xandros a des fans qui ne démontent pas de la qualité de la distribution et pour qui le prix est clairement en adéquation).

*La liste des distributions évoquées ici n'est pas exhaustive et ne vise qu'à présenter des exemples. Il ne s'agit en aucun cas de faire une quelconque promotion commerciale de ces outils et de leurs auteurs respectifs.*

Sidoine Pierrel

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Je ne sais pas programmer, puis-je aider des projets de Logiciels libres ?

Absolument ! La programmation n'est que l'un des aspects d'un logiciel et il n'est absolument pas nécessaire d'être programmeur ou informaticien pour avoir son nom sur le tableau d'honneur. Voici différentes manières de contribuer à un logiciel.

### La traduction

Vous l'avez remarqué, GNU/Linux est international. Certains projets étant traduits en plusieurs dizaines de langues, le travail de traduction est loin d'être négligeable. Ainsi, sur les 800 contributeurs référencés du projet KDE, environ 300 sont des traducteurs !

La manière de gérer la traduction est laissée au choix de chaque projet, mais la plupart utilisent **Gettext**, qui permet d'isoler toutes les chaînes de caractères d'un programme dans un fichier de modèle (*template*). Celui-ci est alors distribué aux traducteurs qui le complètent avec la traduction du logiciel. Ainsi, les traducteurs n'ont à s'occuper d'aucun détail technique. Divers logiciels permettent de modifier les fichiers `.po`, notamment le très bon **Kbabel**.

Sous Ubuntu, la plupart des applications disposent d'une entrée « *Translate this application* » dans le menu **Aide**. En l'utilisant, vous pourrez traduire les chaînes de caractères du logiciel directement à partir d'une interface Web.

### Les rapports de bug

Vous avez trouvé ce que vous pensez être un bug dans un programme ? Celui-ci vous irrite ? Vous pouvez faire en sorte qu'il soit réparé rapidement et contribuer au développement du programme. Les projets imposants disposent généralement d'une interface Web de rapport de bug (Bugzilla, Sourceforge, Launchpad, etc.) que vous pouvez utiliser pour chercher si votre bug a déjà été signalé, et confirmer son existence ou créer le rapport initial.

Si vous êtes fan d'un logiciel particulier, il est également très apprécié de participer au test des versions bêta (et si vous êtes aventureux, vous pouvez le faire avec votre distribution favorite !). Celles-ci peuvent souvent coexister avec une version stable. Vous aurez alors l'occasion d'influer directement sur le développement en rapportant les dysfonctionnements ou en suggérant des fonctionnalités ou des améliorations.

Le rapport de bug est tout un art. Pour plus d'informations à ce sujet, consultez le document suivant :

<http://www.chiark.greenend.org.uk/~sgtatham/bugs-fr.html>

### La documentation

Vous êtes passé expert dans l'utilisation d'un logiciel à la sueur de votre front, en passant des heures à chercher comment il fonctionnait et en pestant contre l'absence de documentation ? Félicitations, vous êtes à présent la personne la mieux placée pour le documenter ! Pensez à constituer une documentation qui permettra à d'autres de ne pas déboiser le chemin à nouveau. Son format dépend du projet, et, là encore, il est bon de demander l'avis des développeurs.

Tout comme le logiciel, la documentation se traduit. Si vous constatez que celle-ci est existante, mais uniquement disponible dans la langue de Shakespeare, vous pouvez contribuer à une version traduite pour les anglophobes...

### L'art

Le code et la documentation sont effectivement la partie principale d'un logiciel... mais pas forcément le tout. Les icônes, les sons, bref, l'art en général en fait partie intégrante.

Les jeux sont particulièrement demandeurs d'art et laissent le plus de liberté d'expression aux artistes. La qualité artistique fait souvent la différence entre un bon et un mauvais jeu, et c'est dans ce domaine que les artistes peuvent avoir le plus de poids.

### La donation

La création et la maintenance d'un Logiciel libre ne rapportent généralement rien à ceux qui y contribuent, mais ont très certainement un coût. Outre le temps passé, qui aurait pu l'être sur des activités plus lucratives, les coûts en termes de bande passante pour le serveur, et parfois de voyages pour organiser des rencontres entre développeurs nécessaires pour les gros projets sont bien réels.



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>



La plupart des sites de projets libres comportent un bouton par lequel vous pouvez simplement effectuer une petite donation, via *Paypal* ou tout autre système de paiement sécurisé. Celle-ci n'a pas besoin d'être élevée : 1000 utilisateurs qui donnent chacun 3 euros ont plus d'impact qu'un seul qui en donne 500. Elle sert juste à montrer que vous utilisez le logiciel qui vous est offert et que vous appréciez les efforts faits par son ou ses développeurs.

Pour un développeur de Logiciels libres, recevoir une contribution monétaire, même modeste, est une marque de reconnaissance pour son travail, qui encourage à l'améliorer en tenant compte des remarques qui sont généralement faites avec la donation. Les requêtes de fonctionnalités des utilisateurs qui contribuent sont logiquement prises en compte avec plus d'enthousiasme que celles de ceux qui ne font que réclamer.

Certains projets acceptent également les dons en matériel informatique, nécessaire pour faire tourner les serveurs dont ils ont besoin.

Alexandre Courbot



## Est-ce que GNU/Linux est le seul système d'exploitation libre ?

Si GNU/Linux est l'un des OS les plus connus du monde du Logiciel libre, il n'est de loin pas le seul de sa catégorie. D'autant que l'idée de faire des Logiciels libres est particulièrement à la mode et qu'elle a pu séduire certains industriels traditionnellement liés au monde du logiciel propriétaire. Il faudrait probablement un magazine entier pour comptabiliser l'ensemble des systèmes d'exploitations libres qui existent aujourd'hui ! Cette diversité tient notamment au fait que nombre d'OS libres, comme Linux, servent de base à d'autres projets d'OS libres. A lui seul, le noyau Linux a donné lieu à une cascade de projets (toutes les distributions que l'on connaît bien, mais également une multitude de micro-OS destinés en général à un usage très particulier, comme des routeurs, des *firewalls*, etc.). Et Linux n'est d'ailleurs pas le seul à avoir été l'objet d'un tel succès ; si l'on regarde du côté des systèmes BSD, la constatation est identique.

Tous les OS libres n'ont cependant pas la même vision du Libre que le monde Linux. Les libertés ne sont pas exactement les mêmes selon que le système est sous licence GPL, ou sous licence BSD (traditionnellement plus permissive que la GPL), ou que l'OS est le dérivé libre d'un OS commercial (ex. : OpenSolaris et leur licence CDDL sur mesure). Comme toujours, dans la jungle des licences libres, attention à l'usage que vous allez faire de l'OS, vérifiez ce que la licence vous permet de faire !

On peut tenter très globalement de classer les OS libres en grandes familles : la famille GNU/Linux, les systèmes BSD (FreeBSD, OpenBSD, NetBSD...), puis globalement tout le reste, avec quelques mentions spéciales pour FreeDOS, les dérivés de BeOS, OpenSolaris, etc. Comme on l'a dit, pratiquement tous les OS libres ont des dérivés ; vu qu'ils sont libres, il ne se passe en général pas très longtemps avant que certains réutilisent l'OS de base pour monter un nouveau projet autour.

### S'il ne fallait en essayer qu'un ?

Venant du monde Linux, les premiers OS libres à essayer sont peut-être les systèmes BSD (les plus célèbres : FreeBSD – <http://www.freebsd.org>, OpenBSD – [openbsd.org](http://openbsd.org), NetBSD – [netbsd.org](http://netbsd.org)), qui ont l'énorme avantage d'appartenir au monde UNIX. Parmi toutes leurs qualités, les utilisateurs Linux apprécieront particulièrement leur centralisation : finies les recherches interminables sur Internet pour savoir comment mettre à jour ses *packages*, ou pour savoir comment démarrer tel ou tel service ; pas besoin d'aller chercher les *how-to* correspondants, tout est dit dans le manuel (FreeBSD) ou dans la FAQ (OpenBSD), disponibles sur le site. En pratique, c'est un énorme avantage par rapport à Linux à qui l'on peut reprocher parfois la dispersion des projets (mais c'est également ce qui fait sa force).

Au-delà de ça, chaque OS a ses particularités : si FreeBSD est un système « généraliste », à la fois capable de servir de base à un *cluster*, et de station de travail, OpenBSD est, lui, plutôt dédié à la sécurité et NetBSD est fortement orienté multiplateforme (ça tourne sous Amiga !). Les trois sont d'une solidité et d'une robustesse à toute épreuve (**Yahoo.com** tourne sous FreeBSD...) et valent vraiment le coup d'être essayés.

Vous souhaitez créer votre propre OS libre ? A condition de ne pas être rebuté par la technique, *Linux From Scratch* (LFS, <http://www.linuxfromscratch.org/>) sera sans doute un point de référence. Même s'il existe de la documentation détaillant un peu les mêmes procédures pour des systèmes BSD, LFS est sans doute le projet le plus abouti. Il regroupe une série de documents détaillant toutes les étapes pour construire votre propre OS, partant des sources de Linux. Un *must* pour toute personne désireuse de se constituer son propre système.

Après tout, qui sait, la prochaine distribution Linux portera peut-être votre nom ?

Lucie Robin

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

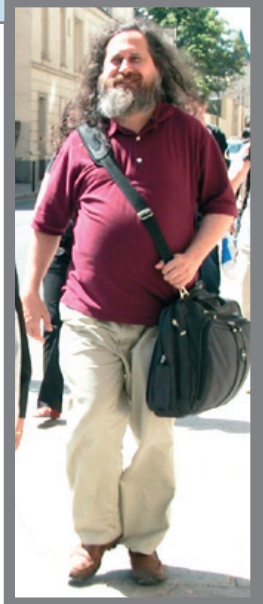
<http://www.editions-diamond.fr>

## Qui est Richard M. Stallman ?

Richard Matthew Stallman est l'initiateur du projet GNU, de la *Free Software Foundation* et de la licence publique générale (GPL) ayant servi de fondement au développement des premiers Logiciels libres tels que nous les connaissons aujourd'hui.

Présenté parfois comme un idéaliste chevronné aux convictions inébranlables, il est intéressant de se pencher un peu sur le parcours du personnage. Né à Manhattan en 1953 (il a donc 53 ans cette année), il obtient son « baccalauréat » en 1969 à New York, travaille un peu chez IBM, puis entame des études de physique à l'Université d'Harvard, pour arriver au département d'intelligence artificielle du fameux *Massachusetts Institute of Technology* ou MIT. Là, il rejoint une joyeuse communauté de *hackers* (au sens noble du terme, comme on le trouve défini ici : <http://fr.wikipedia.org/wiki/Hacker>) aux mœurs étranges. En effet, ces spécialistes de la programmation de puces, virtuoses et inventifs, tatillonnent sur la sécurité de leurs programmes, partagent librement leurs codes entre eux, afin de permettre à chacun de relire le code des autres, d'y trouver des erreurs, des failles ou des inélégances. Malheureusement, ces programmes n'étaient pas portables, et lorsque la production des puces en question (Digital PDP-10) fut stoppée, la communauté de hackers où RMS se sentait si épanoui se disloqua (comme la communauté de l'anneau...) devant l'adversité et le pouvoir malsain de l'appât du gain, conduisant les gens à garder secrets leurs demi-éclairs de génie codés, pour tenter d'en vivre. Toujours est-il que RMS ne trouva aucune alternative existante à la communauté où il avait appris à coder efficacement et proprement...

Source photo : Wikimedia Commons - Auteur photo : Jérémie Zimmermann



Source photo : StopDRM.info

### La naissance du phénomène...

C'est précisément là que le personnage s'élève. Puisant dans ses convictions toute la pugnacité nécessaire, il prend le taureau par les cornes et annonce en 1983 le lancement du projet GNU visant à développer un système d'exploitation portable et « libre ».

En 1984, il crée la FSF afin de donner une structure légale à son mouvement.

En 1985, il publie le manifeste GNU, dans lequel il explicite ses objectifs et motivations.

En 1989, la première version de la licence publique générale (GPL) est publiée, donnant une dimension juridique concrète à la notion de « Logiciel libre ». S'en suivent des années de développement rigoureux, qui font sortir petit à petit de l'ombre et de la marginalité le projet.

Il est encore bon de noter qu'en 1999, RMS pose les bases philosophiques et motive la création du projet d'encyclopédie libre dont résulte Wikipédia (<http://fr.wikipedia.org/wiki/Accueil>), avec le succès qu'on lui connaît aujourd'hui.

Enfin, dernier fait marquant pour nous, Français, le 9 juin 2006, Richard Stallman se présente devant l'hôtel de Matignon dans le but de remettre en main propre au Premier ministre une version imprimée de la cyber pétition française contre la loi DADVISI, matérialisée par une feuille de papier comportant plus de 165 000 signatures et parrainée par le collectif StopDRM. Il ne sera tout simplement pas autorisé à entrer et la grande feuille finira dans le caniveau devant le bâtiment clos. Selon le chef de la sécurité de M. De Villepin, il s'agissait d'une décision « mûrement réfléchie ». Il est peut-être aussi possible d'y voir un certain déni de démocratie. (La pétition, la suite du combat : <http://eucd.info/petitions/index.php?petition=2> et <http://stopdrm.info/>)

En parcourant en grosses lignes plus de 20 ans d'action, on constate à tout le moins la constance du personnage. Et si RMS est parfois caricaturé sur ses positions « extrêmes », c'est bien qu'il n'a pas fléchi sa détermination sur la libéralisation des propriétés intellectuelles telles qu'algorithmes, documentations et graphismes. RMS est de tous les combats dans ce domaine, et le dragon ne s'est jamais endormi depuis les premières flammes de la FSF. Il suffit de jeter un coup d'œil sur son site perso pour s'en persuader (<http://www.stallman.org/>).

Un génie ? Le mot est peut-être trop fort, mais en tout cas, il ne serait pas le premier génie en avance de phase sur son époque. Après tout, c'est à vous d'en juger : quatre doctorats honorifiques, des prix et des professorats dans plusieurs pays (il parle couramment le français et possède de solides notions d'espagnol ou d'indonésien). Il a écrit les premières versions du compilateur C de GCC, du débogueur GDB, de l'éditeur GNU emacs, et j'en passe.

Simon Descarpentries

● Pour en savoir plus : <http://www.gnu.org/gnu/thegnuproject.html> et [http://fr.wikipedia.org/wiki/Richard\\_Stallman](http://fr.wikipedia.org/wiki/Richard_Stallman)

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Pourquoi utiliser les Logiciels libres dans l'éducation ?

L'école doit permettre à ses élèves de vivre pleinement leur vie de citoyens à leur sortie. C'est pour cela que l'informatique y a toute sa place. Depuis 2000, un Brevet Informatique et Internet (B2I) valide les compétences acquises dans ce domaine par les élèves de primaire, collège et, depuis peu, ceux du lycée. Il s'agit pour l'élève d'acquérir de l'autonomie vis à vis des technologies de l'information et de la communication (TIC).

L'école de la République se doit de respecter trois principes fondamentaux : liberté, égalité et fraternité. En utilisant les Logiciels libres, l'école est au cœur de ces principes avec :

----> La liberté d'utilisation. Certains logiciels propriétaires sont gratuits pour un usage non commercial, d'autres proposent une licence à tarif réduit à condition de ne les utiliser que pour l'enseignement (ce qui peut empêcher des enseignants chercheurs de les utiliser pour leurs activités de recherche), etc. Avec les Logiciels libres, ces limitations n'existent pas, la liberté d'utilisation est totale : en classe, à la maison, plus tard en entreprise...

----> L'égalité des chances pour tous ceux qui possèdent un ordinateur chez eux grâce au partage légal de logiciels ; un enseignant peut donner une copie d'un Logiciel libre à tous ses élèves. De cette façon, les élèves issus d'un milieu aisé ne seront plus les seuls à pouvoir accéder aux logiciels utilisés en classe.

----> La fraternité, d'une part parce que l'entraide et la coopération sont à la base du Logiciel libre, d'autre part parce que les élèves peuvent s'échanger des logiciels.

C'est pourquoi en octobre 1998, le ministère de l'Éducation Nationale a signé un accord avec l'Association Francophone des Utilisateurs de Linux et des Logiciels libres (AFUL) portant sur la promotion et le déploiement des Logiciels libres au sein de l'éducation (voir <http://www.aful.org/education>). L'AFUL a pour mission d'aider la mise en réseau « clef en main » des collèges et lycées ainsi qu'à titre expérimental pour les écoles. L'Éducation Nationale pourra aussi faire appel à l'AFUL pour la formation de son personnel sur l'utilisation des ressources libres.

En 2004, une proposition de mise à jour de cet accord a été prononcée. Ainsi, les organisations syndicales ont particulièrement insisté sur la formation que peut apporter l'AFUL. Celle-ci porterait principalement sur des notions informatiques et non sur des logiciels particuliers. Une campagne sur des questions de fond (telle que le titre de cet article) a été menée.

Il paraît bon de noter, également, que la plupart des Logiciels libres sont gratuits et, pour beaucoup d'établissements scolaires, ce n'est pas un détail. Sachant que le piratage et l'utilisation des copies de logiciels propriétaires sont des pratiques illégales, l'Éducation Nationale se doit de montrer l'exemple en matière de respect des licences. En utilisant des Logiciels libres, les enseignants s'intègrent complètement dans cette démarche et peuvent alors tester des outils pédagogiques sans la contrainte d'un achat préalable.

Cependant malgré tous ces avantages, un inconvénient perdure. En effet, la grande majorité des ordinateurs personnels sont fournis avec le système d'exploitation Windows ; ceux-ci étant dotés de nombreux logiciels propriétaires, les utilisateurs ont acquis au fil du temps une aisance d'utilisation non négligeable les concernant. L'école n'a pas échappé à ce phénomène. De ce fait, les enseignants conservent leurs habitudes en utilisant la plupart du temps des logiciels propriétaires (certainement par manque d'intérêt ou de formation pour l'outil informatique ou tout simplement par manque d'information sur le Logiciel libre). Cependant, ne soyons pas trop pessimiste. Notons que la suite bureautique OpenOffice.org est très largement répandue (ce qui est déjà un très grand pas) et que de nombreux logiciels propriétaires trouvent leur équivalent dans la panoplie de Logiciels libres qui fonctionnent très bien sous Windows. Il y a également bon nombre d'enseignants et inspecteurs de l'Éducation Nationale très motivés et conscients des bienfaits du Libre.

Peggy Delarbre

- Pour aller plus loin : <http://www.epi.asso.fr/revue/96/b96p171.htm>

Logo du projet SkoleLinux



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Quels Logiciels libres puis-je utiliser avec mes enfants ?

Vous souhaitez que votre enfant développe des compétences grâce à l'outil informatique, sans tomber dans le domaine de l'enseignement ? Le monde du Logiciel libre compte de nombreux logiciels ludo-éducatifs. La méthode la plus simple pour en tester un grand nombre est d'utiliser **Freeduc**.

La distribution **FREEDUC-primaire** (1.5) est un des nombreux dérivés de Knoppix/Debian. Elle est distribuée par OFSET (*Organization for Free Software in Education and Teaching*, <http://www.ofset.org/>) en partenariat avec le CRDP Aquitaine et les membres enseignants d'ABULEDU (<http://www.abuledu.org/>). Il s'agit d'un ensemble de Logiciels libres ludo-éducatifs destiné à des enfants de maternelle et de primaire. Sous forme de *live-CD*, il ne nécessite pas d'installation. Son utilisation est donc rapide et simple pour les parents comme pour les enseignants.

Ce live-CD propose de nombreuses applications comme :

- > des logiciels de mathématiques ;
- > des applications scientifiques (SunClock, Celestia...) ;
- > des applications audio (entraînement au solfège...) ;
- > des outils technologiques (dessin...) ;
- > des logiciels de géographie (Xrmap...) ;
- > des applications linguistiques (dictionnaires...) ;
- > des applications ludo-éducatives (détaillées ci-après) ;



edubuntu

Pour vous aider dans l'utilisation et la découverte de ce live-CD, vous trouverez en ligne sur le site de l'OFSET, un manuel complet (« Le livre de Freeduc-cd ») détaillant toutes les applications du live-CD de la version générale 1.4 allant de la maternelle au lycée.

Parmi les applications qui nous ont le plus séduite :

---> L'homme patate : Il s'agit de déguiser une patate en homme. Votre enfant (de niveau maternelle) apprendra où placer les yeux, les oreilles, les cheveux... Et il pourra ainsi développer ses connaissances sur le schéma corporel (espace et vocabulaire). D'autres aires de jeux sont disponibles comme le déguisement d'un pingouin ou le remplissage d'un aquarium.

---> Gcompris est destiné aux enfants de 3 à 11 ans. Il comprend plusieurs activités de tous domaines tels que les mathématiques, le français, la logique ainsi que l'apprentissage de l'ordinateur avec la manipulation de la souris et l'utilisation du clavier.

Les activités de lecture pour les enfants de 6 à 8 ans sont principalement ciblées sur la reconnaissance visuelle, des associations mots/images et mots à trou. Ce sont des exercices répétitifs qui prennent une dimension ludique grâce à l'outil informatique (couleur, dynamisme de pages, son...)

Pour les enfants de 9 à 11 ans, Gcompris propose une activité de géographie. L'enfant doit faire glisser des éléments pour construire une carte. Les cartes disponibles sont les continents, l'Amérique du Nord, l'Amérique du Sud, l'Europe de l'Ouest et les régions françaises.

---> Entraînement au calcul mental : les opérations tombent du haut de l'écran. Il faut trouver le résultat avant qu'il ne soit trop tard. La pluie de nombres peut porter sur une ou plusieurs opérations en même temps. Ces dernières peuvent être des additions simples ou non, soustractions, divisions, du calcul approché, du calcul en arbre... Le nombre de calcul réussi est comptabilisé et un résumé de travail contenant les réussites et échecs de l'enfant est proposé (il est imprimable).

Pour finir, on peut également citer la distribution **Edubuntu** (dont la version 6.10 est disponible depuis fin octobre) (<http://www.edubuntu.org/>). Dérivée de la célèbre distribution Ubuntu, Edubuntu est clairement orientée éducation et pédagogie ; outre les applications bureautiques de base, elle comprend la suite **KDE Edutainment** (qui inclut de nombreuses applications de science, de mathématique et de géométrie, d'apprentissage des langues étrangères, etc.), **Gcompris** (cité plus haut), les logiciels **Tux4Kids** (**Tuxpaint** pour le dessin, **TuxMaths** pour le calcul et **TuxTyping** pour l'apprentissage de la frappe au clavier), le **Schooltool Calendar** (un calendrier spécialement conçu pour les structures éducatives) et bien d'autres encore. Edubuntu peut, au choix, être installée sur votre disque dur ou être utilisée comme un live-CD.

Peggy Delarbre

● Un ensemble de logiciels ludiques proposés par le Centre National de Documentation : [http://cndpl11.hosting.cri74.org/article.php3?id\\_article=80](http://cndpl11.hosting.cri74.org/article.php3?id_article=80)

● Le site du pôle de compétences en Logiciels libres du réseau SCEREN : <http://logiciels-libres-cndp.ac-versailles.fr/>

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Doit-on dire Linux ou GNU/Linux ? Qu'est-ce que Linux ?

Au sens strict du terme, « Linux » ne désigne que le noyau, un bout de code de la toute première importance, puisqu'il gère entre autres la mémoire vive et les accès au matériel. Son fichier exact se trouve dans le répertoire `/boot`, et s'appelle `vmlinuz-2.x.x-versionde-compilation`. Par exemple, sur notre Mandriva 2007, il correspond à `vmlinuz-2.6.17-5mdv` et pèse 1,8 Mo. Mais les commandes de base qui gravitent autour de lui, comme la majeure partie de celles que nous pouvons taper dans un terminal, ainsi que ses outils de développement et de compilation, sont issus du projet GNU, lancé en 1985 par le père de Logiciel libre, Richard M. Stallman. GNU devait être un système d'exploitation complet, mais son noyau, Hurd, se fait toujours attendre... Quelque peu vexé que Linux se greffe ainsi sur son projet et lui vole la vedette, Richard Stallman demanda donc que l'on prenne l'habitude de faire référence à notre système sous le vocable « GNU/Linux ». Celui que l'on surnomme parfois RMS est certes une personnalité très respectée dans la communauté du

Logiciel libre, mais cette appellation n'a rien d'officiel, et n'est donc nullement obligatoire, d'autant plus qu'on peut légitimement en trouver la graphie horrible, et qu'elle est assez obscure à expliquer aux débutants. C'est pourquoi beaucoup, dont l'auteur de ces lignes, rechignent à l'employer. N'oublions pas enfin l'existence en français d'une tournure de style, la synecdoque, qui consiste à nommer la partie pour désigner le tout. De la même façon qu'un « fer » ou une « lame » peuvent désigner une épée, Linux peut tout à fait désigner l'ensemble du système, soit les outils GNU, mais également tous les autres issus des autres projets. Un système Linux est de toute façon un édifice composé de beaucoup de briques différentes. À ménager la susceptibilité de tout le monde, on se retrouverait vite à devoir prononcer « GNU/Linux/GTK+/QT/wxWidgets/tk/XFCE » en fonction de ce qu'on utilise. Mais il est vrai que les outils GNU sont indispensables au bon fonctionnement du noyau Linux, alors que les autres restent optionnels...

Jean-Noël Pierron



## Où trouver de l'actualité ?

Si l'on avait dû attendre que les médias traditionnels parlent de nous, vous ne seriez pas en train de lire ce magazine... Heureusement, l'entreprenante communauté du Logiciel libre a su combler les manques en la matière, par le biais de magazines spécialisés par exemple.

D'ailleurs, acheter ce genre de publications rédigées par des passionnés est un excellent moyen de se tenir informé des avancées de la communauté, mais ce n'est pas le seul. Pour de l'info en continu, 24h/24h, il vous suffit de noter quelques bonnes adresses et de consulter vos lecteurs de flux RSS.

L'un des sites de référence en matière d'actualité « libriste » est incontestablement :

→ <http://linuxfr.org/>

Mais il n'est pas le seul. Il faut aussi compter sur :

→ <http://www.toolinux.com/>

→ <http://www.unixtech.be/>

→ <http://www.labo-linux.org/>

Enfin, le plus confortable pour en consulter plein d'un coup, c'est encore d'aller sur un site comme :

→ <http://www.tootella.org/>

Ce dernier regroupe sur une seule page les flux RSS d'une pléthore de sites d'information dédiés au libre. Les sites y sont classés par catégorie, n'hésitez pas à en consulter plusieurs !



Simon Descarpentries



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Où trouver des logiciels supplémentaires ?

Les distributions GNU/Linux viennent toujours avec un certain lot de logiciels pré-installés (c'est d'ailleurs un des éléments qui permet de les différencier entre elles), mais il est possible qu'un logiciel vous manque, ou que vous ne sachiez lequel utiliser... Alors, où trouver les Logiciels libres ?

### Installation

Il faut tout d'abord savoir qu'un logiciel, dans une distribution GNU/Linux, n'a généralement pas besoin d'être installé à partir d'un fichier d'installation exécutable comme il est courant de le voir pour d'autres systèmes d'exploitation. En effet, la grande majorité des distributions GNU/Linux offrent un système intégré de gestion des applications. Une installation de logiciel pour une distribution GNU/Linux devient alors :

- Chercher le nom du logiciel qui correspond à votre besoin ;
- Le retrouver dans la liste des paquetages disponibles pour votre distribution ;
- Signifier à l'outil de gestion des paquetages qu'on désire installer sa trouvaille.

À cette étape, un certain nombre de tâches automatisées s'exécutent (comme la récupération du paquetage à partir d'un dépôt sur internet ou à partir d'un CD, le dépaquetage du logiciel, la copie de ses fichiers pour installation), et, lorsque ces dernières se terminent, le logiciel est disponible sur votre machine. Un petit `$ man nom_de_logiciel` et vous voilà bien parti !

### Découverte

Bien, mais dans les étapes précédemment citées, il en est une qui ne s'invente pas : trouver le nom du logiciel qui correspond à votre besoin.

Heureusement, dans la joyeuse communauté du Logiciel libre, on a pensé à tout. Ainsi, les logiciels disponibles pour votre distribution peuvent-ils être affichés par catégorie dans votre gestionnaire de paquetage afin de simplifier vos recherches. Mais il y a plus simple ! En effet, si vous avez déjà fait vos premières dents en informatique sur un autre système d'exploitation, il est encore plus rapide de consulter un tableau de correspondance entre les logiciels propriétaires que vous pouvez connaître, et leurs équivalents libres. C'est notamment ce que proposent les sites :

---> <http://www.alter-web.org/>

---> <http://kaella.linux-azur.org/documentation/table.php>

Enfin, quand la période de transition est passée, il est possible de faire de nouvelles découvertes sur les sites suivants :

---> <http://www.framasoft.org/>  
(rubrique « Logiciels libres »)

---> <http://www.jesuislibre.org/>

---> <http://www.unixtech.be/>

---> <http://www.labo-linux.org/>

Simon Descarpentries



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com><http://www.unixgarden.com><http://www.editions-diamond.fr>

## Où trouver de l'aide ?

Le monde des Logiciels libres est une communauté, qui, avant de connaître de francs succès comme ceux de FireFox ou d'OpenOffice, était plutôt confidentielle. Il reste de cette période de bonnes habitudes concernant la documentation des logiciels (habitude indispensable pour un travail en groupe) et tout une culture d'entraide entre les développeurs et utilisateurs de Logiciels libres. Ainsi, il est quasiment toujours possible de trouver réponse à ses questions, en consultant de la documentation ou des articles d'aide sur un Logiciel libre.

### Documentation

La documentation sur un Logiciel libre s'installe généralement sur la machine, en même temps que le logiciel lui-même. On trouve le plus souvent avec les sources d'un programme des fichiers textes nommés **README** et/ou **FAQ**, expliquant brièvement comment installer et lancer le logiciel, ou apportant les réponses aux questions les plus fréquemment posées. On trouve ensuite dans la plupart des cas de l'aide sur l'utilisation d'un logiciel via la commande `man ($ man nom_de_logiciel)` (essayez donc `$ man man` pour commencer...). On trouve enfin des documentations plus détaillées, généralement en format HTML, dans le répertoire `/usr/share/doc/nom_de_logiciel` ou directement sur le site web des développeurs du logiciel.

### Aide

Trouver de l'aide peut sembler plus simple dans un premier temps, mais c'est aussi un piège qui empêche d'avancer par soi-même. Néanmoins, de nombreux sites d'entraide existent, regroupent beaucoup de personnes et recèlent de précieux contenus. Citons donc quelques sites généralistes :

- > <http://lea-linux.org/>
- > <http://frlinux.net/>
- > <http://forum.framasoft.org/>
- > <http://www.c-sait.net/>

Et d'autres accompagnant chaque distribution :

- > <http://forum.ubuntu-fr.org/index.php> (Ubuntu)
- > <http://forum.mandrivaclub.com/index.php?op=Fr> (Mandriva)
- > <http://fr.gentoo-wiki.com/Aide> (Gentoo)
- > <http://www.andesi.org/> (Debian)
- > <http://forums.knoppix-fr.org/> (Knoppix)
- > <http://www.alionet.org/> (Suse Linux)
- > etc.



debian

ubuntu

Novell.  
suse

Mandriva

Toutefois, poster efficacement un message sur ces forums nécessite d'observer quelques règles élémentaires de civisme... Par exemple, il est préférable de ne pas poster une question dont la réponse est mise en évidence par le premier lien de votre moteur de recherche préféré. Il est également bien vu de décrire son problème précisément, dans un bon niveau de langage, et en donnant le plus de précisions possible sur le contexte du problème : quelle distribution GNU/Linux est utilisée ?, avec quelles versions de logiciels votre problème se manifeste-t-il ?, sur quel ordinateur ?, etc.

Simon Descarpentries

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>  
<http://www.unixgarden.com>  
<http://www.editions-diamond.fr>

## Quelle est l'origine de Tux, la mascotte de Linux ?

C'est autour de l'année 1996, alors que le développement du noyau Linux fédérait de plus en plus de développeurs, que le projet décida de se doter d'un logo. Oscillant entre des images d'aigles ou de requins, le débat fut stoppé net lorsque Linus Torvalds, initiateur du projet, annonça qu'il était plutôt fan de manchots, qu'il s'était déjà fait charger et mordre par une de ces bêtes et qu'il les trouvait *fun*s.

Parmi les développeurs interloqués, certains se gaussèrent de l'allure chétive d'un manchot pygmée face au démon rouge armé d'un trident (le logo des systèmes BSD) ou face à un gros requin (comme il était proposé au cours du débat). Mais Linus ne se laissa pas démonter pour autant et il annonça que quiconque s'était déjà pris un coup de bec de manchot, fusse-t-il pygmée, ne raillerait pas un tel logo !

Le dessin final de la mascotte fut alors choisi parmi les participants à un concours de logo organisé la même année et c'est donc à Larry Ewing que l'on doit l'image originale du mythique Tux.

Mais pourquoi ce nom ? Bien sûr, de nombreuses suppositions existent sur l'origine de ce nom, certains avancent un jeu de mot en anglais avec le terme *Tuxedo*, mais l'explication la plus crédible est sûrement celle qui y associe l'acronyme suivant : *Torvalds Unix*...

Pour finir, beaucoup de gens pensent, à tort, que la mascotte de Linux est un pingouin, notamment parce qu'en anglais le terme *penguin* désigne aussi bien le pingouin (*auk* ou *razorbill*) (famille des alcidés) que le manchot. Mais il s'agit bel et bien d'un manchot, pygmée de surcroît (famille des sphéniscidés).



Tux la mascotte

### Signatures célèbres

Voici en art ASCII une représentation de Tux trouvée sur la signature de David Navarro, co-développeur du noyau Linux :

```

  .-.
 /  \  L I N U X
//   \ >Phear the Penguin<
/(   )\
 ^-_-^

```

Et voici celle de Paul Gray's :

```

-o)
/\  Message void if penguin violated
\_v Don't mess with the penguin

```



### Un dessin libre

Le dessin du manchot fut rendu libre par son auteur, qui demanda toutefois à être prévenu des éventuelles modifications apportées à son logo dans un premier temps. Les « sources » du manchot, au format The Gimp, sont donc disponibles sur la page perso de l'auteur consacrée à la mascotte : <http://www.isc.tamu.edu/~lewing/linux/>.

De fait, une quantité phénoménale de reprises et adaptations du logo original furent faites un peu partout de par le monde, ici pour les besoins d'un groupe d'utilisateurs de GNU/Linux, ailleurs pour une grande occasion, ou encore juste par humour... Des collections assez complètes des dérivés du Tux original sont entretenues aux adresses suivantes : <http://lwn.net/Gallery/> et <http://tux.crystalxp.net/>.

Simon Descarpentries

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Quel est l'historique de Linux ?

L'histoire du système que l'on appelle aujourd'hui « Linux » ou « GNU/Linux » remonte à bien plus longtemps que l'on pourrait l'imaginer et implique de nombreux acteurs. Voici ce qu'on peut en retenir.

En 1969, Kenneth Thompson, employé dans les laboratoires Bell, s'inspire du système Multics pour développer un nouveau prototype de système d'exploitation à temps partagé. Son nom, Unics, est un jeu de mot sur son prédécesseur. Avec le temps, il sera contracté en Unix.

Entre 1970 et 1975, Unix se définit et se raffine : il est écrit en C pour assurer sa portabilité. Il sait partager le temps processeur entre plusieurs tâches et plusieurs utilisateurs. Sa philosophie est de représenter les ressources du système sous forme de fichiers et de fournir une multitude de petits outils assurant chacun une fonction basique, pouvant être branchés les uns sur les autres afin d'accomplir des tâches complexes. Unix est alors diffusé aux organisations gouvernementales ainsi qu'aux universités sous forme de licences. En 1980, des chercheurs de l'université de Berkeley, lassés des nouvelles licences trop restrictives à leur goût, développent leur propre dérivé d'Unix : BSD.

En 1983, la situation devient encore plus critique pour Richard Stallman, « hacker » au MIT : alors que le logiciel était jusqu'alors diffusé sous forme de code source, de plus en plus de fournisseurs ne le distribuent plus que sous forme binaire fermée. Cette généralisation du logiciel propriétaire est perçue comme une menace pour la culture *hacker* basée sur le partage des connaissances et la coopération. C'est en 1984 que Richard Stallman quitte son travail pour se consacrer à la réalisation de l'audacieux projet GNU, visant à développer et maintenir un système Unix complet et totalement libre.

En 1991, même si Microsoft domine le marché des ordinateurs personnels avec son MS-DOS (déjà complètement dépassé technologiquement), Unix est encore largement utilisé dans l'industrie et dans la recherche. Le projet GNU se porte bien : il dispose de la majorité des outils nécessaires au système et du support de certaines organisations qui les utilisent sur des systèmes Unix propriétaires. Mais une pièce maîtresse manque pour compléter le système : le noyau, nommé « Hurd », est encore loin d'être fini.

C'est alors qu'un étudiant finlandais nommé Linus Torvalds poste le message suivant sur la liste de diffusion de Minix (un dérivé d'Unix développé à des fins éducatives par Andrew Tanenbaum) :

*« Je suis en train de faire un système d'exploitation gratuit (c'est juste un hobby – il ne sera pas aussi gros et professionnel que GNU) pour les clones 386 AT. Il infuse depuis avril et commence à être utilisable. »*

Plusieurs volontaires se passionnent pour ce nouveau système qui leur permet de comprendre et de bidouiller la machine et soumettent des patches à Linus. Initialement nommé « Freax » par ce dernier, le système prend rapidement le nom de « Linux », plus amusant aux yeux des autres contributeurs. En 1992, le noyau Linux est couplé au système GNU, produisant ainsi le premier système d'exploitation 100% libre : GNU/Linux.

Rapidement, les premières distributions, comprenant un système complet prêt à être installé par l'utilisateur, apparaissent : Patrick Volkerding crée Slackware en 1993 ; peu après, Debra et Ian Murdock créent Debian, Robert Young et Marc Ewing, Red Hat. En 1998, Gaël Duval crée Mandrake (rebaptisée Mandriva en 2005). Les premières versions de Gentoo apparaissent en 1999 ; enfin, en 2004, le milliardaire Mark Shuttleworth crée Ubuntu, une distribution grand public basée sur Debian. Avec l'évolution de ces distributions, les systèmes GNU/Linux deviennent de plus en plus simples à installer et à utiliser, popularisant le système auprès des hackers, des informaticiens, des passionnés d'informatique et finalement du grand public.

Aujourd'hui, on estime le nombre d'utilisateurs de GNU/Linux dans le monde à environ 30 millions (source : *Linux counter*). Le système est déployé depuis les gigantesques fermes de calcul de Google jusqu'aux machines de bureau. Après avoir rattrapé les environnements commerciaux en termes de performance, de convivialité et de simplicité d'utilisation, la vague du Logiciel libre part en avant : alors que le noyau Linux vient d'intégrer officiellement des éléments de support du temps réel, Openoffice.org commence à faire sérieusement de l'ombre à la suite Office de Microsoft et KDE 4 nous promet de révolutionner le bureau de l'utilisateur en 2007. Sur le plan pratique, la présence de plus en plus forte des DRM et autres restrictions numériques poussent de plus en plus d'utilisateurs vers les systèmes libres. L'inspiration donnée par les Logiciels libres et popularisée par GNU/Linux s'applique enfin à de plus en plus de domaines : Wikipedia en est une sur le savoir. La vague « libre » en est encore à ses débuts : l'avenir s'annonce passionnant...

Alexandre Courbot



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Boot, chargeur, noyau, services, X... Comment tout cela démarre-t-il ?

Lorsque l'ordinateur est mis sous tension, un programme initial situé en mémoire morte de la carte mère s'exécute : le **BIOS** (*Basic Input/Output System*). Il est responsable des tâches basiques d'initialisation du matériel et effectue quelques vérifications sur celui-ci. Il dispose également d'un utilitaire de configuration accessible par une combinaison de touches particulière (variable selon le modèle) permettant de configurer certains aspects du matériel (fréquence processeur, délai d'accès à la mémoire, etc.) ou du démarrage, notamment l'ordre dans lequel les lecteurs sont parcourus à la recherche d'un secteur de démarrage.

Pour chacun des lecteurs à tester, le BIOS vérifie si le premier secteur est *bootable*, c'est-à-dire héberge un programme chargé de démarrer le système d'exploitation (*bootloader*). Une fois qu'un lecteur satisfaisant cette condition est trouvé (typiquement, votre disque dur – mais cela peut aussi être une disquette, ou un *live-CD*), ce programme est exécuté. Si un système Linux est installé, il s'agira sans nul doute de **GRUB** ou de **LILLO**. Chacun de ces deux programmes est capable de charger et démarrer un noyau Linux, mais également de démarrer d'autres systèmes d'exploitation comme Windows. Pour les systèmes *multi-boot*, ils vous donnent généralement le choix du système à démarrer sous forme de menu. GRUB va plus loin que cette simple tâche puisqu'il s'agit également d'un véritable mini-système d'exploitation, permettant de monter des partitions et de les explorer.

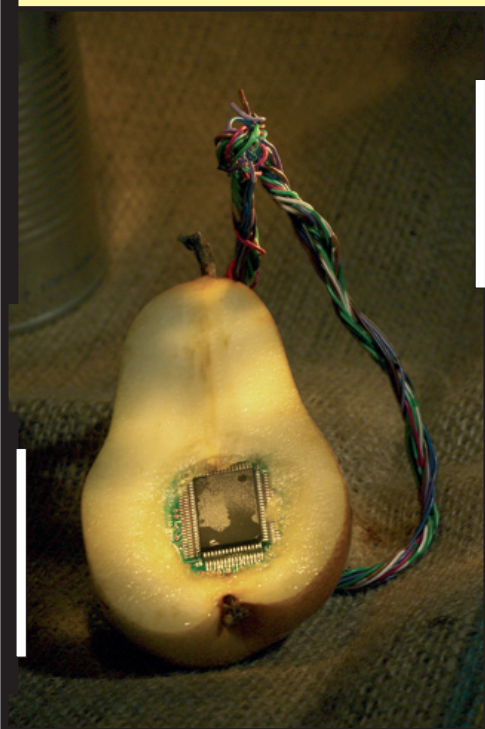
Une fois que la sélection du système à démarrer est effectuée (en l'occurrence, Linux), le bootloader charge le noyau et l'exécute. C'est à partir de ce moment-là que Linux « démarre » véritablement. Le noyau effectue quelques opérations d'initialisation sur le matériel et lui-même, puis monte la partition correspondant à la racine du système de fichiers (celle-ci aura été spécifiée par le bootloader). Enfin, il exécute le programme **init**.

Ce programme est chargé de terminer le démarrage du système en exécutant toutes les tâches situées dans le répertoire `/etc/rcX.d`, où X correspond au « niveau » d'exécution du système (généralement, 2 ou 5). Chacune de ces tâches est en fait un script chargé d'effectuer une opération spécifique (monter tous les systèmes de fichier) ou de démarrer un service (par exemple, KDM ou GDM pour le *login* graphique). Une fois que tous ces scripts sont exécutés, le système est entièrement démarré et peut être utilisé.

Actuellement, la tendance des distributions est d'abandonner **init**, considéré comme dépassé, au profit d'autres schémas de démarrage. En effet, l'exécution séquentielle d'un ensemble de scripts offre très peu de flexibilité dans le démarrage. Quid du démarrage d'un service à la condition qu'un matériel soit présent ? Et si ce dernier n'a pas été branché au même endroit que d'habitude ? Pourquoi démarrer des services jamais utilisés ? Plusieurs solutions ont été développées, notamment **initNG** qui permet de déterminer l'ordre optimal de démarrage de ces scripts et de lancer parallèlement deux scripts indépendants, afin de réduire le temps de démarrage. Récemment, une alternative plus moderne a vu le jour. **Upstart** est un remplaçant à **init** démarrant les services non plus d'après une liste, mais à la réception d'événements. Ceux-ci peuvent être le démarrage du système (auquel cas, on monte les systèmes de fichiers), mais également la présence d'une imprimante (démarrage du service d'impression), etc. Cette manière de faire permet de réduire grandement le temps de démarrage du système en l'affranchissant des services inutilisés, et en délayant l'activation de certains services à partir du moment où ils deviennent utilisés.

Upstart est utilisé dans la toute récente Ubuntu 6.10 (Edgy), et d'autres distributions devraient l'adopter sous peu...

Alexandre Courbot



## Serveur X, gestionnaire de fenêtres, bureau...

### Qui fait quoi ?

Serveur X, gestionnaire de fenêtre et bureau : voilà des notions qui déroutent bien souvent le débutant sous Linux et s'embrouillent dans son esprit. Comme bien souvent dans Linux, avant d'arriver sur votre bureau aux icônes savamment rangées, plusieurs couches se seront empilées pour vous servir. **Le serveur X**, ou **X.org**, est la toute première. Il ne sait presque rien faire, mais c'est le plus important et il le fait bien : il sait uniquement dessiner des primitives en fonction de ce que lui demandent les applications des couches supérieures (rectangles, lignes, ronds, lettres, couleurs...), surveiller la souris et le clavier. Il sait dessiner des rectangles spéciaux qui seront des fenêtres, et même cette fenêtre spéciale, sans bordure et aussi grande que votre écran, qui sera le fond d'écran. Cette fenêtre racine (ou *root window*) sera celle dans laquelle toutes les autres se placeront. La grande originalité du serveur X est qu'il peut fonctionner à distance, et envoyer une fenêtre vers un autre ordinateur en réseau. Pour l'instant, s'il fonctionne seul, les fenêtres qu'il trace sont impossibles à déplacer et à manipuler : il manque la deuxième couche.

**Le gestionnaire de fenêtre** (ou *window manager*) est la couche qui va s'occuper de faire figurer autour des fenêtres une bordure, les signes tels que Réduire, Agrandir, Enrouler ou Fermer. C'est grâce à lui que la souris va pouvoir accrocher les fenêtres et les déplacer, les recouvrir ou les empiler. Il propose généralement des thèmes de décorations et, par un clic-droit dans le vide, un menu de lancement d'applications. Des plus spectaculaires aux plus minimalistes, il existe des dizaines de gestionnaires, comme *Ion*, *Sawfish*, *twm*, *wmX*... *Kwm* et *Metacity* gèrent respectivement les fenêtres de KDE et Gnome. En lui-même, un gestionnaire de fenêtres pur ne propose donc ni icônes, ni barre des tâches. Mais quelques-uns, tels *IceWM*, *Fluxbox*, *WindowMaker* ou *fvwm*, incluent des *docks* ou des barres des tâches minimalistes.

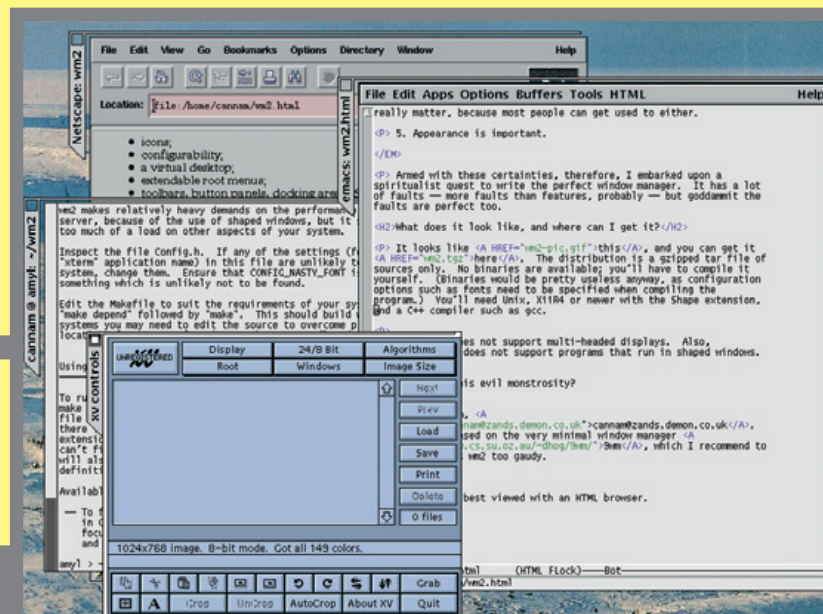
Enfin, le bureau arrive. **Un bureau** est un vaste ensemble cohérent : il inclut non seulement un gestionnaire de fenêtres, mais aussi des icônes, une barre des tâches, un menu de lancement, et surtout toute la machinerie interne pour que les applications puissent se parler, se surveiller, échanger des données, manipuler les fichiers et les périphériques, notamment le son, un véritable cassette à lui tout seul.

Pour conclure, gardons à l'esprit que, dans un souci de vulgarisation, cette description est volontairement très simplifiée : d'autres couches existent, notamment si on voulait intégrer dans cette explication l'interaction avec les pilotes.

Jean-Noël Pierron

*wm2*, un gestionnaire de fenêtre minimaliste :

des fenêtres, et c'est tout. Pour ses concepteurs, tout le reste n'est qu'encombrants gadgets...



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>  
<http://www.unixgarden.com>  
<http://www.editions-diamond.fr>

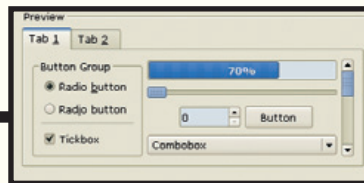
## GTK+, QT, programmes Gnome ou KDE ? Peut-on les mélanger ?

On parle souvent de programmes conçus pour Gnome ou pour KDE. Pourtant, la plupart du temps, il vous sera possible de lancer indifféremment un programme écrit pour l'un, tout en étant dans l'autre, et vice-versa. Ce qui différencie ces logiciels, c'est la bibliothèque de *widgets* utilisée. Que sont les widgets ? C'est l'ensemble des éléments qui constituent une interface graphique : bouton radio, case à cocher, onglet, jauge, bouton de texte comme **OK** ou **Annuler**, ascenseur, menu... Il en existe beaucoup sous Linux, mais les deux principaux sont **GTK+**, qui sert à développer Gnome, et **QT**, utilisé par KDE. Pour chacun existe un environnement de développement avec lequel on construit les applications.

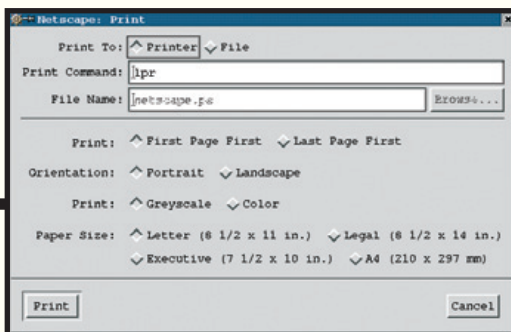
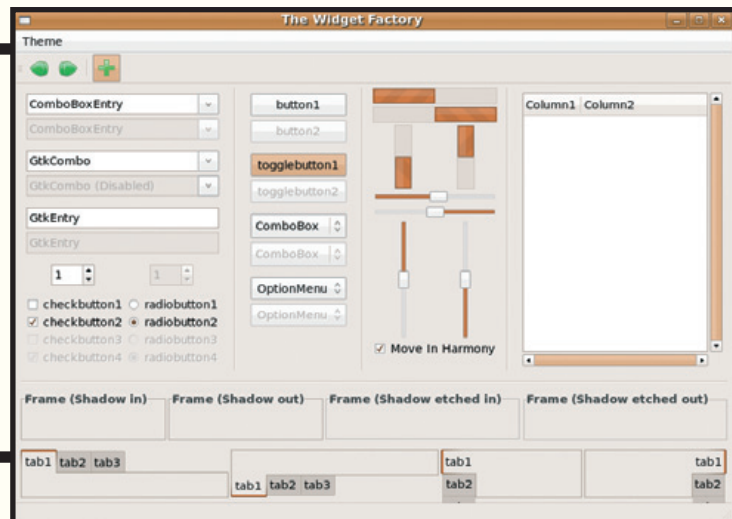
Au sein d'un même bureau, une bibliothèque de widgets en action n'est pas exclusive des autres : il est donc tout à fait possible d'utiliser, tout en étant dans Gnome, K3B, le logiciel de gravure de KDE, ou de lancer la messagerie Gaim dans KDE. Dans la mesure du possible, cette action n'est cependant pas à conseiller. En effet, cela revient à charger en mémoire vive non pas une mais deux bibliothèques de widgets, avec leurs règles et logique. De plus, certaines applications ont besoin de démarrer, en tâche de fond, presque toute l'intégralité du bureau auquel elles appartiennent. Lancer Konqueror ou K3B dans Gnome, par exemple, revient presque à faire tourner les deux bureaux en même temps. Si votre ordinateur est un peu limité en mémoire vive, il risque vite de ramer sévèrement. À cet égard, les applications GTK+ sont souvent plus indépendantes de Gnome que les programmes QT ne le sont de KDE, mais cette observation est loin d'être systématique.

Comme exemples d'autres widgets, citons wxWidgets, utilisé par VLC Media Player et Audacity, et qui ressemble fort à Windows 2000 ; tcl/tk, utilisé par aMSN ; les antiques Athena ou Motif, utilisés par les anciens Netscape 4, WordPerfect 8 ou Acrobat Reader 5. Certains programmes s'installent même avec leurs propres widgets, à l'instar d'OpenOffice. Tous fonctionnent indifféremment sous tous les bureaux, et on peut les charger ensemble en mémoire, même si la cohérence visuelle du bureau s'en ressent alors gravement. Au milieu d'un Gnome à l'apparence parfaite, un aMSN ou un Acrobat Reader 5, avec leurs boutons à l'ancienne et leurs polices de caractère crénelées font un peu tache...

Jean-Noël Pierron



Des exemples de widgets de QT et GTK+ : il est en outre possible de leur appliquer une multitude de thèmes.



Motif est une bibliothèque de widgets qui accuse son âge. Guère élégante, pas toujours très claire, c'était celle de Netscape 4 ou Acrobat Reader 5.

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com><http://www.unixgarden.com><http://www.editions-diamond.fr>

## Faut-il une configuration puissante pour utiliser GNU/Linux ?

GNU/Linux est un système d'exploitation idéal pour les configurations matérielles les plus modestes. L'installer sur une machine vieillissante est souvent une bonne solution pour lui redonner un petit coup de jeune, car elle peut servir, par exemple, de serveur et/ou de routeur, de poste bureautique ou de navigation internet, sans pour autant bourse délier lors d'une remise à niveau du matériel.

L'un des points forts de Linux est sans doute sa gestion de la mémoire, qui lui permet de lancer un plus grand nombre d'applications sans pour autant monopoliser la totalité de la mémoire disponible. Qui plus est, la gestion de la mémoire virtuelle de Linux est également meilleure, ce qui fait que le disque dur de la machine aura tendance à moins *swapper*. La possibilité de compiler soi-même son noyau ou ses applications (ou d'installer des paquets binaires précompilés pour votre processeur) permet également de tirer le meilleur parti du matériel plutôt que de n'en exploiter qu'une fraction, ce qui est un plus certain lorsque le matériel n'est pas des plus performants.

Un autre point fort de Linux est sa légèreté. Les applications sont très modulaires, basées sur des bibliothèques communes à plusieurs logiciels, et il en résulte certains gains de place et/ou usage de composants logiciels stables et éprouvés. Mais de façon générale, il est possible de choisir un à un les composants à installer et ainsi bénéficier d'une distribution complète, adaptée à vos besoins et ne consommant qu'un espace disque relativement faible. A ce sujet, il est bon de noter qu'il existe des mini-distributions Linux intégrant un environnement de bureau léger (comme **Fluxbox**, **IceWM** et d'autres) et les applications essentielles comme un client de courriels, un navigateur internet graphique, une suite bureautique, des applications graphiques et des outils de développement ou d'administration. Dans ce cadre, des installations complètes peuvent ne prendre que quelques dizaines à quelques centaines de Mo d'espace disque, ce qui met vraiment Linux à portée des machines vieillissantes.

Quelques exemples de mini-distributions :

----> Damn Small Linux : [www.damnsmalllinux.org/](http://www.damnsmalllinux.org/)

----> DeLi Linux : [delili.lens.hl-users.com/](http://delili.lens.hl-users.com/)

Il existe également des distributions prévues pour fonctionner à partir de CD-rom ou DVD-rom, de clé USB ou de disquettes. Certaines d'entre elles sont des versions *live* de Linux. Pour les curieux, voici un recensement de quelques mini-distributions : [www.linuxlinks.com/Distributions/Mini\\_Distributions/](http://www.linuxlinks.com/Distributions/Mini_Distributions/).

Toujours au chapitre des avantages de Linux par rapport à un système concurrent, comme Windows XP, figure sa relative immunité aux agressions externes, dues à une connectivité quasi permanente à l'Internet, depuis l'avènement du Haut-débit en France. Ainsi, les virus et les *spywares* qui harcèlent littéralement les plateformes Windows sont pratiquement inconnus sur plate-forme Linux. Il est important de noter que les logiciels qui traquent en permanence les virus, les programmes espions ou les codes malicieux installés sur votre machine consomment des ressources non négligeables en termes de charge du processeur ou de la mémoire RAM, sans compter la place prise sur le disque dur par leurs bases de données sans cesse extensibles (de nouvelles menaces apparaissent quotidiennement). L'avantage de Linux est d'être relativement insensible à ces agressions, et donc d'économiser une part non négligeable de ses ressources en se passant des anti-virus et autres logiciels anti-intrusion. Avec la prochaine arrivée de Vista, le successeur de Windows XP, pas moins d'1 Go de mémoire RAM et une configuration musclée seront nécessaires pour une activité bureautique et internet modérée mais confortable, là où le quart ou la moitié satisferont la plupart des usagers de Linux.

Enfin, si vous vous demandiez pourquoi s'embêter à installer Linux sur une vieille machine alors que celle-ci est si simple à changer de nos jours, il vous suffirait de penser aux étudiants, aux foyers défavorisés ou encore aux pays du Tiers Monde qui n'ont pas le même accès que vous à ces technologies récentes. Linux permet dans leur cas de démocratiser l'informatique, d'améliorer l'accès aux connaissances et à l'information et, *in fine*, de les aider à progresser. Saviez-vous que des associations se sont créées pour assurer la collecte d'anciens ordinateurs, installer des systèmes Linux, et les offrir à des écoles défavorisées, en France comme dans le monde entier ?

Olivier Saraja





Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Qu'est-ce que je ne pourrai pas faire facilement sous Linux ?

Si Linux sait faire plein de choses et suffit amplement pour une utilisation quotidienne, le débutant qui s'apprête à faire le grand saut doit quand même se préparer à certaines limitations. Foin du zèle, n'ayons pas peur de dire ce qui peut déranger les dévots du Libre ! Et plus tôt le nouveau venu est averti, moins l'effet de déception sera fort. Affirmons donc clairement ici que de grandes difficultés attendent celui qui veut tout faire marcher sous Linux, tout ce qu'il avait l'habitude d'utiliser facilement, sans se poser de questions, sous Windows. Ce sont en général les périphériques trop typiquement « windowsiens » qui poseront problème sous Linux, car trop spécifiques, à diffusion trop restreinte ou trop dépendants du cœur de Windows...

Les périphériques de jeux, ou autres jouets à brancher sur le port USB, comme les volants à retour de force ou les tapis de football, ne marcheront pas « *out of the box* » ; les périphériques de sécurité, style carte à puce sonore ou lecteur d'empreinte digitale sur clé USB, seront également silencieux ; les gadgets, comme les molettes de zoom sur les claviers USB, de même. Si vous êtes très dépendant, dans votre organisation, de fonctions avancées de votre *smartphone*, méfiance ici aussi : presque aucun ne sera reconnu sur simple branchement sous Linux. Et même s'ils le sont pour la partie stockage, ils ne pourront pas synchroniser les contacts et rendez-vous avec Kontakt ou Evolution. Même chose pour les assistants numériques : ceux sous PalmOS sont en général reconnus sans problème, mais ceux sous Windows Mobile donneront du fil à retordre même à un développeur.

Et que dire des webcams ? Jamais comme avec celles-ci le *plug-and-play* n'a été une aussi lointaine légende. Nous en avons récemment testé une petite dizaine, sous Mandriva 2006 : pas une, pas une seule, n'a été rapidement et aisément fonctionnelle, sur simple branchement... Même le support des clés USB est, encore aujourd'hui, bien trop souvent capricieux : nom de volume qui change d'une session à l'autre, clé reconnue comme ayant deux volumes avec des noms différents, *gnome-volume-manager* qui génère sans cesse des erreurs dans Mandriva 2007 puis plante mystérieusement, clé qui monte bien puis, la fois suivante, doit être montée à la main... Attention aussi aux matériels dont le constructeur garantit la bonne marche sous Linux, mais que certaines distributions ne prennent pas en compte et qui vous demanderont des heures de bidouillage pour fonctionner. Il y a peu, n'ayant rien trouvé sous Ubuntu 6.06 pour le faire fonctionner facilement, c'est le récepteur TV DVB-T externe USB2 d'AverMedia qui a eu raison de notre patience...

Attention aussi aux matériels dont le constructeur garantit la bonne marche sous Linux, mais que certaines distributions ne prennent pas en compte et qui vous demanderont des heures de bidouillage pour fonctionner. Il y a peu, n'ayant rien trouvé sous Ubuntu 6.06 pour le faire fonctionner facilement, c'est le récepteur TV DVB-T externe USB2 d'AverMedia qui a eu raison de notre patience...

Pour se rassurer, on peut se répéter que ce n'est pas de la faute de Linux lui-même, et que notre système pourrait tout à fait être performant avec tous ces matériels récalcitrants ou silencieux. Car, en vérité, pour des raisons de part de marché, trop de fabricants se désintéressent totalement des pilotes pour Linux, et semblent ne pas connaître de synchronisation avec autre chose qu'Outlook. Ils sont aussi souvent effrayés par les implications des licences de Logiciels libres et ont peur de dévoiler des secrets industriels en fournissant les quelques renseignements qui permettraient à la communauté d'écrire des pilotes performants.



Tout n'est cependant pas perdu, mais il faudra alors plonger sans rechigner les mains dans le cambouis, fouiner sur le web ou sur les forums, consacrer tout un dimanche à lire des documentations en anglais, à installer des bibliothèques et compiler des pilotes, pour enfin pouvoir exporter les contacts de ce foutu téléphone, ou voir s'afficher une image... Autre solution : se renseigner avant, en consultant abondamment la communauté, et n'acheter que des périphériques que vous serez sûr de pouvoir faire fonctionner. Tant il est vrai qu'un Linuxien averti en vaut deux.

Jean-Noël Pierron

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com><http://www.unixgarden.com><http://www.editions-diamond.fr>

## Pourquoi n'y a-t-il pas d'outils de défragmentation sous Linux ?



En fait, il est incorrect de dire qu'il n'en existe pas, mais il conviendra d'insister sur le fait qu'ils sont le plus souvent totalement inutiles. Pour comprendre cette réponse, il est nécessaire de comprendre ce qu'est la fragmentation.

La taille d'un fichier se mesure en kilo-octets. Un disque dur est compartimenté en blocs de quelques kilo-octets (par défaut, 4 ko pour le système de fichiers **ntfs**), ce bloc étant appelé un *cluster*. Il s'agit en fait de la plus petite taille que peut occuper un fichier donné sur votre disque dur. Par exemple, si vous avez un fichier de 17 ko, il va occuper 5 clusters (pour un total de 20 ko ; en effet, 4 clusters seraient insuffisants, car avec 16 ko au total, ils ne permettraient pas l'écriture intégrale du fichier), et il restera 3 ko réservés (20 ko disponibles dans les 5 clusters, mais seulement 17 ko occupés par le fichier), mais cependant non utilisés. Lorsque vous utilisez Windows et que vous avez besoin d'écrire un fichier sur le disque dur, le logiciel se positionnera sur le premier cluster libre

qu'il rencontrera, et commencera à le remplir. S'il a besoin de plus d'un cluster, il écrira également dans le cluster voisin, puis dans le suivant, et ainsi de suite, jusqu'à ce que le fichier entier soit écrit, ou que le cluster qui suit soit déjà occupé par un autre fichier. En ce cas, il continuera à parcourir le disque dur jusqu'à trouver un nouveau cluster libre et écrira dedans et ses voisins, ainsi de suite, jusqu'à ce que le fichier entier soit écrit, en autant de blocs séquentiels logiques possibles.

Bien sûr, il en résulte qu'un même fichier peut être écrit, petit bout par petit bout, en plusieurs endroits du disque. Si les données sont trop dispersées, la lecture d'un fichier peut en être plus ou moins ralentie, et le phénomène empire à mesure que le disque se remplit. On dit alors que le disque est « fragmenté », c'est-à-dire que les données ne sont pas rangées à la suite les unes des autres.

Sous Windows, il existe des utilitaires qui permettent d'analyser le contenu du disque, et de réorganiser son contenu de façon à ce que tous les clusters liés au même fichier soient contigus, et dans le bon ordre. Cette opération d'analyse et de réorganisation s'appelle « la défragmentation ». Elle permet, dans certains cas, de nettes accélérations du système.

Sous Linux, cette opération n'est pas nécessaire, car de par leur conception, les systèmes de fichiers **ext3** ou **reiserfs** (par exemple) vont avoir une tendance naturelle à n'écrire que là où il y aura la place de le faire, et d'optimiser l'espace disque consommé. Bien sûr, lorsque le disque commence à être plein, et que de gros volumes de données sont manipulés, il n'y aura pas de miracle : le système écrira ce qu'il pourra où il pourra, même s'il en résulte une fragmentation des fichiers.

Alors doit-on défragmenter sous Linux ? Si vous écoutez les prêcheurs anti-Windows, jamais de la vie : Linux est *tellement* plus intelligent que son système de fichier ne fragmente (*presque*, mais on l'oublie souvent) pas. Il est vrai que les systèmes de fichiers Linux font un usage optimisé de l'espace, mais il n'est malheureusement pas en mesure de savoir aujourd'hui quelle sera l'organisation optimale de votre disque après une ou deux années d'usage intensif. Pour résumer, si vos disques durs ont une durée de vie supérieure à deux ans, que vous travaillez régulièrement dessus, que vous manipulez quotidiennement de gros fichiers (vidéos, morceaux de musique...), que vous vous livrez au P2P ou encore qu'il vous arrive d'exhumer de vieux (et gros !) fichiers pour retravailler dessus, vous ressentirez vite, comme tout le monde, le besoin de défragmenter votre beau disque Linux !

Donc si vous avez besoin de réaliser cette opération, il ne vous reste plus qu'à vous intéresser à l'un des projets suivants : **Shake** (<http://vieu.net/shake/>) ou encore **Defrag** (<http://ck.kolivas.org/apps/defrag/>). Mais ne sombrez pas dans la paranoïa et assurez-vous bien d'en avoir réellement besoin avant de vous lancer dans l'aventure : pour un utilisateur lambda, cela n'en vaudra certainement pas la peine...

Olivier Saraja

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Est-il indispensable de savoir utiliser la console quand on utilise Linux ?

La réponse n'est pas « oui » ou « non ». Non que je ne veuille pas faire simple, mais tout dépend de ce que vous souhaitez faire sous Linux. S'il s'agit de faire de la bureautique sur une distribution Ubuntu, Mandriva ou similaire et que vous n'administrez pas votre station (votre mari/épouse, fils/fille, frère/sœur, copain/copine s'en charge), la réponse, c'est « non ». Vous n'en avez aucune utilité.

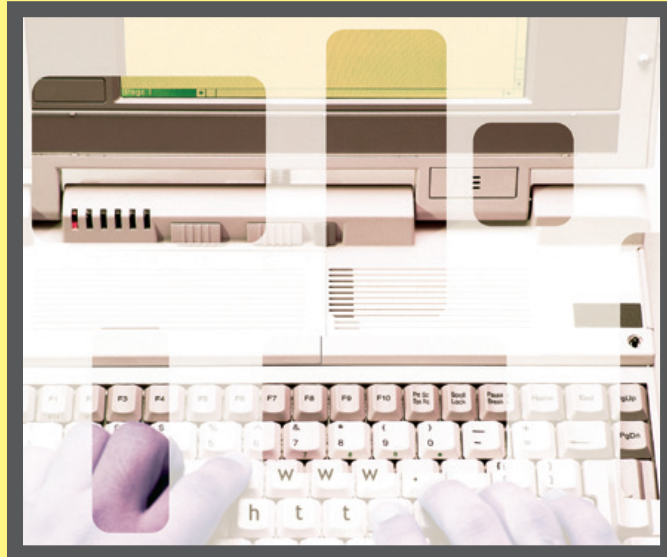
La console vise à permettre une interaction directe entre l'utilisateur et le système, au travers d'une interface la plus simple possible. Il existe d'ailleurs plusieurs langages de console : le Bash (qui est le plus usité, car par défaut sur nombre de distributions), mais il y a aussi le Csh, Tsh, Zsh, etc. Le monde GNU/Linux n'est jamais monolithique et la console ne fait pas exception. On entend par « simple » le fait qu'elle demande un minimum de ressources. Un système d'exploitation, c'est d'abord une console pour interagir avec l'utilisateur. C'est la base.

Néanmoins, si vous voulez administrer un petit serveur de partage de fichiers et disposer d'un outil de P2P derrière, la console n'est pas indispensable, mais pratique. Pas indispensable, car la plupart des logiciels sous Linux disposent d'une interface graphique de gestion de leurs fonctionnalités. Mais d'une part, ce n'est pas le cas de tous et, d'autre part, cette interface a parfois des limites... Souvent, on vous demandera de passer à travers la console (par exemple via Konsole sous KDE) pour les tâches d'administration (installation/retrait de paquets, configuration de logiciels, etc.). Parce que, dans ce cas précis, on est beaucoup plus rapide et plus efficace via la console que via les outils graphiques de la distribution. Les outils graphiques savent en général tout faire, mais ils demandent beaucoup de clics là où une ligne de commande suffit. C'est la même chose sur les systèmes propriétaires avec interface graphique. La ligne de commande existe et peut être très pratique (voire essentielle pour corriger une erreur), mais elle ne sert jamais ou presque dans la vie de tous les jours. Linux n'est pas différent pour les distributions bureautiques...

Cependant, si vous devez administrer un serveur qui est séparé de la station bureautique (ce que je conseille vivement pour des raisons de performance, de risque et d'entretien logiciel et matériel), la console est alors indispensable. Un serveur n'a pas besoin d'interface graphique, sauf cas particulier. C'est un outil dédié à une ou plusieurs tâches en réseau, mais qui ne nécessite pas d'être piloté via une artillerie lourde. D'ailleurs, certaines distributions-serveurs spécialisées n'embarquent pas par défaut d'interface graphique et ne proposent que des interfaces très simplifiées. Apache, ProFTPD ou encore MySQL ne s'administrent pas directement via une interface graphique de type GNOME ou KDE. Certaines existent, mais, le plus souvent, on utilise d'abord la ligne de commande, puis, ensuite, une interface web dédiée (PhpMyAdmin par exemple pour MySQL ou encore SWAT pour Samba).

Maintenant, comme vous pouvez le constater, tout dépend de votre usage et de vos besoins. Alors, la console, c'est bien, mais pas forcément nécessaire. La réponse à la question ci-dessus est donc « non » pour la majorité, « oui » pour les autres !

Sidoine Pierrel



## Quelle distribution choisir ?

La question du choix de la distribution revient régulièrement de la part des personnes qui débutent sur Linux. Elle donne lieu souvent à de beaux échanges, parfois vigoureux pour ne pas dire virils entre supporters de chaque distribution (voir la définition d'un troll page 33). Mais cela peut aussi venir de personnes chevronnées qui se demandent ce que va leur apporter (ou retirer) un changement d'environnement de travail. Car, avec près de 200 distributions différentes, autant dire qu'il y en a pour tous les goûts ! J'aurais presque envie de répondre que le choix d'une distribution étant très subjectif, vous n'avez plus qu'à vous rendre sur le site Distrowatch (<http://www.distrowatch.org>) et ensuite sur les sites officiels des distributions pour vous faire votre propre opinion. Mais je sens que c'est une réponse un peu courte qui risque de ne pas vous plaire, alors je vais vous aider quelque peu...

En fait, le choix de la distribution dépend de ce que vous voulez faire avec. Il vous revient de déterminer l'usage de la distribution pour ensuite pouvoir la choisir. Pour votre information, j'ai vécu ces huit dernières années avec successivement : une **Red Hat** (4.2, 5.0, 5.1), une **Mandriva** (5.3), une **Suse** (de la 5.3 à la 8.2), puis une **Gentoo** (environ 1 an et demi) et enfin une **Debian** et ses dérivées (**Knoppix**, **Kubuntu**, etc.). Actuellement, je vis heureux avec une Kubuntu et je ne m'en séparerai pas pour le moment. Mais pourquoi un tel cheminement ? Je cherche des distributions bureautiques (station de travail) qui soient faciles à administrer et surtout qui disposent d'un nombre maximal de paquets et de logiciels, avec une nette préférence pour les logiciels sous KDE (mais ce dernier point reste au goût de chacun). Avec de tels critères, vous obtenez le cheminement que je qualifierais de naturel de la part d'un utilisateur quotidien. En sachant tout de même qu'un changement de distribution n'a rien d'anodin ! Mieux vaut prendre le temps de bien choisir.

Ainsi, il y a des distributions en fonction des usages. On pourra cependant distinguer quatre grandes familles :

----> **Les distributions live** (voir page 37) : idéal pour un tout premier contact avec Linux. Ces distributions, qui tiennent sur un CD ou DVD, vous permettent de découvrir Linux et de nombreux Logiciels libres en toute simplicité. Et pour cause, vous n'avez absolument rien à installer sur votre disque dur. Le CD-live est complètement autonome. Retirez le CD et vous retrouvez votre ancien système d'exploitation !

----> **Les distributions généralistes** : ces distributions visent à vous permettre de tout faire, du serveur en RAID 5 matériel pour bases de données, à la station bureautique haut de gamme. Le nombre de paquets de ces distributions est souvent impressionnant et le nombre d'utilisateurs est imposant. Ce sont les mastodontes du monde Linux. Mais aussi, celles qui ont la plus féroce concurrence. A une époque en France, Mandriva régnait en maître, aujourd'hui c'est (K)Ubuntu, et, avant elles, c'était Red Hat... Autant dire que le vent tourne parfois brusquement et rien ne permet de prédire qui sera la prochaine. Ce sont celles que l'on rencontre en premier. Choisir l'une d'entre elles est autant une question de goût que de facilité d'accès (forums, *mailing-lists*, canaux IRC et *newsgroups*).

----> **Les distributions spécialisées** : à l'inverse des distributions citées ci-dessus, ces distributions ne visent qu'un usage. Elles sont donc utilisées par un certain type d'utilisateur de Linux, pour un certain domaine d'application (par exemple l'embarqué, ou encore les serveurs sur matériel spécifique), ou pour un usage spécifique (pare-feu, messagerie sécurisée, etc.). Elles ont des communautés d'utilisateurs réduites, mais dévouées. Elles sont souvent la réponse à des besoins de développement ou de production particuliers. On ne les utilise que lorsque le besoin auquel elles répondent se pose. Ce ne sont clairement pas des distributions à utiliser de prime abord, à moins de savoir exactement ce que l'on veut. Elles ont aussi souvent beaucoup moins de paquets que les généralistes, et pour cause, ce n'est pas leur vocation. Mais ce qu'elles font, elles le font bien !

----> **Les distributions sources** : je les mets de côté, car ces distributions sont un outil particulier dont la vocation pédagogique autant que technique est évidente. Ces distributions remplissent donc un double objectif : permettre d'apprendre comment fonctionne Linux et ses dérivés, mais aussi offrir à des professionnels un outil sur mesure. Par contre, elles demandent du temps et des moyens pour comprendre comment elles fonctionnent et les mettre en œuvre d'une manière efficace. Ces distributions sont donc généralistes, mais elles ont un mode opératoire si différent qu'une rubrique ne serait pas de trop pour en parler...

Voici rapidement un classement des distributions en fonction de l'usage :

----> Découverte, bureautique et petits serveurs : **K/X/Ubuntu**, **Mandriva**, **OpenSuSE**, **Fedora**. Ces distributions disposent de nombreux outils graphiques de gestion et d'une interface utilisateur agréable et installée par défaut. Idéal pour commencer ;

----> Serveurs généralistes et matériels spécifiques : **Debian**, **OpenSuSE** ;

----> Distributions sources : **Gentoo**, **Source Mage**, **LFS** ;

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>  
<http://www.unixgarden.com>  
<http://www.editions-diamond.fr>



---> Les distributions live (CD/DVD) : **Knoppix**, **Kanotix** (idéal pour tester, réparer, etc.) (voir page 37).

*Je ne cite pas, volontairement, les distributions payantes. De même, la liste est forcément incomplète. Je vous prie de ne pas y voir de ma part de volonté malveillante, mais il est impossible de classer ici les quelques 200 distributions actuellement disponibles...*

Pour finir, je vous invite à regarder, tester et re-tester si nécessaire, mais surtout ne pas désespérer... Parfois, tout ne donne pas le résultat que l'on veut du premier coup (autant tout le monde a appris à être tolérant envers les outils propriétaires, autant je suis parfois surpris des attentes que l'on peut avoir concernant les outils libres !). Un mot pour finir sur les distributions sources : ce sont de formidables outils pédagogiques. À nos lecteurs et lectrices disposant d'une réelle volonté d'apprentissage (et du temps qui va avec), je ne peux que vous encourager à faire un détour (long) par ces distributions. Une fois que vous l'aurez fait, vous aurez une maîtrise autrement plus puissante et plus efficace de Linux et des outils GNU qui l'accompagnent ! Et vous apprécierez alors à sa juste valeur les facilités qu'offrent maintes distributions généralistes...

Sidoine Pierret

## Pourquoi n'y a-t-il pas de nom de volume sous GNU/Linux ?

Dans un ordinateur, les fichiers proviennent de plusieurs supports physiques : disques durs, DVD-ROM, clef USB, etc. Le système Windows a choisi de les présenter sous forme de « volume ». Ainsi, pour accéder à un fichier, on donne d'abord le nom du média physique ou de la partition où il se trouve (**a**: pour l'antique lecteur de disquette, **c**: pour le disque dur principal, **d**: pour 2ème disque, **z**: pour un disque réseau, par exemple). Chaque volume dispose de sa propre arborescence.

Sous Unix, le choix est différent. Il n'y a pas de concept de « volume ». Il y a simplement une arborescence, dans laquelle est « monté » chaque périphérique ou partition d'un périphérique. Cela signifie que le système place le contenu du CD-rom par exemple sous `/mnt/cdrom`, que le disque principal est monté sous `/` (la racine). Un gros disque de données peut être utilisé pour les données personnelles. Il sera donc monté dans `/home`.

Chaque disque, partition et périphérique, est représenté par un fichier spécial se trouvant dans le répertoire `/dev` : `/dev/hda` représente le premier disque IDE ; `/dev/sda` le premier disque Serial ATA ou SCSI ou une carte mémoire. L'opération de « montage » correspond en fait à l'association d'un fichier spécial avec un emplacement de l'arborescence. Ainsi, le système est très souple. On choisit l'endroit où l'on rajoute des disques et, surtout, leur « place » ne change pas si l'on rajoute un disque dur.

On voit que l'on peut monter un périphérique sous une arborescence gérée par un autre périphérique. Par exemple, j'ai eu un problème de place dans mon répertoire personnel : le disque placé sous `/home` commençait à être rempli... J'ai ajouté un disque dur supplémentaire, que j'ai monté sous `/home/nb/hd200`. En gros, cela signifie que j'ai créé un répertoire `hd200/` dans mon compte `nb` qui va contenir les données du nouveau disque.

On voit d'ailleurs une limite au système : il faut séparer les données artificiellement vers un autre répertoire. Il existe une possibilité d'agrandir un espace lorsque l'on rajoute un disque : on utilise pour cela la fonctionnalité LVM (*Logical Volume Management*). Mais cela nécessiterait l'objet d'un article complet pour en parler...

La commande `df` en ligne de commande permet de connaître l'état des montages et le taux d'utilisation des différentes partitions. On peut ainsi visualiser la liaison entre les fichiers spéciaux de `/dev` qui correspondent au périphérique et l'endroit où ils sont montés. Par exemple :

Sys. de fich.	Tail.	Occ.	Disp.	%Occ.	Monté sur
<code>/dev/sdb1</code>	413G	159G	233G	41%	<code>/</code>
<code>/dev/sda5</code>	780M	25M	716M	4%	<code>/boot</code>
<code>none</code>	1,8G	44K	1,8G	1%	<code>/tmp</code>
<code>//192.168.12.24/~pub</code>	166G	108G	59G	65%	<code>/mnt/serv/pub</code>
<code>/dev/hda</code>	498M	498M	0	100%	<code>/mnt/cdrom</code>

On voit d'abord `/dev/sdb1` qui représente une partition montée à la racine du système de fichier (`/`). Ensuite, on voit `/dev/sda5` qui représente la 5ème partition du premier (**a**) disque dur (**sd** pour le scsi ou un disque SATA, **hd** pour un disque IDE) monté dans `/boot`. Le répertoire `/tmp` est marqué `none`, il utilise `tmpfs`, un système de fichier spécial qui utilise uniquement la mémoire vive. Cela permet de gagner énormément en vitesse, mais si on coupe le courant, on perd tout (cela dit, ce n'est pas très grave pour un répertoire temporaire...).

Ensuite, on peut voir un montage Samba (le système de partage d'origine Windows qui marche très bien sous Linux). On voit que l'on monte le répertoire `~pub` du serveur dont l'adresse IP est mentionnée. Ici, il n'y a pas de fichier spécial : en effet, sous Unix, la gestion du réseau est une des seules choses qui ne passe pas par un fichier. Enfin, on voit le montage d'un CD-rom IDE (d'où le `hda`) dans `/mnt/cdrom`.

Nicolas Boulay



## À quoi correspondent les numéros des distributions ?

« Mandriva Linux 2007 », « Ubuntu 6.10 », « Kaella 3.0 RC1 », « Frugalware Linux 0.6 Pre1 », « Damn Small Linux 3.1 RC3 », « Debian GNU/Linux 3.1r4 », « PCLinuxOS 0.93a », etc. Il y a de quoi se perdre dans cette jungle de sigles... Première chose à retenir, sachez que chaque éditeur de distribution a instauré son propre système de numérotation. Ainsi, une Ubuntu 5.10 n'est pas plus récente qu'une Kaella 3.0...

Ces numéros sont fixés par les éditeurs ; plus le chiffre est élevé, plus la distribution fournie par cet éditeur est récente (et ne voyez aucun rapport avec les numéros de version du noyau Linux).

En général, les distributions sont suivies de 2 chiffres : 1.0, 1.1, 1.2, 2.0, etc. La règle respectée par tous les éditeurs est qu'en cas d'évolution mineure d'une distribution, c'est le deuxième chiffre qui est augmenté ; en cas d'une évolution impliquant des améliorations beaucoup plus importantes, c'est le premier chiffre qui est augmenté. On retrouve d'ailleurs ce principe dans la numérotation des versions de logiciels. Ex. : Debian 2.0, Debian 2.1, Debian 2.2, Debian 3.0, etc.

Ensuite, certains sigles sont utilisés par tous :

- Les lettres a, b, etc. désignent respectivement les versions alpha, bêta, etc. d'une distribution. Cela signifie en général que vous avez affaire à une version en tout début de développement, encore en phase de test intensive.
- La lettre r ou R, signifie simplement *Release* (pour version). On peut trouver ainsi r1, r2, etc. Elle peut désigner une version stable.
- L'acronyme RC signifie *Release Candidate*, autrement dit, la version dotée de ce sigle est en bonne voie pour devenir la version stable finale. Plusieurs RC peuvent d'ailleurs se succéder, en général dans un laps de temps assez court (RC1, RC2, etc.).
- De la même façon, le sigle pre ou PR, pour *Preview Release* désigne une version en phase de devenir une version stable.

Retenez que dès que vous voyez apparaître une ou plusieurs lettres parmi le numéro de la distribution, il s'agit, la plupart du temps, d'une version de test (ce qui ne veut pas dire qu'elle n'est pas fonctionnelle ! Mais vous n'êtes pas à l'abri de rencontrer certains petits bugs...).

À côté de cela, certaines distributions suivent une numérotation un peu différente, qui leur est propre. Ainsi, la distribution Mandriva, qui sort une nouvelle version tous les ans, reçoit tout simplement le nom de Mandriva 2006, Mandriva 2007, etc.

La distribution Ubuntu (et toutes ses dérivées : Kubuntu, Xubuntu, Edubuntu) a adopté, quant à elle, un système de numérotation basé sur la date de sortie des versions : deux versions sortent jusqu'à présent tous les ans, une en avril (sauf en 2006 où l'équipe de développement a sorti exceptionnellement la version en juin), l'autre en octobre (ceci pour être synchronisé avec les versions de l'environnement de bureau Gnome qu'elle intègre, dont une nouvelle version sort tous les 6 mois). Nous obtenons ainsi pour 2005 : Ubuntu 5.04 et Ubuntu 5.10 ; pour 2006 : Ubuntu 6.06 et Ubuntu 6.10.

Enfin, certains éditeurs, en plus du numéro de version, dotent leurs versions stables de petits noms sympathiques. Ainsi, chez Canonical, les différentes versions d'Ubuntu reçoivent des noms d'animaux : Ubuntu 5.10 = *Breezy Badger* (le hérisson vénérable), Ubuntu 6.06 = *Dapper Drake* (le canard pimpant), Ubuntu 6.10 = *Edgy Eft* (le triton nerveux), etc.

De même, chez Debian, on a le plaisir de retrouver des noms de personnages issus du film d'animation *Toy Story* des studios Pixar : Debian 1.1 = *Buzz*, Debian 2.0 = *Hamm* (le cochon-tirelire), Debian 2.2 = *Potato* (Monsieur Patate), Debian 3.0 = *Woody*, Debian 3.1 = *Sarge* (le chef des soldats), etc.

Et puisque l'on parle de Debian, sachez enfin que cette distribution est par ailleurs toujours disponible en trois « branches » : *stable* (correspond à la dernière distribution Debian officiellement sortie ; actuellement, il s'agit de la version 3.1r4, appelée « *Sarge* »), *testing* (contient des paquets qui n'ont pas encore été acceptés dans la version stable. De ce fait, elle propose des versions de logiciel beaucoup plus récentes ; actuellement, il s'agit de *Etch*), *unstable* (version de développement, constamment alimentée de nouveaux paquets ou de mises à jour de paquets ; cette version est appelée « *Sid* »).

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Êtes-vous incollable sur le système GNU/Linux ?

vrai faux

- |                          |                          |    |  |
|--------------------------|--------------------------|----|--|
| <input type="checkbox"/> | <input type="checkbox"/> | 1  | Linux est le premier système Unix.   |
| <input type="checkbox"/> | <input type="checkbox"/> | 2  | Richard Stallman est l'inventeur du système GNU/Linux.   |
| <input type="checkbox"/> | <input type="checkbox"/> | 3  | Tux est le nom du noyau du système GNU/Linux.  |
| <input type="checkbox"/> | <input type="checkbox"/> | 4  | Il faut être doué en informatique pour utiliser Linux.   |
| <input type="checkbox"/> | <input type="checkbox"/> | 5  | Un <i>live-CD</i> désigne un CD qui ne peut s'utiliser qu'une seule fois.  |
| <input type="checkbox"/> | <input type="checkbox"/> | 6  | Si j'installe Linux, je vais perdre toutes mes données et mon système Windows.   |
| <input type="checkbox"/> | <input type="checkbox"/> | 7  | Si je veux installer GNU/Linux, j'ai besoin d'un ordinateur plus récent.   |
| <input type="checkbox"/> | <input type="checkbox"/> | 8  | Mon matériel (souris, clavier, webcam, imprimante, etc.) sera automatiquement reconnu sous mon système GNU/Linux.  |
| <input type="checkbox"/> | <input type="checkbox"/> | 9  | Il y a, à ce jour, peu d'utilisateurs de GNU/Linux.  |
| <input type="checkbox"/> | <input type="checkbox"/> | 10 | Il est indispensable d'avoir un accès à Internet pour utiliser GNU/Linux.  |
| <input type="checkbox"/> | <input type="checkbox"/> | 11 | Il n'y a pas de virus sous GNU/Linux.  |
| <input type="checkbox"/> | <input type="checkbox"/> | 12 | Linux est un super système, car il ne fragmente pas le disque.   |
| <input type="checkbox"/> | <input type="checkbox"/> | 13 | Un logiciel est dit « libre » lorsque l'on peut l'utiliser sur tous les systèmes d'exploitation.   |
| <input type="checkbox"/> | <input type="checkbox"/> | 14 | Les Logiciels libres sont gratuits.  |
| <input type="checkbox"/> | <input type="checkbox"/> | 15 | Les Logiciels libres ne plantent jamais.   |
| <input type="checkbox"/> | <input type="checkbox"/> | 16 | Il est très facile de se procurer et d'installer de nouveaux logiciels pour son système GNU/Linux.   |
| <input type="checkbox"/> | <input type="checkbox"/> | 17 | Pour obtenir les versions récentes des logiciels pour Linux, je dois faire un don à son équipe de développement.   |
| <input type="checkbox"/> | <input type="checkbox"/> | 18 | Si je trouve un bon logiciel sous licence GPL, je peux le copier et le distribuer autour de moi en toute légalité.   |
| <input type="checkbox"/> | <input type="checkbox"/> | 19 | Un « fork » désigne un logiciel dont le développement est basé sur un logiciel déjà existant.  |
| <input type="checkbox"/> | <input type="checkbox"/> | 20 | Chaque distribution GNU/Linux propose un système de fichiers qui lui est propre. De ce fait, il est difficile de s'y retrouver d'une distribution à l'autre.         |
| <input type="checkbox"/> | <input type="checkbox"/> | 21 | KDE et Gnome sont les environnements de bureau les plus répandus. Si j'en choisis un, je ne pourrai pas utiliser les applications destinées à l'autre et vice versa. |
| <input type="checkbox"/> | <input type="checkbox"/> | 22 | Je peux utiliser mes documents au format MS Word (.doc) sous Linux.  |
| <input type="checkbox"/> | <input type="checkbox"/> | 23 | Tous les logiciels propriétaires ont un équivalent libre.  |
| <input type="checkbox"/> | <input type="checkbox"/> | 24 | Un système GNU/Linux ne permet pas d'écouter de la musique ou de regarder une vidéo.   |
| <input type="checkbox"/> | <input type="checkbox"/> | 25 | Contrairement aux systèmes MS Windows, il n'existe pas de jeux pour GNU/Linux.   |
| <input type="checkbox"/> | <input type="checkbox"/> | 26 | Je peux faire tourner des applications Windows sur mon système GNU/Linux.  |
| <input type="checkbox"/> | <input type="checkbox"/> | 27 | Un « troll » désigne familièrement un fêru d'informatique.   |
| <input type="checkbox"/> | <input type="checkbox"/> | 28 | Lorsqu'un démon tourne sur ma machine, cela signifie que celle-ci a été corrompue.   |
| <input type="checkbox"/> | <input type="checkbox"/> | 29 | UTF-8 est une norme de codage de caractères, généralement utilisée par défaut dans les systèmes GNU/Linux.   |
| <input type="checkbox"/> | <input type="checkbox"/> | 30 | Pour ne pas rencontrer de problème en utilisant son système GNU/Linux, il vaut mieux se connecter en <i>root</i> en permanence.                                      |

( Réponses page 80 )

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>  
<http://www.unixgarden.com>  
<http://www.editions-diamond.fr>

## Qu'est-ce qu'un newbie/noob et pourquoi ceux qui savent ne semblent pas aimer les débutants ?



Dixit le Jargon Français (<http://www.linux-france.org/prj/jargon/>), le terme *newbie* est une contraction de *new in business*, un nouveau venu donc. La traduction française la plus proche et ayant la même connotation péjorative pourrait être « un bleu ». L'interprétation de ce qualificatif est très litigieuse : il peut à la fois signifier « débutant » et « idiot » (ou un mélange des deux).

La seconde partie de la question met en lumière l'écart qu'il peut y avoir entre un utilisateur averti et un débutant. Habituellement, le premier aimerait que le second se débrouille seul. Selon lui, le meilleur chemin vers l'acquisition de connaissances techniques se résume en une succession de recherches, d'erreurs, et de corrections d'erreurs.

Le second, le débutant, attend généralement de l'utilisateur expérimenté une bribe d'information, un coup de main, de l'assistance ou de l'assistantat. C'est cette dernière requête qui est généralement le point de friction. Il n'est donc pas rare que les utilisateurs expérimentés voient un newbie (sens péjoratif) en chaque débutant et que ces derniers voient en chaque utilisateur bien plus avancé un élitiste avare en conseils.

Pour éviter les conflits, mieux vaut donc rechercher par soi-même et faire preuve de patience, d'humilité, d'écoute et de politesse. N'appellez à l'aide qu'en dernier recours. Dites-vous simplement que ces utilisateurs avancés ont suivi le même chemin et qu'ils y sont bien arrivés. Et puis, après tout, s'il n'y avait pas de problèmes à résoudre, l'informatique ne servirait pas à grand chose...

Denis Bodor

## Ça veut dire quoi « RTFM » ?

Prenant votre courage à deux mains, vous avez osé poser une question sur le forum des utilisateurs de telle ou telle distribution, tel ou tel logiciel, etc. Et pour toute réponse, vous obtenez un « RTFM ». Grand classique du monde Linux, RTFM est en fait l'acronyme de l'expression *Read The Fucking Manual*, qui signifie littéralement, « Lis le p\*tain de manuel » (sous-entendu « de la distribution/du logiciel/des commandes Unix, etc. »). Poétique, n'est-ce pas ? Autrement dit, personne n'a envie de perdre du temps à répondre à votre question...

« Pourquoi tant d'agressivité, alors que je ne suis qu'un pauvre petit utilisateur débutant qui ne demande qu'à progresser ? » En général, cela signifie que votre question mérite un peu plus de recherche de votre part, car la réponse est on ne peut plus évidente à dénicher si vous prenez la peine de jeter un œil au manuel de l'utilisateur. Ne voyez donc pas dans « RTFM » une insulte directe à votre personne, mais plutôt un conseil constructif du genre « lisez attentivement la documentation avant de crier à l'aide ».

F. B.

## C'est quoi un geek ?

Le *geek* (prononcez [dʒi:k] ou [gi:k] au choix) est défini comme suit par le Jargon Français :

« Fou taré d'informatique, tout comme le *nerd*. Typiquement, il est binoclard, avec des boutons, et théoriquement sans petite copine ».

On ne retiendra que la première partie de cette définition (l'autre partie étant basée sur un ensemble de préjugés complètement infondés...). Sachez qu'à l'origine, le terme « geek » est l'équivalent de « passionné », « fou de quelque chose », et ce, quel que soit le domaine. Désormais, il est surtout employé dans le domaine de l'informatique et s'est paré d'une connotation plutôt positive (n'est geek que l'individu qui dispose de réelles compétences en informatique, d'un niveau bien supérieur à la majorité de la population).

Pour résumer, le terme « geek » désigne un informaticien « qui-s'y-connaît » et qui passe un temps considérable devant son ordinateur. Au féminin, on parlera de « geekette » (néanmoins, elles sont beaucoup plus rares !). Le geek aime programmer, développer, déboguer, accumuler du matériel informatique plus ou moins obsolète (« ça pourrait servir... »). Il est sans cesse à la recherche de nouveautés et d'actualités en matière d'informatique et de technologie. C'est pourquoi il est de nature très curieuse (dans le bon sens du terme).

Les geeks se retrouvent la plupart du temps (virtuellement) sur divers canaux IRC de leur domaine favori (histoire de partager leurs expériences et leurs connaissances d'informaticien hors pair) ou encore lors de manifestations organisées autour de sa passion : les Rencontres Mondiales du Logiciel Libre, les *install party* organisées par le LUG de leur région, etc.

Pour parfaire la description du geek, je ne saurais que trop vous recommander le site Web des Copines de Geek (<http://www.copinedegeek.com/>), criant de vérité et très comique de surcroît...

R\*X (10^F  
 O THEN D  
 R# (F#) : F

R\*X (10^F

F. B.



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Qu'est-ce qu'un troll ?

Le troll est un personnage du bestiaire heroïco-fantastique qu'il ne fait pas bon rencontrer au détour de son voyage. Sa laideur tout comme sa taille et son odeur n'aident pas à rendre le personnage sympathique. Et son mauvais caractère légendaire fait de vous, pauvre voyageur du monde numérique, un met de choix pour son prochain casse-croûte. Car le troll est un personnage mythique de l'informatique, en particulier de l'informatique libre. Il occupe une place de choix dans les débats techniques ou sociologiques qui peuvent avoir lieu dans les forums, listes de diffusion, canaux IRC ou autres endroits de discussion et donc potentiellement de bagarre.

On appelle « troll » le fait que plusieurs personnes s'invectivent par messages électroniques interposés. Et nourrir le troll signifie que l'on ajoute des arguments à la bagarre verbale qui a lieu au sein de la liste de diffusion, du forum ou du canal IRC auquel on est abonné ou connecté. On est expressément prié ne pas nourrir le troll ! Mais au contraire le laisser mourir de sa belle mort. On l'espère la plus rapide possible tant un troll est désagréable pour le lecteur ou le participant qui ne veut pas en faire partie et qui, lui, avait une discussion des plus passionnantes en cours !

Il faut d'ailleurs bien comprendre que le sujet qui démarre un troll n'a pas d'importance. Le troll une fois lancé n'a plus rien à voir avec ce qui en fut l'origine. Il existe pour lui-même et plus il grossit, plus il est désagréable. Et tous les sujets peuvent potentiellement donner naissance à un troll pour peu qu'un ou plusieurs participants soient mal lunés. Par exemple : « Quelle est la meilleure distribution ? Pourquoi le logiciel X ne fonctionne pas sous la distribution X alors qu'avant il était génial ? La plate-forme X, c'est la meilleure ! etc. »

Le seul moyen d'éviter un troll est de ne pas répondre aux insultes ou invectives lorsqu'elles ont lieu, rappeler le cas échéant les règles de sociabilité et de bienséance qui siéent à toute discussion entre personnes de bonne compagnie (et c'est le cas des informaticiens, mêmes « libristes ») et enfin, de la part des gestionnaires du lieu de discussion, de reprendre le contrôle des choses au plus vite. Quitte à bannir le ou les impétrants... Plus le troll est jeune, plus il peut être contrôlé, puis tué. Enfin, de la part des utilisateurs qui lancent une question à fort potentiel « trollesque », il est bon de rester dans le fil de la question tout autant que de calmer les ardeurs de celui ou de ceux qui n'ont pas compris que vous ne voulez pas polémique, mais obtenir une réponse technique/sociologique/humaine satisfaisante.

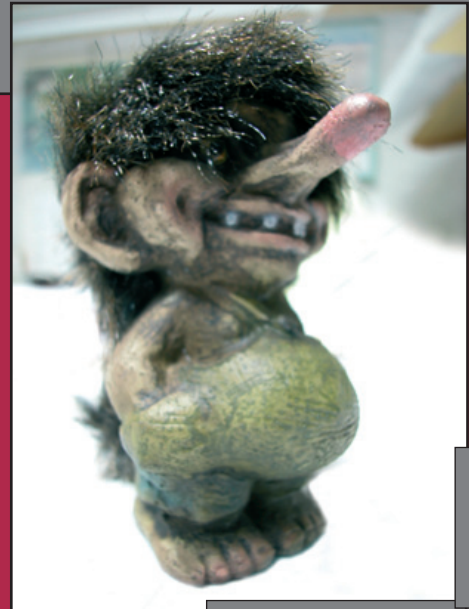
Il est bon de se souvenir qu'un forum, une liste de diffusion, un canal IRC ou un *newsgroup* sont des lieux où la parole est permanente. C'est-à-dire qu'à la différence d'une discussion orale à bâtons rompus, le texte va rester et le poids des mots en sortir profondément renforcé. Littéralement, insulter quelqu'un oralement peut toujours se reprendre, par contre, à l'écrit, la force des mots est considérablement renforcée.

Enfin, il ne faut pas confondre troll et plaisanterie. Il arrive que quelques dinosaures du monde Unix se lancent volontairement dans une série de messages à tendance « trollesque », mais dans lesquels chacun prend le recul nécessaire pour que cela reste poli, plaisant voire franchement amusant. Souvent, ces « trolls pour de rire » sont indiqués par les balises suivantes : <troll> texte </troll>. Ce troll peut être nourri pour autant qu'il reste dans les marges de la sociabilité et de la bienséance. Il doit seulement amuser tout le monde. C'est le sérieux qui rend le troll méchant.

En bref, par principe, on ne nourrit pas le troll, même si l'on se sent directement visé...

Sidoine Pierrel

• Un lien sur le sujet : <http://www.uzine.net/article1032.html> [fr]



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## C'est quoi l'OSI ?

L'*Open Source Initiative* (<http://opensource.org>) est le fruit d'une réflexion de Bruce Perens et d'Eric S. Raymond. Ces deux personnages sont des membres clés de la communauté Open Source et certainement en partie responsables de son évolution au cours des années 1996 à 2000, lorsque cette dernière a vu aller à elle de nombreuses sociétés de logiciels nord-américaines.

L'OSI est une organisation de promotion du Logiciel libre. Elle a été fondée en 1998 au moment où Netscape a rendu public le code source de son navigateur, devenu depuis le projet Mozilla. Cette organisation vise à promouvoir la notion « d'Open Source » (code source ouvert littéralement). Cette notion est moins connotée idéologiquement que celle de « Logiciel libre » telle que proposée par Richard Stallman (l'auteur de la licence GPL) et la *Free Software Foundation*. L'objectif de l'OSI est de fournir à l'industrie du logiciel et aux sociétés de services en ingénierie une définition des Logiciels libres et un vocabulaire moins restrictif que celui de Richard Stallman et ses équipes.

L'OSI est à l'origine aussi d'un processus de certification des licences afin de garantir que les logiciels proposés par des groupes de développeurs ou des sociétés sont bien en adéquation avec les valeurs du Logiciel libre. Cette certification a été souvent débattue et combattue par un certain nombre, car elle a participé à l'inflation parfois schizophrénique du nombre de licences. De plus, on a souvent accusé l'OSI d'être permissive dans sa définition de l'Open Source en estampillant des licences incompatibles avec la GPL !

On peut clairement parler de schisme, un terme religieux grec qui signifie « séparation » ! L'OSI a été vécue par certains comme une trahison. Car, si la définition et le cadre offerts par l'OSI ont permis l'essor actuel des Logiciels libres en les déconnectant de la notion de « libre » de la licence GPL proprement dite et de la FSF, la permissivité (cohabitation avec le logiciel propriétaire) de l'OSI n'est pas un objectif ou une valeur acceptable pour les plus fondamentalistes des communautés de développement ou de pensée du Logiciel libre. Et même si avant l'OSI, il existait déjà des licences libres alternatives à la GPL, il n'y avait pas de groupe de pression en faveur du Logiciel libre hors de la FSF, qui est aussi la structure officielle d'accueil pour Stallman et la GPL.

Car l'OSI a aussi participé grandement au développement des Logiciels libres via une série de publications pendant la période 1998 – 2000. C'est elle qui est à l'origine, via ses dirigeants, de la publication de mémos internes de Microsoft montrant sa farouche opposition aux Logiciels libres et sa volonté de les détruire au plus tôt (voir [http://en.wikipedia.org/wiki/Embrace,\\_extend\\_and\\_extinguish](http://en.wikipedia.org/wiki/Embrace,_extend_and_extinguish)).

L'objectif des fondateurs était d'offrir à l'industrie et aux particuliers un cadre philosophique et juridique dépassionné pour permettre la production de Logiciels libres. Huit ans après, cet objectif s'est avéré essentiel dans la croissance desdits Logiciels libres. Outre une définition définitive des Logiciels libres, il a permis d'aller au bout de la notion de « licence » par l'absurde (trop c'est trop !) et de faire revenir de nombreux acteurs vers la GPL tout en maintenant pour les autres un cadre qui leur permet de respecter le Logiciel libre tout en ayant leur différence et leur particularité. Il a offert aux plus grosses sociétés de l'informatique traditionnelle un cadre souple et efficace pour qu'elles changent sans heurt leur stratégie. On peut regretter la dilution de certains aspects politiques dans le marketing et le business. On peut regretter les positions de la FSF (qu'elle a toujours d'ailleurs). Mais ce qui est sûr, c'est que le Logiciel libre ne serait pas ce qu'il est sans l'OSI et ses compromis intelligents et intelligibles !

Sidoine Pierrel



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## C'est quoi la LSB ?

La LSB, acronyme de *Linux Standard Base*, est une norme. En soit, on peut la respecter ou pas ! Mais si vous voulez que votre logiciel ou votre distribution soit compatible avec les autres distributions Linux, il vaut mieux que vous la lisiez et que vous l'appliquiez !

La LSB est née en juin 2001 des différences grandissantes dans la gestion des binaires et de l'arborescence entre les principales distributions Linux. Cela prenait des proportions inquiétantes qui faisaient redouter une incompatibilité définitive entre les distributions (principalement SuSE, Red Hat, SCO, Debian), de la même façon que cela avait eu lieu entre les différents UNIX propriétaires vingt et quelques années plus tôt. Et donc un affaiblissement notable du mouvement du Logiciel libre. Car il aurait fallu non plus gérer les paquets en fonction des distributions, mais aussi les différences d'installation. Cela devenait cauchemardesque avec des dizaines de distributions à suivre ! Ainsi, les gestionnaires de certaines distributions commerciales (donc pas Debian !), se sont réunis et ont décidé de créer une norme qui garantisse la compatibilité entre les distributions. La LSB spécifie par exemple un ensemble

de bibliothèques standards, la structure de la hiérarchie du système de fichiers, etc. Cette norme peut même faire l'objet d'une validation par le groupe informatique *The Open Group*, en collaboration avec le *Free Standards Group*.

La LSB a été bien accueillie dans la forme, mais pas forcément sur le fond. D'une part, parce qu'elle s'est passée de l'accord d'un groupe majeur du Logiciel libre, Debian, mais aussi parce qu'elle a pris certains partis qui n'ont pas fait l'unanimité. Le choix par exemple du RPM comme format de paquet officiel a été vécu comme un non-sens et une manière d'exclure les distributions non commerciales à base Debian ou dérivées. Depuis, la situation a évolué et Debian inclut même un support optionnel de la LSB. Mais d'autres critiques ont eu lieu sur la méthode même choisie pour les tests de validation. En particulier, la pauvreté des tests écrits ont été décriés avec force.

Malgré tout, la LSB est un élément essentiel à ce jour, car cette initiative a permis de maintenir la cohérence et la cohésion des distributions et ainsi d'éviter une mort prématurée du Logiciel libre par dispersion des efforts dans un environnement technique qui serait devenu kafkaïen.

Sidoine Pierrel

## C'est quoi l'EFF ?

L'*Electronic Frontier Foundation* (<http://www.eff.org/>) est une association américaine (*non-profit organization*, l'équivalent de la loi de 1901 en France – hors territoire du concordat où s'applique la loi de 1908 -) dont l'objectif est de défendre le droit à la liberté d'expression (tel que défini dans le premier amendement de la constitution des États-Unis d'Amérique) dans le contexte du monde numérique.

Basée aux États-Unis, cette Organisation Non Gouvernementale s'est illustrée dans le combat contre les lois liberticides en termes de droits numériques (DRM, *Millenium Act* mais aussi brevets logiciels). Adeptes d'un *lobbying* forcené, elle est née des premières opérations de l'état fédéral américain pour contrôler le monde numérique au début des années 1990. Forte de certains succès en termes d'assistance juridique, mais aussi de *lobbying* ainsi que d'une abondante littérature et de très nombreuses propositions, cette ONG n'est pas non plus exempte de critiques autour de son action (changer la loi lorsque celle-ci est liberticide et non pas la limiter dans ses moyens en créant systématiquement une jurisprudence), de sa définition « très ouverte » des droits à la libre expression y compris lorsque celle-ci contrevient à d'autres droits tels que les droits d'auteurs et des débats plus techniques autour, par exemple, du spam et des moyens de lutter contre ce dernier.

L'EFF est aujourd'hui un contributeur essentiel au monde des Logiciels libres par sa réflexion sur l'usage des technologies numériques (travaux théoriques, mais aussi recommandations pratiques) et leur évolution, ainsi que par son soutien sans faille à un monde dans lequel l'utilisateur n'est pas uniquement consommateur, mais aussi producteur, s'il le désire.

On peut apprécier ou pas son action, être d'accord ou pas avec tout ou partie de ses objectifs, force est de constater au vu de son histoire que, sans elle, les Logiciels libres ne seraient certainement pas ce qu'ils sont. L'EFF n'est pas parfaite, n'est pas unique, mais elle est nécessaire dans le microcosme des Logiciels libres.

Sidoine Pierrel



## Qu'est-ce qu'une distribution Linux ?

Il est bon d'abord de définir ce qu'est une distribution. On appelle « distribution » l'ensemble cohérent de logiciels que forment :

- > un noyau ;
- > les outils système liés au noyau utilisé ;
- > et tous les autres logiciels nécessaires pour faire fonctionner un micro-ordinateur et remplir les missions de production qui lui sont assignés par ses utilisateurs (serveur ou station, environnement graphique ou logiciels de bases de données, etc.).

Une distribution se présente sous la forme d'un support d'installation autonome (CD, DVD, clé USB, disque dur à mémoire flash) et d'un ensemble de logiciels spécifiques pour gérer le quotidien du micro-ordinateur sur lequel elle sera installée : installation/désinstallation de logiciels de production, ajout/retrait de fonctions matérielles, etc. Ainsi, le système d'exploitation fourni par Microsoft sous le nom commercial de Windows est une distribution sur base d'un noyau Microsoft !

On parle souvent de distribution Linux, car c'est la grande majorité des systèmes d'exploitation en termes de nombre. Ces distributions s'appuient sur :

- > un noyau Linux ;
- > les outils système GNU ;
- > de nombreux logiciels (souvent plusieurs milliers).

Mais il existe aussi des distributions BSD ou encore Hurd. Alors, si vous êtes curieux et que vous avez un peu de temps, allez voir !

La notion de « distribution » n'implique pas qu'il s'agisse de Logiciels libres ni que l'ensemble de logiciels puisse tout faire, depuis la micro-électronique jusqu'au serveur web ! Chaque distribution répond à des besoins précis et une philosophie particulière. Elles ont toutes une histoire et si vous êtes un utilisateur régulier de telle ou telle distribution, je vous invite à vous renseigner sur l'histoire de cette dernière et les motivations de ses auteurs. Cela vous aidera à mieux appréhender les choix techniques qui sont effectués au fil du temps et qui font que telle ou telle distribution sont des objets de vénération ou de rejet à certaines époques...

Pour information, le site Distrowatch.com (<http://distrowatch.com/>) est dédié aux distributions à base de Logiciels libres. On compte tout de même plusieurs centaines de distributions !

Comment les différencier ? Dans le cas des distributions à base de Logiciels libres, un des éléments essentiels et fondateurs est le choix des paquets. C'est le format sous lequel vont être présentés les logiciels pour s'installer sur le micro-ordinateur. Il existe plusieurs formes possibles et qui ne sont pas forcément compatibles ! Soyez attentif, car chaque format de paquets s'accompagne de logiciels spécifiques pour vous aider dans la gestion quotidienne. Certains formats vont ainsi disposer d'une interface graphique en plus de l'interface traditionnelle en ligne de commande. De même, certaines distributions vous imposeront des paquets à compiler (les distributions que l'on appelle « source » comme **Gentoo**) tandis que d'autres vous proposeront au choix d'installer des logiciels déjà compilés pour votre plate-forme matérielle (appelés aussi binaires) ou les sources. La qualité des outils de gestion des paquets est essentielle dans le choix d'une distribution et dans sa qualité finale.

Ensuite, ce qui fait la différence entre les distributions est le rythme d'évolution. Certaines distributions vont évoluer très lentement (**Debian** par exemple) afin de garantir une très grande stabilité tandis que d'autres vont chercher à être en permanence novatrices (comme **Ubuntu** ou **Mandriva**).

Cela dépend des auteurs, mais aussi du marché visé par les sociétés qui se cachent parfois derrière. Les distributions les plus « novatrices » sont aussi celles à destination des utilisateurs finaux. Cependant, pratiquement toutes les distributions majeures insistent sur le fait qu'une version donnée fera l'objet pendant un temps déterminé d'un suivi régulier et garanti. Là encore, renseignez-vous avant de choisir (voir page 28)...

Sidoine Pierrel

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Qu'est-ce qu'un live-CD ?

Un live-CD (ou live-DVD, ça existe aussi et souvent pour les mêmes projets) est une distribution Linux (voir définition ci-contre) stockée sur un support (CD, DVD mais ça peut aussi être une clé USB ou un mini-disque dur à base de mémoire flash) capable de se lancer seule (via le BIOS de l'ordinateur) au démarrage et qui n'a pas besoin pour fonctionner d'être installée sur un disque dur ou tout autre support définitif. Cette distribution ne modifie en rien l'ordinateur sur lequel elle s'exécute à moins d'une demande expresse de l'utilisateur. Elle fonctionne en mémoire (minimum 256 Mo actuellement pour un live-CD Knoppix ou Ubuntu/Kubuntu) directement depuis son support amovible. Au redémarrage de la machine, tout reprend sa position initiale comme si rien n'avait jamais eu lieu ! Souvent, elle ne demande pratiquement aucun travail de la part de l'utilisateur pour fonctionner, car, depuis déjà quelques années, elle auto-détecte son environnement matériel. Parfois, elle peut proposer à l'utilisateur de choisir lors du démarrage entre plusieurs modes d'utilisation et certaines options (langue, environnement graphique, logiciels à lancer, etc.). C'est un système complet mais non permanent.

Le premier live-CD qui a connu la gloire dans le monde des Logiciels libres n'est pas le fruit des créateurs du concept. Le concept est né avec un mode d'installation possible de la distribution Slackware (<http://www.slackware.com>). Mais le live-CD le plus populaire sans conteste est une distribution Debian revisitée du nom de Knoppix. C'est le fruit du travail de Klaus Knopper et de son équipe. Ce qui a fait la popularité de Knoppix, c'est son intégration des dernières technologies disponibles en permanence et de manière fiable et sa très grande flexibilité. Cette distribution live a porté le concept à un niveau inconnu jusque-là !

Ainsi, la plupart des distributions proposent un live-CD/DVD pour tester ces dernières et faire leur promotion. Disposant d'un espace d'au moins 1 Go une fois qu'elle est lancée (sur la base d'un CD de 700 Mo) et de tous les logiciels pour un environnement graphique ou le test d'un serveur spécifique, le live-CD/DVD est un outil extraordinaire pour tester des logiciels, valider la pertinence d'une distribution ou d'un ensemble de logiciels donnés, ou encore la compatibilité d'une station ou d'un serveur sans prendre aucun risque. Cependant, du fait de sa grande flexibilité (les limites sont faibles !), les live-CD/DVD sont rapidement devenus des couteaux suisses qui vont de la distribution anti-virus à l'outil de réparation des stations ou serveurs. Knoppix, à elle seule, a donné naissance à plusieurs dizaines de variantes (linguistiques, d'usages ou de supports).

Sidoine Pierrel

## Qu'est ce que la (mémoire) swap ?

La mémoire swap est mécanisme de mémoire virtuelle. Le but de ce mécanisme est de permettre à des applications d'utiliser plus de mémoire vive que n'en contient physiquement la machine. Pour réaliser cela, l'astuce consiste à utiliser un support de stockage de masse (disque dur) pour simuler la mémoire.

Très grossièrement, lorsqu'une application demande plus de mémoire qu'il n'en existe de disponible, le système détermine quelle application utilise la mémoire à un instant donné. Plutôt que d'allouer des morceaux de mémoire à toutes les applications, il met alors en veille certaines d'entre elles et copie sur le disque une image de la mémoire qu'elles utilisent. Il libère ainsi des morceaux de mémoire vive pour l'application gourmande.

Lorsqu'il est temps pour d'autres applications de fonctionner, le système fait l'opération inverse. Il met en veille d'autres applications, copie une image de la mémoire qu'elles utilisent sur le disque, charge l'image de la mémoire des applications en attente et les laisse fonctionner pour un temps. Le processus se poursuit jusqu'à ce que la ou les application(s) gourmande(s) se terminent. Il n'est alors plus nécessaire de faire appel à la mémoire virtuelle et les applications restantes se partagent normalement la mémoire physique.

Il faut bien comprendre que ce mécanisme a un coût pour le système. La mémoire virtuelle n'est largement pas aussi rapide que la mémoire physique. Les lectures/écritures intensives sur le disque, les opérations de mise en veille et reprise de l'exécution des applications et l'organisation même du système provoque une forte consommation du processeur et du disque.



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

De nos jours, avec des configurations possédant souvent plus de 512 Mo de mémoire physique, si votre système GNU/Linux se met à « swapper », c'est que quelque chose ne va pas. Soit, il s'agit d'un dysfonctionnement d'une application, soit vous en demandez trop à la machine (Gnome/KDE + un document OpenOffice de 300 pages + Blender + The Gimp et une image de 75 Mo + Firefox avec 25 onglets de pages pleines de Flash = un utilisateur qui confond GNU/Linux avec le système miracle qui sauvera le monde).

Comment choisir la taille de la partition dédiée au swap lors de l'installation ? Une réponse courante et erronée est que la taille du swap doit être le double de la mémoire physique. Lorsque les ordinateurs ne possédaient que 32 ou 64 Mo de mémoire vive cette réponse était valable. Aujourd'hui, ce n'est plus le cas. La bonne réponse est : ça dépend de vos besoins.

Habituellement 512 Mo de swap est suffisant pour pallier un problème temporaire d'engorgement de la mémoire. Si le problème de l'utilisation de la mémoire virtuelle est récurrent, il faut alors se pencher vers d'autres solutions comme l'ajout de mémoire, la réévaluation de vos besoins ou le changement de vos habitudes. Un système qui swappe tout le temps n'est pas utilisable et caractérise forcément un problème, qu'il s'agisse de 128 Mo de mémoire virtuelle ou de 4 Go.

Dans tous les cas, si vous avez subitement (mais temporairement) besoin de plus de mémoire virtuelle, par exemple pour un rendu gourmand en mémoire avec Blender ou POV-Ray, vous pouvez facilement en ajouter (en tant que *root* bien sûr) :

Avant opération :

```
# free -m | grep Swap
Swap:  494  95  427
```

Nous disposons de 494 Mo de mémoire virtuelle. Commençons par créer un fichier *daswap* (le nom importe peu) pour le swap :

```
# dd if=/dev/zero of=daswap bs=1024 count=30000
30000+0 enregistrements lus
30000+0 enregistrements écrits
30720000 octets (31 MB) copiés, 0.318773 seconde, 96.4 MB/s
```

Nous créons là un fichier de 30000 blocs de 1024 octets, soit 30 Mo de zéros. Nous initialisons ensuite le fichier pour pouvoir l'utiliser comme mémoire virtuelle :

```
# mkswap daswap
Setting up swapspace version 1, size = 30715 kB
no label, UUID=967a3eb8-7907-4343-ac1c-5cbe19d86005
```

Pour que le système soit content et que nous ayons un minimum de sécurité, nous changeons les permissions sur le fichier en le rendant lisible et inscriptible uniquement pour le propriétaire (*root*) :

```
# chmod 0600 daswap
```

Enfin, nous signalons au système qu'il doit l'utiliser comme mémoire virtuelle :

```
# swapon -v daswap
swapon on daswap
```

On vérifie :

```
# free -m | grep Swap
Swap:  523  95  427
```

Enfin, lorsque nous n'en avons plus besoin :

```
# swapoff -v daswap
swapoff on daswap
# rm daswap
```

Swapper dans un fichier est un peu plus lent que sur une partition dédiée, mais le mécanisme est de toute façon lent à la base. Nous pouvons utiliser ainsi n'importe quelle taille de fichier swap et même plusieurs fichiers si nécessaire. Pensez simplement à placer le fichier sur un disque (rapide de préférence) et non dans un répertoire monté en mémoire vive, comme c'est souvent le cas pour */tmp*.

Enfin, je tiens à signaler que le manque de mémoire ne signifie pas nécessairement le plantage du système. Le noyau ayant les pleins pouvoirs, même en cas de carence en mémoire, celui-ci va tuer les processus dont la demande ne peut être satisfaite. Il est donc possible que le problème se résolve de lui-même au bout d'un certain temps si la configuration le permet.

Denis Bodor

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

# Qu'est-ce qu'une bibliothèque ?

Chaque application (votre navigateur de fichiers, votre client de messagerie, votre navigateur Web, votre traitement de texte, etc.) sous sa forme exécutable fait appel à un ensemble de bibliothèques de sous-programmes. Chacun de ces sous-programmes est capable d'exécuter une fonction bien spécifique : calculs mathématiques, écriture/lecture de données, etc.

Très souvent appelées « librairies » par abus de langage (erreur de traduction du terme anglophone *library*), les bibliothèques sont ainsi indispensables au bon fonctionnement d'une application. D'ailleurs, lorsque vous installez un nouveau paquet de logiciels à l'aide de votre gestionnaire de paquets (ou en ligne de commande), vous constatez que bien souvent l'installation dudit paquet requiert l'installation conjointe d'une ou plusieurs bibliothèques. Autre cas de figure : vous venez de récupérer un exécutable sur un site quelconque, mais, à son lancement, celui-ci refuse de démarrer en vous signalant qu'il vous manque telle ou telle bibliothèque.

Il arrive que les bibliothèques soient directement incluses dans l'image du fichier exécutable d'une application. Dans ce cas, cette dernière est totalement autonome et on parle de « bibliothèque statique ». Parallèlement, il existe des « bibliothèques dynamiques » (aussi appelées « bibliothèques partagées ») : dans ce cas, les sous-programmes ne sont pas inclus dans le fichier exécutable de l'application : en revanche, leur présence est absolument nécessaire au moment de l'exécution ; il faudra donc les installer au préalable. Plusieurs applications peuvent avoir besoin des mêmes sous-programmes, c'est pourquoi l'utilisation de bibliothèques partagées est très pratique et très répandue. En outre, l'utilisation de ces bibliothèques externes permet d'alléger considérablement la taille du fichier exécutable.

Les différentes bibliothèques installées sur votre système se trouvent dans les répertoires `/lib` et `/usr/lib`. Une bibliothèque statique est un fichier d'extension `.a` (pour *archive*) ; une bibliothèque dynamique a pour extension `.so` (pour *shared object*). Le nom complet d'une bibliothèque partagée se termine par son numéro de version, sous la forme `majeur.mineur` (correspondant respectivement à des modifications majeures ou mineures de version de la bibliothèque). Par exemple, la bibliothèque `libkdeprint.so.4.2.0` désigne la révision de majeur 4 et de mineur 2.0 de la bibliothèque `kdeprint`. Si le développeur de la bibliothèque effectue une modification importante de la bibliothèque (ex. : l'ajout d'une fonction supplémentaire), alors la nouvelle version sera distribuée avec un numéro majeur supérieur. En revanche, si la modification est minime, c'est le numéro mineur qui sera supérieur à celui de la version précédente et le majeur restera le même.

Comme n'importe quel logiciel, les bibliothèques évoluent et sont actualisées plus ou moins fréquemment. Une application peut fort heureusement utiliser une bibliothèque plus récente que la version avec laquelle elle a été compilée, à la condition que le numéro majeur soit le même. C'est pourquoi, dans vos répertoires de bibliothèques, vous remarquerez la présence de nombreux liens symboliques, un pour chaque majeur, pointant vers la bibliothèque la plus récente compatible avec le majeur. Par exemple :

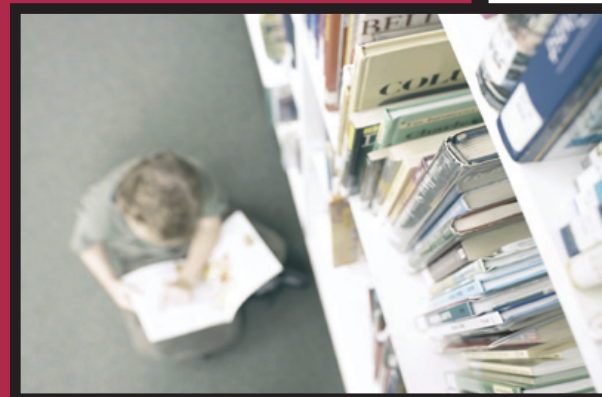
```
lrwxrwxrwx 1 root root      16 2006-06-08 11:50 libwrap.so.0 -> libwrap.so.0.7.6
-rw-r--r-- 1 root root     27052 2006-03-01 18:28 libwrap.so.0.7.6
```

Comme précisé plus haut, les bibliothèques sont indispensables voire essentielles, comme la célèbre bibliothèque `libc`, la bibliothèque standard du langage C. Sans elle, les commandes du système ne fonctionneraient plus ! En effet, la `libc` est utilisée par l'ensemble des applications. Son rôle est primordial.

Pour connaître la liste des bibliothèques utilisées par un fichier exécutable, vous pouvez saisir la commande suivante : `ldd chemin/vers/executable`. Par exemple :

```
~$ ldd /usr/bin/evince
linux-gate.so.1 => (0xffffe000)
libdjbvulibre.so.15 => /usr/lib/libdjbvulibre.so.15 (0xb7db0000)
libkpathsea.so.4 => /usr/lib/libkpathsea.so.4 (0xb7da3000)
libgnomeui-2.so.0 => /usr/lib/libgnomeui-2.so.0 (0xb7d1d000)
[...]
```

La liste est encore longue : pas loin d'une cinquantaine de bibliothèques sont utilisées par le lecteur PDF `Evince` ! Le plus important est de veiller à ne pas effacer par erreur une bibliothèque utilisée par un ou plusieurs programme(s)...



Fleur Brosseau

## Qu'est-ce que le shell ?

Entre le cœur de votre machine UNIX, le *kernel*, et vous, utilisateur, se trouve le *shell*. Le shell est un programme qui sert d'interface de dialogue entre l'utilisateur et le système, une sorte de « coquille » (d'où son nom anglais) autour du noyau. Le dialogue se fait par les commandes, que l'utilisateur saisit. Le rôle du shell est alors de lire la commande, de l'interpréter, de l'exécuter et éventuellement d'en donner le résultat.

Vous l'aurez compris, le shell, ce n'est ni plus ni moins que le mode texte, la bonne vieille ligne de commande, qui, pour certains, rappelle les temps anciens, avant que les applications graphiques rendent *a priori* plus aisée l'utilisation d'un PC. Pourquoi a priori ? Parce que le shell est un instrument puissant, comme on va le voir, qui permet de réaliser des tâches qu'on ne peut même pas espérer concevoir en mode graphique. Et puis aussi parce que la ligne de commande est plus pratique parfois qu'une interface graphique dont il faut explorer les multiples recoins avant de trouver ce qu'on cherche.

Chaque utilisateur du système possède un shell par défaut, dont l'identité est précisée dans le fichier de configuration `/etc/passwd`. Le shell est démarré automatiquement dès que vous vous loguez, et pour cause : sans lui vous ne pourriez exécuter aucune commande ! C'est d'ailleurs lui qui vous invite à passer ces dernières si vous démarrez une console, grâce à ce qu'on appelle le *prompt*, qui se présente sous la forme suivante (la syntaxe du prompt peut varier selon le shell utilisé) :

Si vous êtes un utilisateur normal :

```
nom_utilisateur@nom_machine:~$ ou encore nom_machine:nom_utilisateur:$
```

Si vous agissez en tant que *root* :

```
root@nom_machine:/home/nom_utilisateur# ou encore nom_machine:/home/nom_utilisateur:#
```

On notera d'ailleurs que la différence de symbole entre `$` et `#` permet de savoir rapidement si vous êtes logué en tant que *root* ou non.

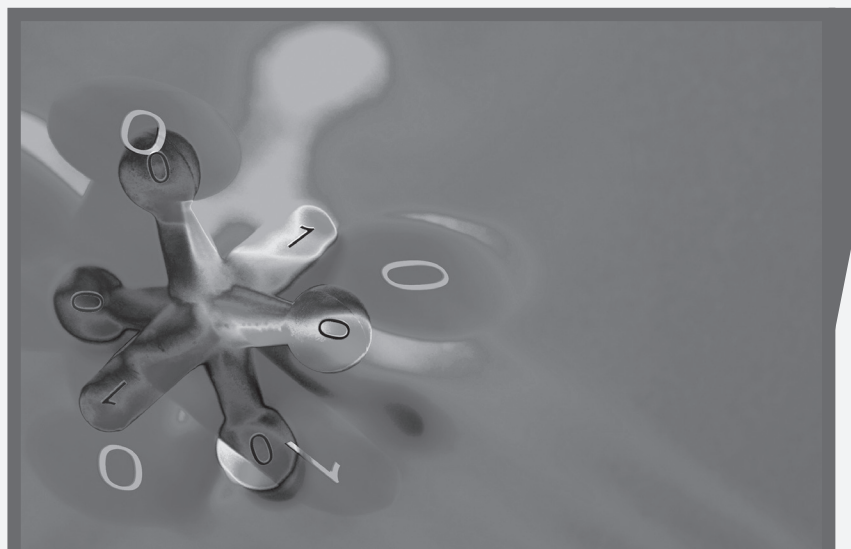
Une fois votre commande tapée, c'est au shell que revient la responsabilité d'interpréter et d'exécuter celle-ci. On parle d'usage interactif du shell, parce que l'utilisateur dialogue (interagit) avec le système. Le prompt offre de multiples possibilités. Vous pouvez naviguer dans vos répertoires par exemple :

```
ordino:moi$ ls Documents/GRECO/
Agony.jpg      View_of_Toledo.jpg
The_adoration_of_the_Shepherds.jpg the_
miracle_of_christ_healing_the_blind.jpg
ordino:moi$
```

Dans ce cas précis, le résultat de votre commande est affiché : vous visualisez l'ensemble du contenu du répertoire `/home/moi/Documents/GRECO`. Une fois la commande exécutée, le prompt réapparaît et vous pouvez en passer d'autres.

Vous pouvez aussi lancer des programmes, en tapant le nom du programme, puis [Entrée] :

```
ordino:moi$ xedit
```





Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>  
<http://www.unixgarden.com>  
<http://www.editions-diamond.fr>



Dans ce cas, le shell ne rendra la main que lorsque le processus en cours sera achevé : en d'autres termes, le prompt ne réapparaît pas. Cela peut être ennuyeux d'ailleurs si vous voulez lancer d'autres commandes sans ouvrir une autre console. Dans ce cas, il faut ajouter une esperluette à la fin de la ligne de commande, et le prompt revient :

```
ordino:moi$ xedit &
ordino:moi$
```

Mais le shell ne se résume pas au prompt. Il a également pour fonction d'être utilisé comme langage de programmation. Il peut en effet se révéler fastidieux d'avoir à passer la même suite de commandes que vous utilisez régulièrement, surtout si celle-ci est longue. Vous avez alors la possibilité de créer ce qu'on appelle un « script shell » ou « fichier de script », où ces commandes sont conservées. Vous n'avez alors plus qu'à rappeler le nom du fichier pour exécuter le programme.

Bien connaître son shell revient donc à optimiser les possibilités de son système. Un script shell vous permettra par exemple d'automatiser des tâches de sauvegarde, de démarrer automatiquement votre programme préféré, etc.

Et puis, il faut tout de même rectifier une erreur : il n'y a pas UN mais DES shells, en raison de l'évolution constante des systèmes UNIX. *Bourne shell*, *C shell*, *Korn shell*, *Tenex C shell*, etc. sont autant de shells qui se ressemblent beaucoup en tant qu'interfaces de dialogue, et dont les différences sont surtout liées à leurs possibilités de programmation. Le plus connu reste toutefois le *Bourne Again shell* (dont le nom est abrégé en *bash*), pour la bonne et simple raison qu'il est installé par défaut sur la plupart des distributions GNU/Linux.

Lucie Robin

## C'est quoi un fork ?

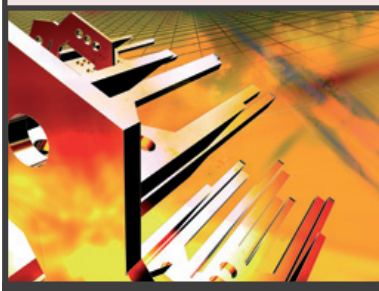


*Fork* est un terme anglais qui signifie littéralement « fourchette ». Quel rapport avec l'informatique ? Eh bien, c'est en fait la connotation d'embranchement (une même racine pour plusieurs branches) qui nous intéresse ici. Un fork désigne donc un projet informatique qui possède une racine commune avec un autre projet. Ces deux projets peuvent suivre cependant des voies de développement tout à fait différentes. On peut aussi parler de « projet dérivé ».

Comme vous pouvez le deviner, pour qu'un fork voit le jour, il est nécessaire que les auteurs du projet initial aient mis leurs sources à disposition et autorisé la modification, l'utilisation et la redistribution de ce code source. C'est pourquoi on entend souvent parler de fork dans le monde des Logiciels libres. Malheureusement, l'apparition d'un fork entraîne souvent la mort du projet initial, peut-être moins abouti ou dont l'équipe de développement est moins active. Seuls les projets les plus performants et les plus « séduisants » aux yeux des utilisateurs survivent. Ainsi, les forks permettent en quelque sorte une sélection naturelle parmi la profusion de Logiciels libres...

L'un des forks les plus connus est sans doute le serveur graphique **X.Org**, qui est basé sur la version 4.4 RC2 d'**XFree86**. Et aujourd'hui, la plupart des distributions intègrent X.Org au lieu d'XFree86... On peut également citer le célèbre logiciel de dessin vectoriel **Inkscape**, qui, à l'origine, est un fork de **Sodipodi**.

Dans un système UNIX, *fork* désigne également une fonction permettant à un processus de se dupliquer, en parallèle au premier. Le processus initial est ensuite désigné en tant que processus-père, la duplication est un processus-fils. Ce dernier hérite de tous les attributs système du père. La fonction *fork* est par exemple très utilisée dans les applications client-serveur avec plusieurs clients simultanés.



Cette fonction peut d'ailleurs être à l'origine de bien des problèmes, puisqu'elle entre dans la composition de ce que l'on appelle une « *fork bomb* » (aussi connu sous le nom de « *logic bomb* »). Une *fork bomb* est une forme d'attaque sur votre système : il s'agit d'un programme qui s'auto-réplique en créant un grand nombre de processus très rapidement

jusqu'à saturation de l'espace disponible, ce qui entraîne la plantage de votre système d'exploitation (quel qu'il soit). Aucun nouveau programme ne peut démarrer, les programmes actifs ne peuvent plus être utilisés (car le processeur et la mémoire sont saturés).

La seule solution permettant de venir à bout d'une *fork bomb* étant de détruire toutes ses instances, une fois la *fork bomb* activée, vous n'avez plus qu'à redémarrer votre système ! Cela dit, le nombre maximal de processus pouvant être exécutés par un programme ou un utilisateur est le plus souvent limité (via le fichier `/etc/security/limits.conf` par exemple). Ce qui réduit considérablement l'impact d'une telle attaque...

Fleur Brosseau

## C'est quoi une commande ?



De manière générale, on peut dire qu'une commande, c'est un programme, qui va être exécuté dès lors que vous l'appellez. Mais la particularité de ces programmes, c'est qu'ils nécessitent d'être décodés par un interpréteur de commandes particulier : le *shell* (voir page 40), qui n'est lui-même qu'un programme servant d'interface entre vous et votre système. Quand vous ouvrez une console, votre shell est inactif. Mais quand vous tapez une commande, le shell va réagir en cherchant dans le système s'il est possible d'exécuter ce que vous lui avez demandé.

Cela dit, il faut préciser qu'il existe différents types de commandes. On distinguera :

- > les commandes internes au shell : ce ne sont pas vraiment des programmes, mais des fonctions intégrées au *kernel*. Le shell les exécute directement, sans avoir à chercher dans le système les instruments de leur exécution. Un exemple : la commande `cd`, qui permet de changer de répertoire ;

- > les programmes binaires exécutables : ce sont tous les programmes que contient votre système, dont le chemin d'accès est précisé grâce à la variable `PATH`. Le shell les trouvera dans les répertoires `/usr/bin`, `/bin` et dans les différents `/home/nom_utilisateur/bin` ;

- > les scripts shell (voir page 51), dont il faut cette fois préciser au shell le chemin d'accès ;

- > les alias, commandes « raccourcies » définies par les utilisateurs.

Une fois la commande tapée, le shell va agir de manière très ordonnée pour l'interpréter. Il regarde d'abord s'il s'agit d'une commande interne ; si c'est le cas, il l'exécute, sinon il passe à l'étape suivante, c'est-à-dire la recherche du chemin d'accès à la commande dans les différents répertoires. Lorsqu'il trouve la commande, il l'exécute. Et s'il ne la trouve pas ? Alors, dans ce cas, il renvoie un message d'erreur :

```
~$ cot
bash: cot: command not found
~$
```

Avec l'habitude, on se rend compte que le message d'erreur *Command not found* est assez courant. Et pour cause, il arrive très souvent que l'on orthographie mal le nom de la commande. L'exactitude s'impose donc. Elle ne se limite d'ailleurs pas à l'écriture du nom de la commande : la casse (majuscules/minuscules) et l'ordre des différents paramètres de la commande entrent en ligne de compte. Ainsi, on veillera :

- > à ce que le nom de la commande soit correctement orthographié ;

- > aux différentes options possibles. On peut modifier le résultat de presque toutes les commandes grâce aux options dont elles disposent. Une option est précisée immédiatement après le nom de la commande et précédée d'un tiret (-). Par exemple, la commande `ls` va avoir pour résultat de vous lister le contenu d'un répertoire ; mais `ls -l` va en plus vous donner, pour chaque contenu, un certain nombre de précisions (permissions, propriétaire, taille...). On notera aussi que si l'on souhaite activer plusieurs options, on peut souvent les regrouper derrière le même tiret : `ls -la` va avoir le même résultat que `ls -l -a`.

- > aux arguments. Certaines commandes nécessitent qu'on leur précise ce qu'elles doivent faire et avec quoi : c'est ce qu'on appelle les arguments, dont la mention vient après celle du nom de la commande et de ses éventuelles options. Par exemple, pour copier un fichier dans un répertoire, il faut bien évidemment préciser quel fichier on veut copier, et où le mettre : `cp mon_fichier.txt /Mon-répertoire-de-destination`. Ça a l'air simple comme ça, mais ça ne l'est pas toujours ; notamment l'ordre des arguments est important : `cp /Mon-répertoire-de-destination mon_fichier.txt` ne vous apportera qu'un message d'erreur...

Pour connaître la syntaxe d'une commande, le mieux est encore de taper un petit `man nom_de_la_commande`, où tout vous sera précisé !

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Qu'est-ce que UTF-8 et Latin1 ?

UTF-8 et Latin1 sont deux locales que votre système libre préféré supporte certainement. Mais que viens-je donc de dire ?

Dans le monde de l'informatique en général, et de GNU/Linux en particulier, où tant de choses se gèrent grâce à des fichiers textes, une guerre fait rage dans le murmure des clavotements et les indignations étouffées en forme de : « Mais ils sont où mes é !? ». Cette guerre a commencé lorsque les premiers utilisateurs non américains d'ordinateurs ont voulu écrire sur leurs machines et dans leur langue maternelle (ils auraient dû fuir, pauvres fous...). En effet, vu la nationalité des créateurs des premiers PC, le premier jeu de caractères disponible dans les systèmes d'exploitation de ces machines ne contenait que les caractères utiles aux Américains pour communiquer... Il s'agit de la fameuse table de caractères ASCII, pour *American Standard Code for Information Interchange*.

Or, que se soit pour écrire en français, en allemand ou en norvégien, les possibilités du jeu de caractères normalisé trouvent vite leurs limites. Il y manque, par exemple, les caractères accentués, le ß allemand et le Å rond en chef norvégien, Å.

Pour pallier ce problème, chaque pays a, dans un premier temps, fait standardiser son propre jeu de caractères en incluant les symboles locaux qui manquaient jusque-là. C'est comme ça qu'une version de la norme ISO-8859 est apparue pour chaque pays dont l'alphabet complet pouvait se résumer à la table ASCII plus 127 caractères. Ainsi, la norme **ISO-8859-1** correspond à l'alphabet enrichi utile aux pays de l'Europe occidentale (comme la France, l'Allemagne, la Norvège...). C'est elle qui est aussi connue sous le nom de **Latin1**. Il est alors intéressant de noter qu'à l'arrivée de la monnaie unique en Europe, ce codage dû être repensé pour inclure le symbole €. On arrive alors à la norme **ISO-8859-15**, ou **Latin9**.

Mais comme tous ces jeux de caractères reposent sur un codage (correspondance entre valeur entière et symbole alphanumérique) de 8 bits (les 7 premiers bits de la table ASCII standard, et un 8e bit pour coder 127 caractères nationaux supplémentaires), ils sont tous limités à 255 caractères. Or, il est des problèmes posés par le codage des caractères d'écriture que la simple multiplication des sets de lettre ne suffit pas à résoudre...

En effet, dans le cas des langues asiatiques (chinois, japonais, coréen, tibétain, etc.), disposer de 255 modalités de codage se révèle bien insuffisant ! Et comment faire si l'on souhaite pouvoir écrire un texte où du russe se mélange à du grec ancien ?

Cette fois, il fallait voir plus grand. Et c'est dans cette optique qu'une table Unicode fut normalisée pour la première fois en 1991 par l'Organisation Internationale de Normalisation sous l'index ISO/CEI 10646, avec pour but de regrouper, sans limite d'espace, tous les caractères connus sur la planète, soit, en 2005 avec la version 4.1.0 de la table, un total de près de 245 000 éléments.

Une fois ces symboles répertoriés, le problème devint alors de trouver un codage adapté à une telle table. Et c'est pour répondre à ce besoin que le format universel de transformation des caractères Unicode (UTF) fut mis au point. Là encore plusieurs versions ont été normalisées, **UTF-8** par exemple, est un système de codage relativement complexe, qui a la particularité de coder les 127 premiers caractères de la même manière que la table ASCII originelle. Il conserve donc une compatibilité ascendante, merveille de l'informatique, permettant à tous les ordinateurs du monde de continuer à voir les anciens caractères présents dans les nouveaux textes. Mais ce UTF-8 n'est malheureusement pas parfait. Son inconvénient principal est qu'il utilise plusieurs octets pour coder les lettres après la 127ème. Ainsi les lettres d'un même texte ne sont pas codées avec un même nombre d'octets d'un bout à l'autre (ce qui complique grandement les opérations de repérage dans un texte, d'un point de vue algorithmique), et, par exemple, les « lettres »

asiatiques se retrouvent codées sur 3 octets avec UTF-8, contre seulement 2 pour UTF-16 (Autre norme UTF). Un texte asiatique prend donc moins de place en UTF-16 qu'en UTF-8. Présentons donc rapidement UTF-16 alors. C'est une norme de transformation qui se limite à coder le plus de caractères possibles avec au minimum 16 bits. Enfin, nommons UTF-32, codage qui, lui, couvre toute la table Unicode, mais consomme a priori quatre fois plus de place pour coder un texte américain que son homologue UTF-8. Vous comprenez donc maintenant pourquoi le débat sur le choix de son codage est animé...

Toutefois, et bien qu'il mit quelques années à s'imposer, le format UTF-8 est désormais supporté par tous les principaux navigateurs web et la majorité des systèmes d'exploitation. Sous GNU/Linux, il tend à se généraliser comme locale par défaut pour coder vos textes simples, fichiers de configuration et noms de fichiers sur le disque dur.

```
!"#$%&'()*+,-./
0123456789:;<=>?
@ABCDEFGHIJKLMNO
PQRSTUVWXYZ[\]^_
`abcdefghijklmno
pqrstuvwxyz{|}~
```

Les 95 caractères ASCII affichables, espace blanche comprise



Simon Descarpentries

Exemplaire offert par Les Éditions Diamond

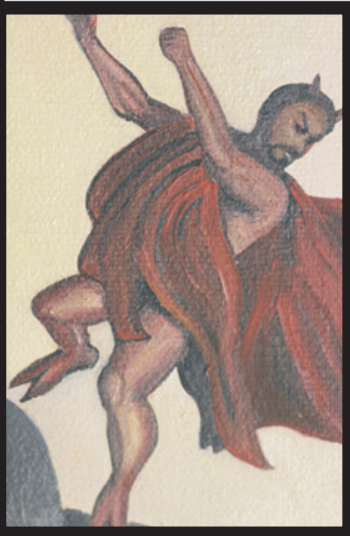
<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Qu'est-ce qu'un démon (ou daemon) ?

Le terme *daemon*, si l'on en croit le Jargon Français, viendrait de l'expression *Disk And Execution MONitor*. En réalité, le terme « démon » aurait été suggéré par les inventeurs d'Unix pour décrire un processus non invoqué manuellement, qui s'exécute en tâche de fond et qui ne demande aucune intervention de l'utilisateur. En effet, un démon n'est rien d'autre qu'un programme qui s'exécute en arrière-plan de votre système (et ce, quel que soit le système d'exploitation). Vous n'avez aucun contrôle direct sur lui, et vous ne vous rendez même pas compte qu'un démon est en train de tourner.



Un démon exécute une tâche prédéterminée à intervalles réguliers ou à la suite d'un événement. Un bon exemple de démon capable de répondre à un événement est un serveur Web qui attend qu'un client soumette une demande. Le serveur Web dort en arrière-plan jusqu'au moment où un client se connecte et demande une page Web. Le démon se réveille alors et transmet la page au client.

Considérons l'exemple du démon `lpd` (*Line Printer Daemon*), qui gère toutes les tâches d'impression sous Linux. Lorsque vous soumettez une demande d'impression, le démon `lpd` lance une autre instance de lui-même. Cette instance transmet le fichier et ses paramètres d'impression à l'imprimante appropriée. Ainsi, l'instance originale de `lpd` peut attendre en arrière-plan d'autres travaux d'impression et les placer en file d'attente.

`Cron` est un exemple de démon qui effectue une opération à intervalles spécifiés. Ce démon programme les tâches des utilisateurs et des tâches associées au système. Lorsqu'un utilisateur programme une tâche périodique (par exemple, le nettoyage du répertoire `/tmp`), le démon se réveille à intervalles réguliers et lance les processus requis pour exécuter cette tâche.

La majorité des démons gèrent les services réseau TCP/IP (`nfs`, `samba`, `apache`, `ssh`, etc.), le gestionnaire d'imprimantes, le système dans son ensemble (`cron`, `syslog`, `ntp`, etc.), etc. La plupart des démons ont un nom, qui se termine typiquement par la lettre `d`, comme `crond`, `httpd`, `lpd`, etc. Mais il en existe d'autres comme `samba`, `nfs`, `ssh`.

C'est dans le répertoire `/etc` que se nichent les démons de votre système, dans divers fichiers. Généralement, les démons sont lancés automatiquement dès le démarrage de votre système, via les scripts placés dans le répertoire `/etc/init.d` et en fonction de leur niveau de démarrage (répertoires `/etc/rcX.d`). À l'inverse, lors de l'arrêt du système, on dit que les démons sont « tués » (arrêtés).

Les répertoires `rcX.d/` contiennent tous les services qui doivent être lancés au démarrage (« scripts de démarrage »), sous forme de liens symboliques qui lancent ou arrêtent les démons. Exemples :

```
/etc/rc2.d$ ls -l
```

```
S20samba -> ../init.d/samba
```

```
S89cron -> ../init.d/cron
```

Les démons sont lancés dans l'ordre numérique. Ci-dessus, le démon `samba` est par exemple lancé avant `cron`. Constatez que les démons se trouvent en fait dans le répertoire `/etc/init.d`.

Il existe un démon spécial, `inetd`, nommé « super démon ». C'est le démon principal du réseau. Il permet principalement de lancer d'autres démons à la demande et de gérer/diminuer la consommation des ressources. Les fichiers `/etc/services` et `/etc/inetd.conf` sont les fichiers de configuration de ce démon. Il fonctionne uniquement avec les services réseau.

La gestion des démons est une partie fondamentale de la gestion d'un serveur. L'administrateur doit s'assurer que les démons sont fonctionnels et que le nombre de démons actifs sur un serveur n'est pas trop élevé, sans quoi ce dernier risque d'être surchargé.

Fleur Brosseau

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Comment plusieurs programmes peuvent-ils s'exécuter simultanément sur ma machine ? Qu'est-ce qu'un processus ?

Les systèmes d'exploitation modernes permettent à plusieurs programmes de fonctionner « en même temps ». Cette expression est mise entre guillemets, car généralement (sauf pour le cas des machines multi-processeurs ou des processeurs multi-cœurs), seul un programme peut être exécuté par la machine à un instant donné. Comment le système d'exploitation peut-il alors exécuter plusieurs programmes en même temps ? Tout simplement en allouant le processeur à un programme pendant un quantum de temps très court, puis en le faisant passer à un autre, et ainsi de suite. Le délai très court alloué à chaque programme donne l'impression que tous fonctionnent en même temps, même s'il n'y a qu'une seule ressource processeur dans la machine.

Un **processus** est l'état d'un programme en cours d'exécution sur le système. À chaque programme en cours d'exécution (The Gimp, OpenOffice, Firefox, etc.) correspond au moins un processus. Ceux-ci sont constitués des éléments suivants :

- Le code du programme à exécuter, c'est-à-dire la forme chargée de l'exécutable correspondant au programme.
- L'espace mémoire dans lequel le programme travaille, contenant ses variables.
- Le numéro du processus (ou **PID**), permettant de le référencer de manière unique.
- L'état du processus, « actif » s'il s'agit du processus actuellement exécuté par le processeur, « endormi » sinon.

Le système d'exploitation peut ainsi préserver l'état des processus endormis pendant que le processus actif s'exécute. De plus, deux autres valeurs notables sont associées aux processus sous Linux :

- L'identifiant de l'utilisateur propriétaire du processus, généralement celui qui l'a lancé.
- La « courtoisie » (*nice*) du processus vis-à-vis des autres. Cette valeur est un entier compris entre -20 et 19 (la valeur par défaut est zéro). Plus cette valeur est petite, plus le processus sera prioritaire sur les autres. Seul l'utilisateur root peut utiliser des valeurs négatives.

On peut distinguer deux types de processus. Les processus « lourds » sont totalement isolés les uns des autres (ils disposent chacun de leur propre zone mémoire). Les processus « légers », au contraire, peuvent partager une zone mémoire en commun, ce qui s'avère pratique dans certains cas (par exemple, un processus chargé de collecter des données pendant qu'un autre les analyse).

La commande **ps** permet de lister les processus en cours d'exécution sur le système :

```
$ ps -A u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   1628  480 ?        Ss   Oct12   0:02 /sbin/init
...
user     5532  0.0  0.0   2508  992 pts/2    R+   15:14   0:00 ps -A u
```

Elle affiche beaucoup d'informations sur les processus, dont l'utilisateur propriétaire, le PID, le temps machine passé à exécuter ce processus, et sa commande correspondante. On voit logiquement que la commande que nous venons de taper est le dernier processus créé par le système. On constate également qu'il y a beaucoup de processus ! Vous pouvez essayer de déterminer lesquels correspondent à vos programmes en cours d'exécution.



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>



Il est possible de changer la courtoisie par défaut d'un processus avec la commande `nice` :

```
$ nice -n19 firefox &
```

lancera ainsi Firefox avec la priorité la plus basse, c'est-à-dire qu'il sera ralenti par le système si d'autres processus se montrent exigeants en ressources.

Enfin, un processus peut recevoir des signaux. Un signal permet de placer un processus dans un état déterminé : gelé, actif, voire de le tuer. La commande `kill` permet d'envoyer un signal à un processus. Elle nécessite de connaître son PID, que nous pouvons retrouver avec `ps`. Par exemple, si nous voulons geler le Firefox que nous venons de lancer, il nous suffit de récupérer son PID et de lui envoyer le signal `STOP` :

```
$ ps -A u
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT
START	TIME	COMMAND					

...

user	32682	4.2	10.6	307944	109660	?	Rt	09:31
15:12		/usr/lib/firefox/firefox-bin						

...

```
$ kill -STOP 32682
```

Notre Firefox est devenu totalement non réactif à nos actions, et pour cause : le système ne l'exécute plus. Le signal `CONT` permet de reprendre son exécution :

```
$ kill -CONT 32682
```

Enfin, les processus récalcitrants peuvent être éliminés sommairement en leur envoyant le signal `KILL` :

```
$ kill -KILL 32682
```

Ces commandes permettent de manipuler les 3 processus de manière primitive. Il existe également des interfaces graphiques à cet effet. **Ksysguard** sous KDE et **Procman** sous Gnome (généralement accessibles par la combinaison de touches [Ctrl]+[Échap]) qui permettent de lister et de manipuler les processus de manière graphique (changer la courtoisie, envoyer un signal, etc.).

Alexandre Courbot

## C'est qui root ?

*Root*, c'est le nom que l'on donne au premier utilisateur (dans l'ordre d'installation du système) d'un système UNIX. *Root* signifie « racine » en anglais. Il existe d'ailleurs aussi sous Mac OS X, mais il n'est pas visible. Il est l'équivalent de « l'administrateur » dans les versions 2000/XP/2003/Vista de Microsoft Windows.

*Root* est identifié par l'indice utilisateur `0` et appartient au groupe *root*, identifié par l'indice groupe `0`. Il a tous les droits sur le système. On ne doit pas changer son nom, ni ses identifiants numériques sous peine de ne plus pouvoir accéder à son système. Il peut tout faire, et ce, sans exception. Il est omnipotent sur le système et inamovible. À vous, si vous l'utilisez, d'être omniscient dans son usage, car il peut aussi tout détruire s'il est mal utilisé. Au final, il a plus de pouvoir que les administrateurs des systèmes graphiques concurrents, car il arrive sous certains de ces systèmes que l'administrateur ne puisse pas agir. Ici, il pourra toujours. Et c'est pour cela qui faut être TRÈS PRUDENT dans son usage...

Aujourd'hui, avec des programmes comme `sudo` (voir plus loin), on n'a pas véritablement besoin, une fois une station installée de se connecter avec l'utilisateur *root*. D'ailleurs, des environnements de bureau graphiques, comme KDE ou GNOME, vous le refuseront s'ils sont bien configurés (par défaut sur Ubuntu/Kubuntu par exemple). Donc, si l'on est un utilisateur « basique », on ne rencontrera jamais directement *root*. Il peut être utile pour accélérer des travaux de maintenance, mais en soit, il n'est pas nécessaire de l'activer dans la vie normale d'une station de travail ou même d'un serveur. Pour les distributions de type Ubuntu, un simple `sudo <ma commande et ses options>` sous le nom d'un utilisateur dûment autorisé (donc authentifié dans `/etc/sudoers`) revient au même !

De fait, sous certaines distributions Linux, on le désactive après l'installation en ne lui donnant pas de mot de passe défini par l'utilisateur (Ubuntu/Kubuntu par exemple). Cela complique la tâche de certains vieux habitués qui ont fait leurs armes avec des versions de Linux où l'usage du *root* était largement répandu. Pour information, on a même connu une version grand public au début des années 2000 qui loguait ses utilisateurs graphiques directement en *root* (*Corel Linux*). Autant dire que la sécurité n'était pas optimale, loin s'en faut !

Car *root* est dangereux du fait de son omnipotence. Et, depuis quelques années, la tendance est à limiter au strict nécessaire son usage. Comme je viens de vous le montrer, avec le programme `sudo`, on n'a en fait plus besoin de lui directement (cela ne signifie pas qu'on puisse l'effacer du système, surtout pas !). Cela protège en partie contre les erreurs...

La première des bêtises est d'effacer des données importantes. On ne peut le faire qu'avec l'utilisateur qui les a créées. C'est parfois frustrant, mais c'est pratique pour éviter des gros désagréments. La deuxième est le formatage d'un disque ou la désinstallation d'un logiciel. Là encore, le fait de passer par une commande un peu plus longue que la normale, et potentiellement bloquée par le fait que votre utilisateur n'a pas tous les droits, vous donne le temps de réfléchir avant d'agir (puis de regretter le cas échéant...).

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

Mais root est aussi dangereux si l'on peut prendre possession de lui à distance. C'est en cela que la distribution dont je parlais plus haut n'était pas sécurisée. Quiconque prend le contrôle du root ou d'un programme s'exécutant sous root, prend le contrôle de votre machine ! Le mot de passe de root doit donc être quelque chose de peu simple à trouver ou à casser le cas échéant. Et on doit utiliser ses pouvoirs avec une très grande économie.

Ainsi, il est formellement déconseillé de faire fonctionner certains programmes sous les droits de root (par exemple, les programmes de P2P ou encore des programmes comme le serveur Apache ou des serveurs de FTP). On doit soit les lancer avec un utilisateur spécifique (par exemple, votre utilisateur pour les programmes de P2P), soit si root est indispensable (pour accéder à certaines ressources de la machine), on dispose alors de moyens de protection que l'on appelle des « cages » ou des « prisons » (le programme `makejail` par exemple). Le programme s'exécute avec les droits de root, mais il est limité dans son espace et, en cas d'attaque, l'attaquant n'a accès qu'aux limites de la cage, c'est-à-dire rien ou presque (le logiciel et ses données). Ce type de sécurisation se fait en fonction du logiciel, en lisant bien la documentation. C'est un peu de travail en plus à l'installation du logiciel, mais, en cas de piratage, c'est BEAUCOUP de tracas en moins !

Sidoine Pierrel

## Qu'est-ce qu'une ABI ?

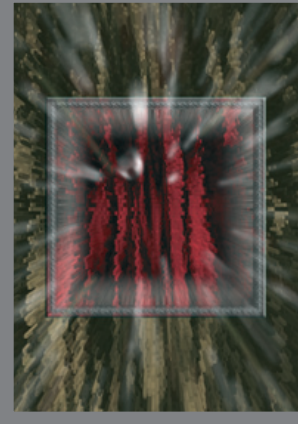
Un programme informatique est écrit dans un langage informatique donné. Le plus souvent en C ou en C++ pour un Logiciel libre. Une des avancées majeures est la modularisation des codes : ainsi, un programme peut utiliser des « bouts » de code qui existent déjà. En effet, pourquoi recoder toujours une bibliothèque qui lit des `.jpeg` quand la `libjpeg` le fait très bien ? Les fonctionnalités de base sont fournies par la `glibc`, qui fait notamment l'interface avec le noyau Linux.

ABI, *Application Binary Interface*, définit la façon dont les modules communiquent entre eux. Ceci est d'ailleurs défini dans les standards des langages de programmation. Malheureusement, ces standards sont complexes et souvent les compilateurs (programmes qui transforment le code source en exécutable) génèrent leur propre version de l'interface pour se simplifier la vie.

L'interface du langage C est assez stable, mais celle du C++ a évolué entre les différentes versions 3.x du compilateur libre `gcc` pour se rapprocher le plus possible du standard. Cela signifie que 2 modules de programmes C++ compilés avec différentes versions de `gcc` ont peu de chances de fonctionner correctement ensemble. Ces modules doivent donc être compilés avec la même version du compilateur. Cela n'est pas un problème avec une distribution GNU/Linux, car tous les logiciels sont compilés ensemble. Mais les problèmes arrivent avec les binaires propriétaires téléchargés sur Internet...

C'est pourquoi un « vieux » binaire propriétaire peut avoir du mal à fonctionner sur une distribution Linux récente. En effet, il essaiera de communiquer avec la bibliothèque fondamentale du système, mais dans un ancien langage. L'ABI que le programme utilise n'est pas compatible avec celle du système hôte. Donc forcément, cela ne marche pas... Pour un Logiciel libre, cela n'est pas gênant, car il suffit de prendre la dernière version d'un programme compilé pour sa distribution.

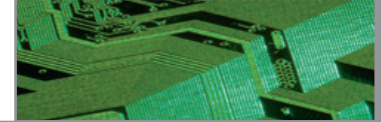
Nicolas Boulay



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com><http://www.unixgarden.com><http://www.editions-diamond.fr>

## Comment connaître son matériel ?



Votre ordinateur contient au minimum une carte mère et une carte graphique. Pour connaître les cartes supplémentaires dont vous disposez, une seule commande à retenir : `lspci` (cette commande fait partie du paquetage `pciutils`, à installer sur votre système si ce n'est déjà fait). La commande `lspci -n` permet de connaître diverses informations comme la classe, la marque, le modèle, etc. La commande `lspci -v` permet, quant à elle, de connaître diverses informations supplémentaires comme la fréquence du bus associé à la carte, les zones mémoires qu'elle occupe, etc.

Pour connaître la liste du matériel USB connecté à votre machine, il suffit de saisir la commande `lsusb` (qui fait partie du paquetage `usbutils`).

Pour obtenir toutes les informations concernant le (ou les) processeur(s) de votre système, utilisez la commande `cat /proc/cpuinfo`. Vous pourrez ainsi visualiser le nom du fabricant, le modèle, la fréquence, etc. Notez également que la commande `top` permet de visualiser l'activité du processeur en temps réel. Plus précisément, elle affiche la quantité de mémoire utilisée par chaque processus.

L'outil `hdparm` permet, quant à lui, d'agir sur les périphériques IDE [1]. En particulier, il permet de consulter les informations se rapportant à un disque, avec l'option `-i` suivie du nom du périphérique que vous souhaitez visualiser, par exemple (sous `root`) : `hdparm -i /dev/hda`.

C'est le disque dur qui est visualisé ici. La première ligne nous renseigne au sujet de la marque, du modèle et du numéro de série du disque dur. La mention `BuffType` renvoie le type de mémoire cache utilisée par le disque dur et la mention `BuffSize` renvoie sa taille (rappelons qu'une taille de mémoire cache élevée pour un disque dur permet d'en augmenter les performances).

La commande `hdparm -t` permet de tester la vitesse de lecture de votre disque. Ce sont les premiers secteurs du disque qui sont testés (les plus rapides), pendant 3 secondes. La valeur renvoyée n'est pas une moyenne, mais une valeur maximale de la vitesse.

```
hdparm -t /dev/hda
/dev/hda:
Timing buffered disk reads: 46 MB in 3.06
seconds = 15.02 MB/sec
```

On peut considérer que la vitesse d'un disque dur est généralement comprise entre 10 et 30 MB/sec.

Un outil permet de récapituler toutes les informations précédentes et bien d'autres encore. Il s'agit de la commande `lshw` (à lancer en tant que `root`). `LSHW` signifie *HardWare LiSter*, ce qui est on ne peut plus parlant. Cet outil affiche, de façon claire et détaillée, des informations concernant la mémoire, la version du `firmware`, la configuration de la carte mère, la version du CPU et sa vitesse, la configuration du cache, la vitesse du bus, les paramètres réseau, etc.

F. B.

[1] IDE est une interface standard permettant la connexion de périphériques de stockage sur un ordinateur, comme les disques durs, mais aussi les lecteurs CD-ROM et DVD.

## Comment connaître l'espace disque disponible sur mes différents systèmes de fichiers ?

Même si aujourd'hui la capacité des disques durs a considérablement augmenté, il est toujours utile de contrôler de temps en temps si l'espace disponible restant est suffisant. Il existe pour cela des commandes simples qui vous donneront rapidement les informations nécessaires.

Ainsi, la commande `df` (pour *disk free*) indique l'espace libre des disques contenant des systèmes de fichiers montés. Les tailles sont exprimées en kilo-octets. Cette commande peut s'accompagner de plusieurs options, parmi lesquelles :

- > `-h` affiche les tailles dans un format humainement compréhensible (par exemple, 140K 258M 19G) ;
- > `-l` limite l'affichage aux systèmes de fichiers locaux ;
- > `-i` affiche les informations sur l'utilisation des inodes [1] et non des blocs ;
- > `-k` affiche les tailles en kilo-octets (cette option peut être inutile selon la définition de la variable d'environnement `POSIXLY_CORRECT`. En effet, si celle-ci est configurée, la commande `df`, de même que la commande `du`, affichera le nombre de blocs de 512 octets).
- > `-T` affiche également le type de système de fichiers.

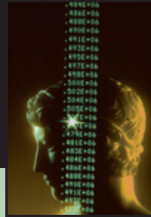
Pour obtenir l'intégralité des options disponibles ainsi que leur rôle respectif, je vous invite à consulter le manuel (saisissez `man df` dans votre terminal de commandes).



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>  
<http://www.unixgarden.com>  
<http://www.editions-diamond.fr>

## De quelle quantité de mémoire je dispose ?



Pour combler un éventuel manque de mémoire vive (RAM trop faible), les systèmes Linux utilisent une mémoire « virtuelle » pour simuler un bloc de mémoire supplémentaire. C'est ce que l'on appelle le SWAP. La mémoire totale disponible pour le système est la somme des deux. C'est le noyau qui gère le stockage en RAM ou en SWAP selon le cas.

Le commande `free` affiche les quantités totales de mémoire RAM et SWAP, libre et utilisée, ainsi que la mémoire partagée et les tampons utilisés par le noyau. L'option `-m` permet de présenter le résultat en Mo. Ainsi, pour afficher à l'écran la RAM et le SWAP disponible sur votre système, saisissez la commande :

```
free -m
```

L'option `-t` permet de retourner la somme des deux types de mémoire.

F. B.

## Comment est partitionné mon disque ?

Il peut être grandement utile de savoir comment est partitionné son disque si l'on souhaite faire cohabiter plusieurs systèmes d'exploitations ou plusieurs distributions. Votre environnement de bureau vous propose sûrement un utilitaire graphique pour visualiser le partitionnement de votre disque dur (comme un « gestionnaire de disques » ou assimilé). Néanmoins, il est très facile (et rapide !) de visualiser le partitionnement via la ligne de commande.

Ainsi, la commande `fdisk -l` permet de lister la table de partition du périphérique désigné. Si vous ne spécifiez pas de périphérique, `fdisk` utilisera le premier disque dur détecté.

Attention, soyez très prudent en utilisant cette commande, car elle permet également de repartitionner votre disque si vous utilisez l'option correspondante.

F. B.

La commande `du`, quant à elle, affiche le nombre de blocs d'1ko utilisés par une arborescence de fichiers (elle peut bien sûr être appliquée à l'ensemble de votre système de fichiers). Les principales options pour cette commande sont :

----> `-s` affiche uniquement le total ; cette option est conseillée dans le cas d'une grosse arborescence, sinon la commande `du` seule entraînera un défilement de lignes interminable... (une ligne pour chacun des répertoires, y compris les répertoires cachés).

----> `-k` affiche les tailles en kilo-octets (même remarque que précédemment).

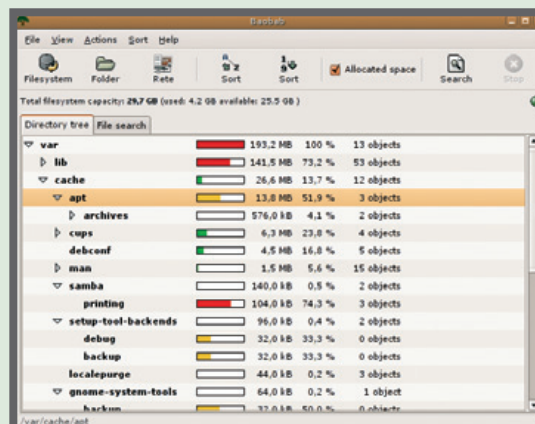
Pour finir, sachez qu'il existe une application graphique, **Baobab**, qui permet de visualiser l'occupation de votre espace disque. Vous pourrez ainsi identifier rapidement, puis supprimer les fichiers inutiles qui prendraient trop de place (site officiel : <http://www.marzocca.net/linux/baobab.html>).

F. B.

[1] Les inodes sont des blocs particuliers du disque dur regroupant les informations essentielles d'un fichier et assurant la liaison entre ce fichier et le reste du système. La quantité d'inodes indique le nombre maximum de fichiers que le système de fichiers peut contenir.

```

fleur@firefly:~$ df
Sys. de fich.      1K-blocs      Occupé Disponible Capacité Monté sur
/dev/hda1         19228276      7776928  18474608  43% /
varrun            128144         4      128140  1% /var/run
varlock           128144         4      128140  1% /var/lock
udev              128144        108      128036  1% /dev
devshim           128144         0      128144  0% /dev/shm
lrm                128144       18856     109288  15% /lib/modules/2.6.15-27-386/volatile
/dev/hda5         38448276     288308     36266868  1% /media/hda5
fleur@firefly:~$ df -h
Sys. de fich.      Tail.  Occ.  Disp. %Occ. Monté sur
/dev/hda1          19G   7,5G   10G   43% /
varrun             126M  140K  126M   1% /var/run
varlock            126M  4,0K  126M   1% /var/lock
udev               126M  108K  126M   1% /dev
devshim            126M   0     126M   0% /dev/shm
lrm                 126M  19M  107M  15% /lib/modules/2.6.15-27-386/volatile
/dev/hda5          37G   282M  35G   1% /media/hda5
fleur@firefly:~$
    
```



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Peut-on tout faire en lignes de commande ?

### En UNIX, tout est fichier

Il est courant d'entendre qu'avec GNU/Linux tout est possible à partir d'une simple invite de commande, et ce n'est pas faux. En effet, les possibilités des systèmes Unix (ancêtres de GNU/Linux), étaient en avance sur leur temps lorsque les ordinateurs étaient encore tous uniquement accessibles à partir d'une console en mode texte. L'administration de ces machines était alors entièrement à la portée d'un éditeur de texte, puisque toutes les configurations étaient stockées dans de simples fichiers textes.

On comprend alors mieux que le débat ait pu porter à l'époque sur le choix de son éditeur de texte, en mode console, et non sur celui de son environnement graphique préféré...

### Des shells puissants

Mais revenons en 2006. Que reste-t-il aujourd'hui de cet héritage ? Il en reste tout d'abord un shell puissant autour du noyau Linux (voir page 40). Que ce soit Bash, Zsh, Bush, Cshell, KornShell, Dash..., ils sont tous capables de lancer le traitement par le noyau d'un fichier binaire exécutable, et ont accès facilement à toutes les applications installées sur un système GNU/Linux. Or, il est souvent plus simple et plus rapide de taper le nom d'une application dans une console, que d'aller farfouiller dans de jolis menus pleins d'icônes et pleins de sous-menus, eux-mêmes ramifiés...

Mais ce n'est pas tout, un shell sous GNU/Linux peut aussi contrôler la priorité d'exécution d'une application qu'il lance, stopper ou tuer un processus en cours d'exécution et, même mieux, dupliquer un processus ! (Ils ne peuvent pas encore le faire avec du pain, mais les hackers les plus renommés sont sur le coup...).

### Des commandes en mode texte

Il subsiste encore de cet héritage du mode texte toute une panoplie de petites applications qui répondent à un besoin précis d'information en quelques lignes de texte à l'écran. C'est le cas de `pwd`, `ls`, `find`, `ps`, `df`... Ces commandes affichent le répertoire courant, listent le contenu de ce dernier, affichent l'adresse des fichiers d'un nom donné, affichent le nom des processus en cours d'exécution ou l'espace disque disponible.

Et ce n'est pas tout, puisque les applications qui répondent du texte étaient monnaie courante, un trousseau d'applications servant à traiter ces informations en mode texte s'est vite créé autour. On trouve ainsi des `grep`, des `sed`, des `awk`, etc., qui permettent respectivement de sélectionner, trier et transformer des lignes dans un texte. Et les shells y ont aussi mis du leur, en permettant par exemple de rediriger la sortie d'une application vers l'entrée d'une autre, évitant ainsi d'avoir à écrire soi-même la réponse d'un programme en entrée d'un autre, ou permettant de générer, puis faire traiter une même information par plusieurs applications avant d'afficher le résultat sur la console.

Par exemple, la commande : `cat /usr/share/dict/french | grep .asse$` permet d'afficher le contenu du fichier `/usr/share/dict/french`, mais le redirige grâce à la commande `|` du Bash et à la commande `grep`, qui en extrait les mots se terminant par « asse ». Cela peut se révéler utile pour faire de la poésie par exemple...

*Remarque : La chaîne de caractère « .asse\$ » est une expression régulière, un langage de description de motif de caractères dans un texte qui mériterait un hors-série de GNU/Linux Pratique à lui seul...*

Enfin, les shells proposent tous un langage de programmation interne permettant en bidouillant 5 minutes de déplacer des montagnes.

### Dualité d'interface

Il est donc possible de faire une quantité phénoménale de choses avec une console sous GNU/Linux, et concernant l'administration d'une machine, il est à la fois nécessaire et suffisant de disposer d'une console. Toutefois, il est vrai aussi que dans les distributions GNU/Linux modernes, des utilitaires graphiques permettent de configurer une machine en tous points ou presque. Et cette dualité d'interface (texte et graphique) s'est étendue, dans les deux sens, à presque toutes les applications.

Ainsi, des applications qui n'avaient été développées jusque-là qu'avec des interfaces graphiques ont été repensées afin de pouvoir être manipulées dans une console texte, comme le moteur de blog Nanoblogger par exemple. Après, de là à « repenser » Tux Racer, BZFlag ou Frozen-Bubble, il faudrait sacrément aimer son Xterm !

Simon Descarpentries

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>  
<http://www.unixgarden.com>  
<http://www.editions-diamond.fr>

## Comment créer un script de commandes ?

Lorsque l'on utilise régulièrement la même suite de commandes, on se dit assez rapidement que ce serait tout de même plus simple de les regrouper ensemble sous une même commande et de les lancer simultanément une fois pour toutes : l'idée d'un script shell est née.

Un script, c'est un petit programme qu'on écrit dans un langage simple. Sous Linux, le plus basique est de se servir du langage de programmation par défaut installé sur toute machine : le shell (voir question/réponse correspondante). C'est un langage assez simple, mais qui permet tout de même de programmer des tâches assez compliquées (si l'on maîtrise suffisamment ses composantes...). Ici, on ne cherchera pas à apprendre les subtilités de la syntaxe du shell, mais simplement à regrouper une suite de commandes.

D'abord, il faut savoir qu'un script shell n'est ni plus ni moins qu'un fichier texte standard, dans lequel on doit indiquer en premier lieu le nom du langage de programmation à utiliser. Admettons que votre système GNU/Linux fonctionne avec `bash`, comme c'est le cas la plupart du temps. Pour le notifier dans le script, on va utiliser le *sha bang* (ou *she bang*), représenté par `#!` ; ces deux caractères permettent d'introduire le chemin vers le programme qui va interpréter les commandes du script (ici `bash`). La première ligne de `mon_script.sh` (donnez-lui cette extension pour vous rappeler immédiatement qu'il s'agit d'un script shell, mais ça n'est pas obligatoire) doit donc être :

```
#!/bin/bash
```

La deuxième étape est de rendre ce fichier texte exécutable. Pour cela, il suffit de modifier les droits d'accès sur le fichier grâce à la commande `chmod` :

```
~$ chmod 755 mon_script.sh
```

A partir de ce moment, dès que vous lancerez votre script, il pourra être exécuté. Attention, si vous saisissez uniquement le nom du script, voici ce qui va se passer :

```
~$ mon_script.sh
```

```
bash: mon_script.sh: command not found
```

```
~$
```

En effet, par défaut l'exécution de scripts dans le répertoire courant est désactivée. Il est nécessaire de spécifier le chemin d'accès au script, chemin absolu ou relatif ; si le script se trouve dans le répertoire courant :

```
~$ ./mon_script.sh
```

Voici un petit exemple de script en guise de démonstration. Comme vous êtes prudent, vous faites des sauvegardes régulières de vos données : vous créez un répertoire `BACKUP` sur votre bureau, vous y copiez/collez les données à sauvegarder, vous compressez le tout, vous renommez le fichier pour lui ajouter la date du jour, vous supprimez le répertoire créé et ensuite vous utilisez votre programme de gravure préféré pour mettre votre archive à l'abri. L'essentiel de ces tâches peut être automatisé par un script comme le suivant :

```
#!/bin/bash
mkdir BACKUP
cp -r Desktop/Mon-boulot/BACKUP/
tar -czf backup`date +%Y%m%d`.tar.gz BACKUP
rm -r BACKUP
```

Avant de terminer sur ce sujet, un petit conseil : il est possible de mettre dans votre script, en plus des commandes à exécuter, des commentaires, c'est-à-dire du texte qui n'est destiné qu'à la lecture par l'utilisateur. Pour différencier les lignes de commande des commentaires, on fait précéder ces derniers du signe `#`. Cela est assez utile pour se rappeler le rôle du script ou la spécificité de certaines parties du programme.

```
#!/bin/bash
# Ceci est un script qui me permet
# de sauvegarder mon boulot
# à une date précise
mkdir BACKUP
cp -r Desktop/Mon-boulot/ BACKUP
tar -czf backup`date +%Y%m%d`.tar.gz BACKUP
rm -r BACKUP
```

Dans notre exemple très simple, cela peut paraître inutile : le nom du script, `backup.sh`, est assez clair et les commandes qu'il contient sont basiques et lisibles. Pourtant, imaginez que par la suite vous créez d'autres scripts de backup pour sauvegarder d'autres données... Vous allez vite être perdu pour savoir quel script fait quoi ! Cela peut donc être assez utile et devient indispensable lorsque vous vous lancez dans la création de programmes plus complexes...



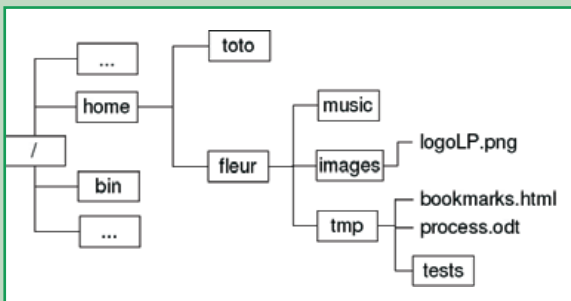
Lucie Robin

Exemplaire offert par Les Éditions Diamond  
<http://www.ed-diamond.com>  
<http://www.unixgarden.com>  
<http://www.editions-diamond.fr>

## Quelles sont les commandes Unix de base ?

Avant d'énumérer les principales commandes à connaître, rappelons que, bien souvent, un fichier ne peut pas être identifié par son nom seul. En effet, il peut arriver que deux fichiers aient le même nom dans un système de fichiers, mais se situent dans des répertoires différents. Ainsi, il est en général nécessaire de spécifier le chemin, sous-entendu le chemin à effectuer dans le système de fichiers, pour atteindre ce fichier.

On en profitera pour noter par ailleurs que, sous Linux, le caractère qui relie chaque répertoire d'un chemin est le slash /, alors que, sous Windows, il s'agit d'un antislash \.



On distingue deux types de chemins : absolu et relatif. Un **chemin absolu** se base sur la racine de l'arborescence. Il commence donc par le symbole /. Si l'on se réfère au schéma ci-contre, quel que soit l'endroit où l'on se trouve dans l'arborescence, on peut faire référence au fichier `bookmarks.html` en entrant le chemin `/home/fleur/tmp/bookmarks.html`

Un **chemin relatif**, lui, dépend du répertoire courant où se trouve l'utilisateur au moment où il fait référence au fichier. Par exemple, toujours d'après le schéma ci-contre, si je me trouve dans mon répertoire `tmp/` et que je souhaite faire référence au fichier `logoLP.png`, j'écrirai le chemin relatif `../images/logoLP.png` (les deux points permettent de « remonter d'un cran » dans l'arborescence).

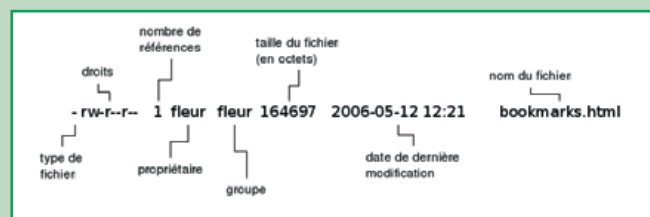
Par ailleurs, sachez que le symbole `~` peut remplacer le chemin absolu vers votre répertoire personnel, autrement dit `/home/login/`.

Par défaut, une console s'ouvre sur le répertoire de base d'un utilisateur. Notez que votre terminal de commandes est doté d'une fonctionnalité d'auto-complétion (activée par défaut dans la plupart des cas) : saisissez les premiers caractères d'une commande ou d'un nom de fichier, puis appuyez sur la touche [Tab]. Le nom du fichier ou la commande est automatiquement complété ! Si vous obtenez un « bip » pour tout résultat, c'est qu'aucun fichier ni aucune commande commençant par les caractères saisis n'existe sur votre système (une autre raison serait que vous tentez d'utiliser une commande réservée à l'administrateur système et qu'il vous faut être connecté en tant que `root` pour l'utiliser). Non seulement cela permet de gagner du temps, mais cela permet d'éviter les fautes de frappe !

La plupart des commandes Unix de base disposent de nombreuses options ; pour en obtenir la liste complète, tapez `man` suivi du nom de la commande ou bien tapez le nom de la commande suivi de `--help`. Il est naturellement possible d'utiliser plusieurs options à la fois, en les juxtaposant derrière la commande principale.

- > Pour connaître à tout moment le chemin absolu du répertoire où l'on se trouve : `pwd` (*print working directory*).
- > Pour afficher le contenu d'un répertoire : `ls` (selon la façon dont votre terminal de commandes est configuré, les répertoires, les images et les fichiers apparaissent chacun dans une couleur distincte pour plus de lisibilité).
- > Pour afficher le contenu détaillé d'un répertoire : `ls -l`

Le tout premier caractère spécifie le type de fichier : `-` = fichier, `d` = répertoire, `l` = lien symbolique, etc. La série de caractères `rwX` indique respectivement les droits de lecture (*read*), d'écriture (*write*) et d'exécution (*execute*) du fichier ou répertoire, pour 1) le propriétaire du fichier, 2) le groupe auquel appartient le propriétaire et 3) tous les autres utilisateurs.



- > Pour afficher les fichiers et répertoires cachés (dont le nom commence par un point), on utilisera l'option `a` : `ls -a`
- > Pour se déplacer dans un autre répertoire (*change directory*) : `cd <répertoire_de_destination>` ; sachez que si vous saisissez `cd` sans aucun argument, vous vous retrouverez dans votre répertoire personnel.

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>  
<http://www.unixgarden.com>  
<http://www.editions-diamond.fr>

- > Pour copier (*copy*) un fichier :  
`cp <fichier_à_copier> <répertoire_de_destination>`
- > Pour copier un répertoire (de façon récursive) :  
`cp -R <répertoire_à_copier> <répertoire_de_destination>`
- > Pour supprimer (*remove*) un fichier :  
`rm <fichier_à_supprimer>`
- > Pour supprimer un répertoire, sans demander confirmation de l'utilisateur :  
`rm -rf <répertoire_à_supprimer>`
- > Pour déplacer un fichier ou un répertoire (*move*) :  
`mv <fichier/répertoire_à_déplacer> <répertoire de destination>`
- > Pour créer un répertoire (*make directory*) :  
`mkdir <nom_donné_au_nouveau_répertoire>`
- > Pour créer une série de répertoires imbriqués :  
`mkdir -p <série/de/répertoires/>`
- > Pour visualiser le contenu d'un fichier :  
`cat <nom_du_fichier>` (attention, `cat` permet uniquement d'afficher le contenu d'un fichier, et non de l'éditer pour le modifier).
- > Pour visualiser le contenu d'un fichier page par page :  
`more <nom_du_fichier>` (appuyez ensuite sur la barre d'espace pour passer de page en page).
- > Pour rechercher la présence d'un motif dans un fichier :  
`grep <motif_à_rechercher> <fichier_à_tester>` (notons que `grep` peut également être utilisé pour filtrer le résultat d'une commande, à l'aide du *pipe*).
- > Pour enchaîner plusieurs commandes :  
`|` (*pipe*).

Ex. : `ps -ef | grep openoffice`. Dans cet exemple, nous listons les processus actifs, dont les lignes de description contiennent le motif « openoffice ».

----> Pour rediriger la sortie d'une commande : `>`.

La sortie standard d'une commande est en effet l'affichage à l'écran. Mais il est possible d'effectuer une redirection vers un fichier texte, ou encore directement vers votre imprimante.

Ex. : `cat fichier.txt > /dev/lp0 (/dev/lp0 correspond ici au port auquel est connectée notre imprimante).`

Comme vous pouvez le constater, ces commandes sont vraiment très simples à utiliser et permettent d'effectuer des manipulations de base beaucoup plus rapidement que par l'intermédiaire d'interfaces graphiques ! Avec un peu d'entraînement, vous ne pourrez plus vous en passer...

Fleur Brosseau

## A quoi ça sert un cronjob ?

En bref : à exécuter des tâches périodiques de manière automatisée. En long : `cron` est un démon qui tourne sous la plupart des distributions Linux et dont la fonction est de gérer certaines tâches périodiques. En cela, un *cronjob* est une tâche périodique définie par l'utilisateur, qui sera exécutée par le système (ou qui a été prédéfinie par les auteurs de la distribution Linux que vous utilisez).

Considérons un exemple simple : nous allons faire en sorte de nous envoyer le même e-mail tous les matins...

```
Moi@machine:~$ echo "La vie est belle !" | mail -s "Mon mail de réconfort " Moi
```

Bon, à ce stade, nous n'en sommes qu'à la rédaction du mail et rien n'est encore automatisé. Pour pouvoir automatiser une tâche, il faut l'indiquer à `cron`, dans des fichiers spécifiques appelés « crontabs » ; chaque utilisateur dispose d'un crontab qu'il gère à sa guise. Pour accéder à votre crontab, tapez :

```
~$ crontab -e
```

Une ligne de commande dans ce fichier est constituée de 6 champs, séparés par des espaces ou des tabulations, dans cet ordre :

minutes	heures	jour-du-mois	mois	jour-de-la-semaine	commande
(0-59)	(0-23)	(1-31)	(1-12)	(0-6, 0=dimanche, 1=lundi, etc.)	la commande à exécuter

La syntaxe des champs peut apparaître différemment selon votre distribution ; ainsi, on trouvera `jj` ou `dom` (*day of month*) pour le jour du mois, `mon` pour *month*, etc. Plusieurs éléments d'un même champ sont séparés par une virgule (ex. : `1,3,5` dans le champ *mois* signifie « janvier, mars, mai ») ; de même, un intervalle s'exprime par un tiret (ex. : `1-3` pour le jour de la semaine signifie « du lundi au mercredi ») ; une étoile (\*) désigne le plus grand intervalle possible.

Dans notre exemple, si on veut s'envoyer un mail tous les jours le matin à 8 heures 01, on tapera :

```
01 8 * * * echo "La vie est belle !" | mail -s "Mon mail de réconfort " Moi
```

Si on veut exécuter un script tous les premiers du mois à 5h42 du matin :

```
42 5 1 * * monscript.sh
```

Ou faire une sauvegarde du lundi au vendredi, tous les jours à 11h59 :

```
59 11 * * 1-5 backup.sh
```

Le tout premier rôle de `cron` est en fait de gérer les tâches d'administration du système. Dans la mesure où celles-ci sont généralement répétitives, c'est assez utile d'avoir un tel système de gestion des tâches à sa disposition... Le fichier de configuration principal de `cron` est généralement `/etc/crontab`. Il y a une petite différence avec la syntaxe des crontabs des utilisateurs ; puisque ce fichier de commandes est exécuté en *root*, la possibilité existe d'indiquer un autre utilisateur que *root* pour l'exécution des tâches programmées.



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>



La syntaxe des commandes est donc la suivante :

```
minutes heures jour-du-mois mois jour-de-la-semaine utilisateur commande
```

Par exemple :

```
7 20 * * * root echo "Cette commande est exécutée tous les jours en root à 20h07"
```

```
7 20 * * * bob echo "Cette commande est exécutée tous les jours par l'utilisateur bob à 20h07"
```

Le reste de la syntaxe devrait vous être familier maintenant.

## Contrôler l'usage de cron

L'utilisation de `cron` peut entraîner la mobilisation des ressources du système, particulièrement si les tâches administratives sont lourdes et nombreuses (ex. : 50 utilisateurs ayant paramétré un cronjob qui s'exécute toutes les minutes peut occuper pas mal de ressources système...). Pour éviter ça, `cron` intègre une gestion des autorisations d'usage. Deux fichiers sont utilisés : `/etc/cron.deny` et `/etc/cron.allow`.

La syntaxe est très simple et identique aux autres démons qui fonctionnent avec des `.allow` et `.deny` : pour interdire à un utilisateur particulier l'usage de `cron`, il suffit d'inscrire son nom dans le fichier `cron.deny`. Par exemple :

```
~$ echo "bob" >> /etc/cron.deny
```

va interdire à bob l'usage de `cron`, tout simplement. Pour n'autoriser que certains utilisateurs nommément désignés, on saisira la commande :

```
~$ echo "ALL" >> /etc/cron.deny && echo "bob" >> /etc/cron.allow
```

... et seul bob sera autorisé à utiliser `cron`.

**Remarque :** si aucun des deux fichiers n'existe, seul le super utilisateur (root) aura le droit d'utiliser `cron`. Par ailleurs, un fichier `/etc/cron.deny` vide signifie que tous les utilisateurs peuvent utiliser `cron`.

Le processus `crond` est normalement lancé au démarrage du système. Pour le vérifier :

```
~$ ps aux | grep cron
```

```
root      4797  0.0  0.2  2120  696 ?        Ss   09:00   0:00 /usr/sbin/cron
```

C'est bon, apparemment ça tourne ! Vous pouvez également le visualiser sur le gestionnaire graphique de services de votre environnement de bureau. Si le service n'est pas démarré, cochez la case correspondante (via l'interface graphique) ou bien saisissez la commande :

```
/etc/rc.d/init.d/crond start ou /etc/init.d/crond start ou encore /etc/init.d/cron start (selon la distribution utilisée...).
```

Lucie Robin

## Comment mettre à jour ses paquets de logiciels ? Est-il toujours facile d'installer un programme récent sous Linux ?



Les principales distributions Linux intègrent, dans leur gestionnaire de paquets, un mécanisme de mise à jour. Il suffit à chaque fois de spécifier un serveur source, puis de comparer régulièrement les versions sur le serveur avec les versions installées, pour télécharger et installer la nouvelle mouture. En général, un utilitaire dédié, tel *MandrivaUpdate* sous Mandriva, exécute toutes ces opérations automatiquement et peut même s'exécuter tout seul en tâche de fond.

La question devient un piège dès lors que l'on veut installer un nouveau logiciel ! La réponse est « oui, c'est très facile » quand vous choisissez un logiciel disposant de paquets `.rpm` ou `.deb`, bien préparés et bien adaptés à votre distribution. Que cette préparation ait été faite par l'éditeur officiel de la distribution, par l'auteur du logiciel lui-même ou par une tierce personne doit quand même attirer votre attention sur la propreté de l'emballage. Le CD-rom de votre distribution, ou son site FTP officiel, regorgent de programmes tout prêts, à installer en quelques clics à l'aide de programmes comme *Yum*, *Synaptic* ou *Rpmdrake* : usez et abusez-en !

La tâche devient plus délicate si vous voulez installer des versions toutes récentes d'un logiciel. Et la réponse sera ici beaucoup plus nuancée. Car c'est là qu'on s'aperçoit de l'obsolescence extrêmement rapide d'une distribution Linux : impossible de trouver des paquets `.rpm` pour installer le dernier Koffice ou le dernier Epiphany sur une Mandriva LE2005, par exemple ! En effet, les développeurs utilisent presque toujours les dernières technologies logicielles à leur disposition, lesquelles ne sont jamais celles d'un système ayant passé les 10 mois d'ancienneté ! Un inconvénient qu'on rencontre beaucoup plus rarement sous Microsoft Windows, où la majorité des programmes fonctionnent aussi bien sous Windows 98 que sous Windows XP.

Sous Linux, si vous voulez sortir des sentiers balisés par l'éditeur de votre distribution et essayer d'autres programmes que ceux prévus à l'instant t de l'installation, vous n'aurez bien souvent que deux possibilités : soit compiler vous-même, soit mettre à jour entièrement votre distribution. Et encore, même la compilation trouvera vite ses limites. Ce « portage arrière » (ou *backport*) nécessite souvent de remettre à jour, éventuellement en compilant, des portions très importantes de votre système, avec des dépendances en cascade très difficiles à gérer sans tout casser : tel module demande telle nouvelle bibliothèque, qui demande telle nouvelle version d'un langage, qui obligera à installer un nouveau bureau et modifiera des modules qui voudront une nouvelle version du noyau ! Vous l'aurez compris, installer un programme trop récent revient bien souvent à passer, tout simplement, à la version supérieure de votre distribution.

Jean-Noël Pierron

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Est-ce bien ou mal la compilation « à la hache » ? Puis-je créer un paquet pour ma distribution ?



Cependant, parfois, on ne dispose pas du tation est grande, dans ce cas, de suivre les c'est le meilleur moyen de rendre votre système impossible à administrer... Une compilation ainsi faite ne sera pas comptabilisée dans la base de données du système et, donc, le risque de conflits et d'instabilité est énorme. D'autant plus que les compilations « à la hache » ne respecteront sûrement pas les règles de la distribution concernant les fichiers de configuration. Il est donc fort probable que vous vous retrouviez avec des fichiers dans des endroits imprévus et, peut-être même, que le système ne reconnaisse pas tout ou partie de votre compilation. C'est donc plutôt une mauvaise idée, mais aussi un excellent moyen d'apprendre par l'erreur ! Compiler est une excellente façon d'apprendre comment fonctionne un système, quel qu'il soit (cela dit, il n'est pas nécessaire de brûler la maison pour savoir utiliser des allumettes...).

Si vous voulez disposer d'un logiciel qui n'existe pas pour votre distribution, il est souvent relativement facile (plus facile qu'une compilation) de faire un paquet dudit logiciel pour votre distribution. Tout d'abord, je vous invite à bien regarder si personne sur la toile ne l'a déjà fait et mis en ligne. Une fois cette première recherche effectuée, je vous invite aussi à contacter, s'il existe, le gestionnaire du paquet antérieur (par exemple pour des questions de version) afin de savoir s'il va se lancer sur la nouvelle version et d'ici combien de temps. Le cas échéant, contactez aussi les responsables de la distribution si vous voulez vous attaquer à un logiciel majeur du monde libre ou à une application métier dont les ramifications risquent de toucher beaucoup d'autres paquets.

Maintenant, il est temps de vous lancer dans un processus de compilation et de fabrication de paquets. Pour ce faire, je vous invite à suivre ce petit manuel du « packageur » débutant. C'est un court résumé qui ne remplacera pas la documentation de votre distribution bien sûr, mais il peut vous aider à ne pas faire certaines erreurs :

→ Ne compilez jamais sur le système qui va se servir du paquet, sauf impérieuse nécessité. La compilation d'un logiciel exige d'installer des outils et des bibliothèques qui n'ont normalement rien à faire sur un système quotidien. En effet, à moins que vous ne soyez développeur pour Gnome, KDE ou autre, vous n'avez normalement pas besoin des outils de développement et de compilation. Or, ces outils sont aussi une source de faille potentielle en termes de sécurité. Si vous le pouvez, je vous invite donc à utiliser une autre machine pour faire ce travail.

→ Lisez attentivement la documentation fournie par les auteurs de votre distribution, c'est essentiel. Toutes les distributions disposent d'informations concernant la fabrication des paquets, les meilleures méthodes et les moyens simples pour le faire. En plus, souvent, ces documentations sont en français, au moins en partie. À tout hasard, vérifiez également que vous avez le droit de créer ce paquet pour votre distribution et que rien n'entrave la libre diffusion du logiciel sur lequel vous travaillez.

→ Faites-vous la main avec un ou deux petits paquets existant pour votre distribution. Vous disposez automatiquement des sources, des informations pour les mettre en paquets et de la méthode. En gros, avant de vous lancer dans le travail de fond, des exercices corrigés, cela ne fait pas de mal !

→ Testez l'installation de votre paquet sur un autre système que celui de production et celui de compilation. Je sais, cela commence à faire beaucoup d'ordinateurs, mais, croyez-moi, c'est plus prudent. Car un paquet, c'est un objet qui va interagir avec le système de façon plus ou moins forte. Donc, s'il y a un souci, il vaut mieux que vous le testiez dans un environnement sécurisé avant de devoir corriger votre station de travail ou le dernier serveur en production !

Enfin, une fois le paquet fait et fonctionnel, faites signe à l'auteur du logiciel ainsi qu'aux personnes qui gèrent la distribution et mettez-le en ligne. Vous rendez ainsi service à l'auteur en diffusant son œuvre et à la distribution en lui offrant un logiciel essentiel. Cependant, souvenez-vous de deux choses :

→ Restez humble ! C'est bien ce que vous avez fait, mais ça ne fait pas de vous un génie. Alors, pas besoin d'annoncer à toute la planète que vous allez révolutionner le Logiciel libre. Ou bien attendez-vous à quelques sarcasmes...

→ Ce n'est pas parce que cela fonctionne chez vous que cela fonctionnera partout ! Il est possible, voire probable, que l'on vous contacte en vous disant que votre paquet ne marche pas dans tel ou tel cas. Si vous vous en sentez la force et les compétences, bienvenue parmi les packageurs de votre distribution. Sinon, indiquez bien que vous ne pouvez/savez/voulez pas faire le suivi de votre œuvre, mais, dans ce cas, mettez quelque part le fruit de votre travail et les explications en ligne afin que d'autres prennent votre suite.

La plupart des systèmes d'exploitation libres ont été dotés relativement tôt de systèmes de gestion de paquets. C'est-à-dire que l'installation d'un logiciel est enregistré dans une base de données afin de faciliter la gestion des dépendances (ce dont a besoin le logiciel pour fonctionner), la gestion des conflits (si le logiciel ne peut pas ou ne doit pas fonctionner en parallèle à un autre pour garantir la stabilité de tout l'ensemble) et la désinstallation éventuelle. Il existe des systèmes d'exploitation qui ne sont pas dotés de tels systèmes par défaut, le Mac OS X par exemple, et cela peut parfois être pénible de gérer l'installation et la désinstallation de logiciels sur de telles plateformes.

logiciel que l'on veut. Ou pas dans la version qu'il faut. Et la ten- indications du concepteur qui sont : compiler et utiliser ! Sauf que

## Comment créer/désactiver un compte utilisateur ?

Lorsque vous installez votre distribution, il vous est toujours demandé de renseigner un nom d'utilisateur (ce sera le premier utilisateur du système) ainsi qu'un mot de passe pour cet utilisateur. Parfois, vous devez faire de même pour l'utilisateur *root* (le super-utilisateur du système qui aura tous les droits d'administration de votre machine).

Pour créer ou désactiver un compte utilisateur, comme pour la plupart des opérations que vous pouvez effectuer sur votre système, deux solutions s'offrent à vous : utiliser un utilitaire graphique ou utiliser la ligne de commande. Toutes deux sont très faciles.

L'utilitaire graphique se situe en général dans la section **Administration** ou **Système** ou encore **Configuration** de votre environnement de bureau. Il s'intitule à peu de choses près « Gestion des groupes et utilisateurs ». Cet outil dispose en général d'un bouton **Ajouter**, qui vous permet de créer un nouvel utilisateur ; vous devrez alors renseigner son nom complet et son mot de passe, ainsi que diverses informations facultatives. Vous aurez également la possibilité de lui attribuer certains privilèges concernant la gestion du système, à choisir dans une liste d'options prédéfinies. Pour supprimer un utilisateur, il vous suffira de sélectionner l'un des utilisateurs du système dans la liste, puis de cliquer sur le bouton **Supprimer**.

Pour gérer les utilisateurs du système en ligne de commande, trois commandes essentielles sont à retenir : `useradd`, `passwd` et `userdel`.

Pour créer un nouvel utilisateur :

```
~# useradd <nom_utilisateur>
```

Notez que l'ajout d'un utilisateur entraîne la création de son répertoire personnel (par défaut, dans le répertoire `/home` où se trouve déjà votre propre compte). Puis, sans attendre, il faut créer un mot de passe pour cet utilisateur :

```
~# passwd <nom_utilisateur>
```

```
Enter new UNIX password :
```

```
Retype new UNIX password :
```

```
passwd : le mot de passe a été mis à jour avec succès
```

J'attire votre attention sur le fait qu'il vous faut les droits de super-utilisateur pour utiliser ces commandes. En outre, il existe des options pour chacune de ces commandes, par exemple :

```
~# useradd <nom_utilisateur> -e <date_expiration> : permet de fixer une date à laquelle le compte utilisateur sera désactivé (la date est spécifiée au format AAAA-MM-JJ).
```

```
~# passwd -l <nom_utilisateur> : verrouille le compte de l'utilisateur (-u pour déverrouiller le compte)
```

La commande `useradd` utilisée sans options utilise les attributs prédéfinis par défaut ; pour les connaître, lancez la commande `useradd -D` :

```
~$ useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/sh
SKEL=/etc/skel
CREATE_MAIL_SPOOL=no
```

Il suffit donc d'utiliser l'option adéquate si vous souhaitez changer l'un de ces paramètres pour l'utilisateur que vous créez (ex. : `useradd <nom_utilisateur> -d <répertoire_de_connexion>`, si vous souhaitez que le répertoire personnel de l'utilisateur soit créé ailleurs que dans `/home`).

Pour supprimer un compte utilisateur :

```
~# userdel <nom_utilisateur>
```

Pour supprimer, par la même occasion, le répertoire de connexion de l'utilisateur, ajoutez l'option `-r`.





Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Maintenant que je suis sous Linux, comment est-ce que je fais pour lire mes fichiers MS Excel, MS Word, etc. ?



Certes, la suite bureautique Microsoft Office est une suite propriétaire, mais cela ne signifie en aucune manière que vos fichiers seront illisibles sur un système GNU/Linux ! C'est grâce aux possibilités des outils bureautiques libres que vous pourrez lire et éditer vos documents.

Ainsi, les célèbres suite **OpenOffice.org** (<http://fr.openoffice.org/>) et **KOffice** (<http://www.koffice.org/>) (cette dernière étant destinée à l'environnement de bureau KDE) peuvent tout à fait ouvrir un fichier MS Office. Vos fichiers MS Word (fichiers **.doc**) pourront être utilisés dans OpenOffice Writer ou dans KWord, MS Excel (fichiers **.xls**) dans OpenOffice Spreadsheet ou Kspread et MS PowerPoint (fichiers **.ppt**) dans OpenOffice Impress, etc.

Vous pourrez non seulement lire ces fichiers (via le menu **Fichier -> Ouvrir** de l'application concernée), mais également les modifier et les réenregistrer dans leur format d'origine. Cela dit, cette dernière étape pourra entraîner parfois la perte du formatage et des styles utilisés dans votre document, qui, eux, sont propres au logiciel propriétaire qui les a conçus. Le Logiciel libre utilisé vous en avertira avant l'enregistrement.

**Abiword** (<http://www.abisource.com/>) est également un excellent traitement de texte, évidemment plus léger que les suites complètes mentionnées ci-dessus. Il peut ouvrir sans difficulté un document **.doc** et enregistrer un document dans ce même format.

Quant à vos bases de données MS Access (format **.mdb**), inutile de faire une croix dessus ! Sous Linux, il existe deux applications phares pour la création et la gestion de base de données : il s'agit du module **Base** de la suite OpenOffice et de **Kexi** (environnement KDE). Même si des progrès sont encore à faire (surtout concernant OooBase) toutes deux proposent quasiment les mêmes fonctionnalités que MS Access : création de tables, de contenu, de requêtes, de formulaires, de rapports, etc. En outre, il existe un ensemble d'outils pour GNU/Linux, nommé **mdbtools** (<http://mdbtools.sourceforge.net/>), qui permet de manipuler sous GNU/Linux des bases de données créées avec MS Access.

Fleur Brosseau

## Existe-t-il pour Linux des clients de messagerie aussi performants que Microsoft Outlook ?



En termes d'application uniquement dédiée au courrier électronique, quatre applications graphiques sont couramment utilisées sous GNU/Linux :

----> Tout d'abord **Thunderbird** (<http://www.mozilla-europe.org/fr/products/thunderbird/>), qui, avec le navigateur Web Firefox, fait partie des 2 produits phares de la fondation Mozilla. Thunderbird est d'ailleurs disponible pour Windows et Mac OS X. Ses utilisateurs apprécient en particulier les nombreuses extensions disponibles qui permettent d'augmenter les fonctionnalités de Thunderbird (à la manière de Firefox) et ses puissantes fonctionnalités anti-spam.

----> Les utilisateurs de l'environnement Gnome en particulier pourront utiliser le client **Evolution** (<http://www.gnome.org/projects/evolution/>). Ce dernier inclut en outre un carnet d'adresses, un calendrier/agenda, un gestionnaire de tâches et d'événements ainsi qu'un bloc-notes.

----> Concernant l'environnement KDE, **Kmail** (<http://kmail.kde.org/>) est lui aussi un client de messagerie tout à fait satisfaisant, de la même trempe qu'Evolution : carnet d'adresses, chiffrement via OpenPGP, support des protocoles POP3, IMAP et SMTP, gestion du filtrage des messages, moteur de recherche intégré, messages au format texte ou HTML, éditeur avancé, etc.

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

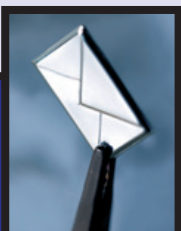
----> **Sylpheed-Claws** (<http://claws.sylpheed.org/>) est un peu moins utilisé, sans doute à cause du moindre tapage médiatique qui en est fait. Pourtant, il n'en est pas moins performant : comptes multiples, carnet d'adresses, support GnuPG, gestion des filtres, sauvegarde automatique des messages, auto-complétion des adresses destinataires, configuration avancée, etc.

Pour ceux qui se sentent à l'aise dans le maniement de la ligne de commande, il existe un client de courriel en mode texte totalement paramétrable nommé **Mutt** (<http://www.mutt.org/>). Néanmoins, son usage est réellement réservé à ceux qui ne souhaitent pas « s'encombrer » d'une interface graphique et qui ne recherchent pas particulièrement l'ergonomie et la simplicité d'utilisation d'une interface graphique en général.

### Pourrai-je ouvrir des pièces jointes d'un format propriétaire (.doc, .ppt, etc.) ?

En général, aucun problème pour cela. Mais si jamais votre client de courriel refuse de l'ouvrir directement, il vous suffira d'enregistrer la pièce jointe sur votre disque, puis d'utiliser l'application adéquate en fonction du type de document.

F. B.



## Comment avoir accès à des données aussi bien avec GNU/Linux que MS-Windows quand on a les 2 sur la même machine ?

### Les clés USB

Comme vous avez sûrement déjà pu le constater, quel que soit le système d'exploitation de l'ordinateur sur lequel on branche une clé USB, on a toujours accès aux informations qu'il y a dessus, alors que lorsqu'on s'intéresse aux partitions d'un disque dur, les choses se compliquent. Le système propriétaire refuse de voir les partitions formatées selon des méthodes libres (telles qu'ext2 ou 3, reiserfs, etc.), et de par le secret de ses sources, il rend l'écriture impossible sur son système de fichier phare NTFS (imposé à l'installation de MS-Windows depuis Windows NT 3.1 en 1993). C'est tout juste si les méthodes de rétro-conceptions permirent aux développeurs du noyau Linux de pouvoir lire les partitions formatées en NTFS, et, plus récemment, de modifier un fichier déjà existant.

Mais alors, quel est le secret des clés USB ?

Toute la magie des clés USB réside dans l'utilisation d'un système de fichier archaïque et propriétaire : j'ai nommé FAT32 (et parfois même FAT16). Ce système de fichier est la propriété de Microsoft et il date de MS-DOS, soit plus de 20 ans. Étant particulièrement simple, il fut facilement rétro-conçu, et donc dispose de pilotes libres fiables. Devant sa quasi-universalité, on comprend que les constructeurs de clés USB en tout genre jettent leur dévolu dessus, mais ils s'en tirent pour l'instant avec une épée de Damoclès au-dessus de leurs têtes, car Microsoft a déjà menacé à plusieurs reprises de réclamer des redevances sur l'utilisation de son format...

### Les partitions de disque dur

Une première solution pour partager des données entre GNU/Linux et MS-Windows est donc de créer une partition au format FAT32 sur son disque. Cette dernière sera visible à partir du système d'exploitation propriétaire, et accessible en lecture et écriture à partir de GNU/Linux. Mais cette solution n'est pas la panacée. FAT32 est lent et pas forcément très fiable. Il suffit d'essayer de le convertir en autre chose à partir d'un logiciel de gestion de partitions, même propriétaire, pour s'en rendre compte. Mais pire encore, de par l'ancienneté de sa conception, il contient des limitations devenues gênantes de nos jours, comme la taille maximale d'un fichier fixée à 4GO... Quand on sait qu'une image ISO de DVD pèse généralement 4,7GO, on comprend qu'il faille chercher un peu plus loin pour trouver une solution pérenne.

Après avoir dit ça, vous vous attendez sûrement à ce que je propose une meilleure solution... Et puisque vous avez lu jusque-là, je vais le faire ! Pour partager efficacement des données entre GNU/Linux et MS-Windows, il suffit de formater une partition avec ext3, et d'ajouter à MS-Windows les pilotes qui lui manquent pour vous donner accès à cette partition, j'ai nommé : *Ext2 IFS For Windows*, le système de fichier Ext2 installable pour MS-Windows. Un pilote libre de support d'Ext2 pour Windows NT ou supérieurs. Une fois ce pilote ajouté à MS-Windows, les partitions Ext2 (et accessoirement 3 aussi) que vous souhaitez monter dans MS-Windows apparaissent à la suite des disques déjà référencés, avec chacune leur nouvelle lettre, rien de plus simple...

Simon Descarpentries

● Ext2 IFS For Windows :  
<http://www.fs-driver.org/>

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com><http://www.unixgarden.com><http://www.editions-diamond.fr>

# Peut-on faire cohabiter Linux et Windows sur la même machine ?

Il est tout à fait possible d'utiliser plusieurs systèmes d'exploitation sur une même machine. En fait, vous avez même l'embarras du choix sur la façon de procéder.

## Le dual-boot

Par cette méthode, vous installez plusieurs systèmes d'exploitation sur différentes partitions de votre disque dur (ou sur différents disques). Au démarrage de la machine, le *bootloader* vous permet de choisir sur quel système démarrer. Si vous souhaitez changer le système en cours d'exécution, vous devez redémarrer votre machine.

Cette solution, si elle n'est pas la plus confortable, reste la plus utilisée car chacun des systèmes vit dans son coin et fonctionne comme s'il était le seul OS installé. Il est, de plus, facile de partager des données entre les systèmes : vos partitions Windows seront accessibles sous Linux. L'inverse est également possible en installant un logiciel dédié, comme **Explore2fs** ou **Ext2 IFS**.

Il convient en revanche de bien maîtriser l'installation des deux systèmes et le partitionnement des disques pour éviter de supprimer l'un en installant l'autre...

Attention également, si vous voulez faire cohabiter Windows et Linux, pensez à installer Windows en premier, et Linux en second : sinon, l'installation de Windows aura pour effet de supprimer le bootloader. Votre système Linux sera toujours présent, mais vous ne pourrez plus démarrer dessus. Pour résoudre cette situation, il faut passer par un CD Linux bootable (comme le CD d'installation de votre distribution) pour réinstaller le bootloader.

## La virtualisation et l'émulation

La virtualisation permet de faire fonctionner un système « invité » au-dessus d'un système « hôte », et permet aux deux de fonctionner simultanément. Imaginez votre Windows tournant dans une fenêtre sous Linux ou l'inverse ! Cette solution est également plus sûre, car l'invité n'a pas directement accès aux ressources de l'hôte (par exemple, le disque dur de l'invité est un simple fichier sur l'hôte). En revanche, l'invité souffre de performances moindres par rapport à un système installé normalement et ne peut pas tirer parti des fonctionnalités matérielles telles que l'accélération vidéo.

Cette solution est idéale pour utiliser un système de manière sporadique, tester une nouvelle distribution ou faire des expériences sur un système « poubelle » avant de les répéter sur un vrai. Le partage des données et de la connexion internet entre les deux machines est possible au travers d'un réseau virtuel, les faisant apparaître comme deux machines distinctes.

Le logiciel de virtualisation le plus connu est le propriétaire (mais gratuit pour certaines versions) **VMWare**, mais une alternative libre est particulièrement crédible : **QEMU** permet ainsi d'émuler un grand ensemble de machines, et offre des performances très acceptables en virtualisation avec le module noyau **KQEMU**. Windows, Linux et beaucoup d'autres systèmes s'y installent et y fonctionnent sans problème.

Vous voulez essayer un *live-CD* sans risque ? Installez **QEMU**, prenez un CD ou DVD bootable, mettez-le dans votre lecteur et lancez la commande suivante :

```
$ qemu -cdrom /dev/cdrom
```

Elle aura pour effet de simuler une machine sans disque dur, mais avec un lecteur optique et de démarrer sur ce dernier. Vous verrez alors le système présent sur le CD démarrer dans une simple fenêtre !

## Les couches de compatibilité

La dernière manière d'utiliser des applications Windows sous Linux (ou inversement) consiste à leur fournir une implémentation des bibliothèques du système d'origine. Aucun système supplémentaire n'est alors lancé, et les applications apparaissent comme des fenêtres « normales ».



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

**Wine** implémente ainsi l'ensemble (ou presque) des bibliothèques de Windows ainsi qu'un chargeur permettant d'exécuter un binaire Windows sous Linux. Son utilisation est des plus simples et ne nécessite pas d'installer Windows au préalable (et donc de disposer d'une licence). Wine permet de faire fonctionner bon nombre d'applications Windows sous Linux avec des performances comparables (sinon identiques) à l'original. Un nombre considérable de jeux fonctionnent également et avec l'accélération 3D s'il vous plaît ! Les personnes désirant jouer maximiseront leurs chances en utilisant **Cedega**, solution propriétaire développée à partir de Wine par *Transgaming* et dédiée aux jeux.

Réaliser l'inverse, c'est-à-dire lancer des applications Linux sous Windows, est bien plus simple. En effet, la nature ouverte de Linux et de ses applications fait qu'il est relativement aisé de porter un logiciel sous Windows. Aussi, une couche logicielle n'est pas toujours nécessaire : souvent, il suffit d'une simple recompilation. **Cygwin** fournit la majorité des outils de base de Linux (*shells*, compilateurs, X-window, bibliothèques spécifiques à Unix, etc.) pour Windows, et permet d'en porter d'autres d'autant plus facilement. On peut également signaler le projet **coLinux** qui cherche, lui, à porter le noyau Linux sous Windows (carrément !). Avantage ? On dispose alors d'un vrai Linux qui fonctionne au sein d'un Windows, sans aucune solution de virtualisation.

Alexandre Courbot

## Peut-on utiliser un logiciel créé pour Windows sous GNU/Linux ?

S'il s'agit d'installer sous Linux, directement et sans autre forme de procès, un logiciel spécifiquement créé pour Windows, la réponse est non. Les deux systèmes d'exploitation n'ont rien en commun, si ce n'est parfois le matériel sur lequel ils peuvent fonctionner. Donc, je suis désolé, mais il n'est pas possible de prendre simplement une application Windows et de la faire tourner sous Linux et vice-versa. Soit dit en passant, c'est la même chose entre Mac OS X et Windows ou Mac OS X et Linux.

Par contre, en utilisant des programmes spécifiques d'émulation qui eux vont faire le travail de cohabitation ou de traduction entre les différents environnements, il est possible de faire fonctionner l'application Windows désirée et de la faire interagir avec les applications Linux (sous certaines contraintes en fonction de l'émulation choisie). On obtient alors le résultat voulu indirectement.

On va prendre les solutions d'émulation de la plus proche de Linux à la plus éloignée :

- > Wine (<http://www.winehq.com/>) ;
- > Crossover office (payant, <http://www.codeweavers.com/site/products/cxoffice/>) ;
- > Win4lin (payant, <http://www.win4lin.com/>).

Ces solutions ont en commun de ne pas avoir d'impact sur la vélocité du système. Les applications Windows tournent à leur vitesse native. On notera au passage que deux des solutions proposées ne sont pas libres. Et pour cause ! Si interagir avec le monde Windows demande beaucoup de travail et que ce sont souvent des sociétés ou des professionnels qui ont besoin de ce type d'émulation, il faut aussi savoir que certaines informations sont payantes auprès de Microsoft et nécessitent des accords spécifiques avec la société de RedMond.

Cependant, commençons par le commencement et le projet **Wine**. Ce projet vise à redévelopper sous Linux les API 32 de Windows (à partir de Windows 95), et ce, afin de pouvoir faire fonctionner nativement un logiciel Windows sous Linux. Cependant, même si ce projet en une dizaine d'années a fait d'extraordinaires progrès autant en termes techniques que de performances, il reste toujours incomplet et instable. En fonction du logiciel que vous voulez utiliser, regardez les informations que les membres du projet Wine vous donnent à son sujet. Cela vous permettra de savoir si Wine est une solution envisageable ou non. L'application Windows spécifique que l'on lance avec Wine est considérée par Linux comme une application Linux et donc dispose de toutes les ressources du système à disposition de l'utilisateur qui le lance. Par contre, la reconnaissance ou non des périphériques reste liée à la qualité de l'émulation de l'API 32 qui le gère : c'est la loterie...

**Crossover office** est une dérivation de Wine. Ce travail de perfectionnement de Wine permet de garantir le fonctionnement convenable des applications bureautiques phares du monde Windows, Microsoft Office et certaines applications comme Sage Contact ou Lotus Notes.

**Win4lin** est un peu différent. Il se rapproche plus de produits comme VMWare, même si son fonctionnement technique reste une surcouche à Linux et non une machine virtuelle. En fait, Win4lin permet d'installer un système Windows (98, 2000 ou XP) complet sur une partition Linux et de faire fonctionner au sein de ce système Windows toutes les applications usuelles. La seule limite tient actuellement dans la reconnaissance des périphériques USB. Tout ne fonctionne pas, mais ça s'améliore ! L'installation se fait dans un répertoire dédié sur le disque dur choisi et donc toutes les données restent accessibles depuis Linux ou depuis le système Windows lorsqu'il est lancé. Windows dans son ensemble est alors considéré comme une application Linux parmi d'autres. Pour ceux qui ont vraiment besoin d'une application de travail et qui sont prêts à mettre quelques euros (de 50 à 90 € environ), c'est une solution qui mérite qu'on s'y attarde.

Sidoine Pierrel



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

Il est parfois difficile d'expliquer aux utilisateurs les plus zélés de Linux et des Logiciels libres que la grande différence entre le Logiciel libre et le logiciel propriétaire n'est pas dans le prix, mais dans l'usage. Car, sur cette bonne vieille Terre, le nombre de logiciels piratés est légion. Je me passerai des statistiques grandiloquentes des éditeurs de logiciels, mais force est de constater que l'on ne paie pas toujours son logiciel et encore moins quand on est un particulier...

Alors pardonnez-moi la digression, mais, avec le Logiciel libre, vous êtes sûr d'être dans votre bon droit et de pouvoir ultérieurement réutiliser vos données sans risque. Maintenant, vu que vous migrez depuis le monde de Microsoft, quels logiciels font quoi ? Il existe quelques listes et portails sur le Web recensant les équivalences, mais ils sont souvent obsolètes, donc je ne les listerai pas ici. Pourquoi cet état de fait ? Parce que tenir à jour une liste de concordances entre les mondes propriétaire et libre est une véritable sinécure. Les deux mondes bougent en permanence et ce qui était un logiciel génial hier ne le sera peut-être pas demain. Donc, à vous de chercher dans les grands portails (<http://freshmeat.net/> par exemple) et de suivre une règle de bon sens : un bon logiciel, c'est celui qui fait ce dont vous avez besoin quand vous en avez besoin !

On notera par ailleurs que certains logiciels commerciaux ont un petit frère sous Linux (Real Player, Nero Suite, Opera, etc.). À vous de voir comment vous voulez travailler et les avantages et inconvénients des uns et des autres.

Voici quelques équivalences actuelles :

- > **KDE/Gnome** <-> Windows (environnement graphique) ;
- > **KOffice/OpenOffice.org** <-> Microsoft Office ;
- > **Amarok** <-> Windows Media Player ;
- > **K3b** <-> Nero Suite ;
- > **Kdevelop** <-> Microsoft Visual Studio ;
- > **The Gimp** <-> Photoshop (il est même compatible avec les greffons !) ;
- > **Thunderbird/Kontact/Evolution** <-> Microsoft Outlook ;
- > **Kopete/Gaim** <-> Yahoo messenger!, MSN, Jabber, etc. ;
- > **Klamav** <-> McAfee et consorts.

Cette liste est loin d'être exhaustive. Fiez-vous surtout à vos propres tests et à vos appréciations personnelles. Par exemple, vous trouverez peut-être qu'en termes de logiciels et de fonctionnalités couvertes, KOffice va beaucoup plus loin qu'OpenOffice.org (il intègre un logiciel de gestion de projets par exemple) ou même que MS Office (il intègre des outils de dessin vectoriels à rapprocher de produits comme Adobe Illustrator). Mais cette suite bureautique souffre d'un terrible manque de filtres pour accepter les travaux effectués avec d'autres logiciels. Donc, bons et mauvais points se côtoient... On notera par ailleurs que dans pratiquement toutes les distributions, CUPS PDF est installé. Donc, on imprime nativement des PDF depuis n'importe quelle application qui peut se servir d'une imprimante. Résultat, l'équivalent d'Adobe Acrobat Distiller est invisible pour l'utilisateur !

## Quel est (sont) le (ou les) Logiciel(s) libre(s) équivalents au logiciel propriétaire que j'utilise tous les jours ?



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

Je me permets ici d'attirer votre attention sur plusieurs points annexes au transfert de données et à l'usage de nouveaux logiciels du monde propriétaire vers le monde libre :

----> l'absence de publicité n'est pas anodine, car il est extrêmement difficile pour un utilisateur d'accepter l'interface polluée et saturée de **MSN Live** ou de certains *sharewares* du monde propriétaire. Le retour sur des outils qui ne comportent aucune publicité est très agréable.

----> la lecture des différents formats de données n'est pas toujours automatique sous Linux. Depuis que les brevets logiciels sont devenus une source de problèmes conséquents, certains formats ont même volontairement été retirés de certaines distributions (ex. : MP3 dans la distribution Ubuntu). Cela dit, il existe des outils permettant de lire ces formats dans les dépôts non officiels de votre distribution. Par ailleurs, il est bon de savoir que si vous voulez lire les formats vidéo propriétaires (WMV, realplayer 8 ou 9, etc.), vous aurez besoin du projet **win32codecs** qui est potentiellement hors-la-loi dans certains pays (mais a priori pas la France). De même, pour lire vos DVD commerciaux, vous aurez besoin de **libdvdcss** qui est aujourd'hui du fait de la loi DAVSI hors-la-loi (contournement de protection numérique pour la lecture de contenus sous le régime du droit d'auteur). Enfin, pour lire des formats de données comme le ACE ou le RAR, il vous faudra installer de petits logiciels non libres.

----> Donc, je vous conseille, si cela vous intéresse, de bien regarder les dépôts non libres (*non free*) et non officiels de votre distribution, ou encore, pour ceux qui sont concernés, les dépôts commerciaux d'Ubuntu (logiciels commerciaux, mais pas forcément payants – <http://archive.canonical.com/dists/>) et le projet du PLF (*Penguin Liberation Front* – <http://packages.freecontrib.org/plf/pool/>). Pratiquement tout est faisable, mais cela va demander de la patience et un peu d'huile de coude parfois...

Néanmoins, j'aimerais finir cet article sur une note optimiste. Si vous passez au Logiciel libre, c'est en partie pour changer de pratique. Alors, plutôt que de chercher frénétiquement le logiciel qui fait exactement la même chose que le logiciel « untel » sous Windows/Mac OS, recherchez la fonction dont vous avez besoin. Et je vous promets que de très bonnes surprises vous attendent tant les développeurs de logiciels peuvent être à l'écoute de leurs utilisateurs. Ce n'est pas toujours vrai, mais, dans le domaine bureautique par exemple, ils ont souvent une longueur d'avance ! Ensuite, souvenez-vous que changer d'informatique exige un peu de patience et de travail, alors ne désespérez-pas !

Sidoine Pierrel

LES ÉDITIONS  
DIAMOND

**NOUVEAU**  
VERSIONS NUMÉRIQUES

**RETROUVEZ UNE  
SÉLECTION DE NOS  
ANCIENS NUMÉROS  
EN PDF  
SUR NOTRE  
PLATEFORME IZIBOOK**

**CLIQUEZ ICI !**

<http://diamond.izibookstore.com>

LINUX  
MAGAZINE FRANCE

LINUX  
PRATIQUE

LINUX  
PRATIQUE  
ESSENTIEL

MISC  
Multi System & Internet Security Cookbook

Open  
Silicium

## Que sont les fichiers d'extension .gz, .tar, .tar.gz, .tar.bz2, etc. ?

Pour répondre simplement, on peut dans l'immédiat répondre que ces fichiers sont des fichiers compressés, à l'instar des fichiers `.zip` que vous utilisez parfois sous un système Windows. Plus précisément, nous sommes ici en face de 2 actions différentes sur des fichiers : 1) l'archivage, 2) la compression.

**Tar** est un programme qui permet d'archiver et de désarchiver un ensemble de fichiers, sans compression. C'est l'un des plus utilisés dans le monde Unix. On obtient alors des fichiers d'extension `.tar`. Or, Tar est la plupart du temps conjointement utilisé avec **Gzip** (*GNU zip*), un outil de compression qui génère des fichiers d'extension `.gz`. Au final, on obtient une archive compressée d'extension `.tar.gz`, que l'on appelle parfois « fichier *tarball* ». Notez que c'est souvent sous cette forme que vous trouverez les sources d'un logiciel mis à disposition sur Internet.

La procédure de compression sous-entend le choix d'un taux de compression. Celui de Gzip est tout à fait raisonnable, mais il existe un autre outil libre : il s'agit de **Bzip2**, qui génère des fichiers d'extension `.bz2`. Cet outil est plus lent que Gzip, mais beaucoup plus puissant. Et comme pour Gzip, il peut s'utiliser conjointement avec le programme Tar, ce qui nous donne des archives compressées d'extension `.tar.bz2`.

Tous ces outils de compression et d'archivage sont pré-installés sur la plupart des distributions actuelles. Néanmoins, sachez qu'il est très simple de les installer à l'aide de votre gestionnaire de paquets de logiciels (paquets `tar`, `gzip`, `bzip2`).

Un ensemble de petites commandes (à saisir dans votre terminal de commandes) peuvent être utilisées pour manipuler ces fichiers :

----> Pour lire le contenu d'un fichier compressé (SANS le décompresser) :

```
zcat fichier.gz ou bzcata2 fichier.bz2
```

----> Pour compresser un fichier :

```
gzip fichier ou bzip2 fichier
```

----> Pour décompresser un fichier :

```
gunzip fichier.gz ou bunzip2 fichier.bz2
```

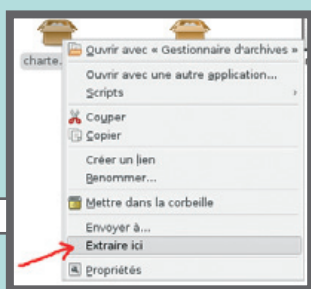


Fig. 1

Sachez que quel que soit votre environnement de bureau, votre navigateur de fichiers vous permet la plupart du temps de désarchiver/décompresser un élément via un simple clic droit sur celui-ci (Fig. 1). Éventuellement, vous n'aurez qu'à préciser où votre navigateur devra enregistrer le contenu de l'archive ou le fichier original. Bien entendu, ceci ne fonctionnera qu'à condition de disposer des outils mentionnés ci-dessus ! Votre navigateur de fichiers ne fait qu'utiliser ces outils...

Par ailleurs, il existe des utilitaires graphiques qui vous permettent de gérer vos archives et fichiers compressés, comme **File-Roller** (Gnome) ou encore **Ark** (pour KDE).

Il est également possible de désarchiver/décompresser une archive très rapidement en lignes de commande :

----> Pour décompresser une archive d'extension `.tar.gz` :

```
tar -xzvf archive.tar.gz
```

----> Pour décompresser une archive d'extension `.tar.bz2` :

```
tar -xjvf archive.tar.bz2
```

L'option `-z` permet de spécifier à l'outil Tar qu'il s'agit d'une archive compressée avec Gzip, et donc, qu'il faut la décompresser. De même, l'option `-j` indique qu'il s'agit d'une compression Bzip2. L'option `-x` indique que l'on souhaite extraire le contenu de l'archive. L'option `-v` (facultative) permet d'effectuer l'opération en mode *verbose*, ce qui signifie qu'on pourra visualiser dans le terminal de commandes la liste des fichiers traités et la progression de la décompression. Enfin l'option `-f` (à toujours placer en dernier), permet d'introduire le fichier concerné. Vous l'aurez compris, les options `-z` et `-j` ne sont à utiliser que si l'on est bien en présence d'une archive compressée (et non d'une archive simple en `.tar`).

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>  
<http://www.unixgarden.com>  
<http://www.editions-diamond.fr>

→ Pour lire le contenu d'une archive SANS la désarchiver :

```
tar -tf archive.tar.
```

Et si elle est compressée : `~$ tar -tzvf archive.tar.gz` ou `~$ tar -tjvf archive.tar.bz2` selon le cas.

→ Pour créer une archive :

Vous constaterez que les commandes sont similaires aux précédentes, à la différence près qu'il faut remplacer le `x` de l'extraction par un `c` (pour *create*) :

```
tar -czvf nom_archive.tar.gz repertoire1/ repertoire2/ ...
```

Vous pouvez nommer l'archive comme il vous plaira, en indiquant bien l'extension `.tar.gz`, ou l'extension `.tgz` (qui sont équivalentes). Puis, suivent les répertoires et fichiers que vous souhaitez inclure dans l'archive, séparés par une espace. Pour créer une archive compressée en `.bz2`, on utilisera :

```
tar -cjvf nom_archive.tar.bz2 repertoire1/ repertoire2/ ...
```

Notez pour finir que la création d'une archive peut également se faire via l'interface graphique de votre navigateur de fichiers (un clic droit sur un répertoire affiche le menu contextuel qui vous propose une option du genre « **Créer une archive...** » (Fig. 2) ; vous n'aurez qu'à préciser le nom et l'extension de l'archive à créer (Fig. 3).

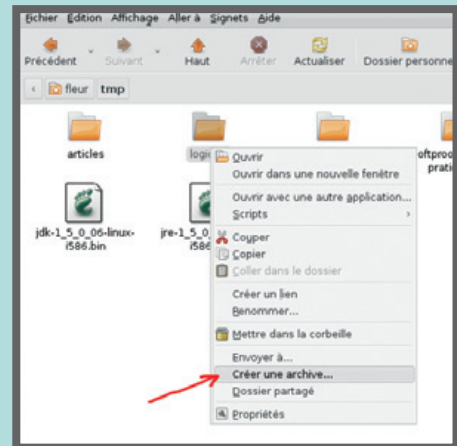


Fig. 2

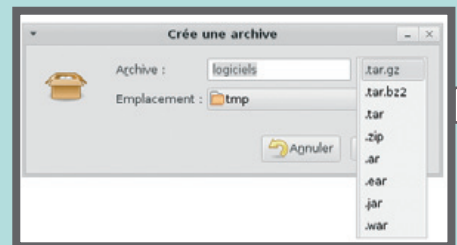


Fig. 3

Fleur Brosseau

## Comment faire une capture d'écran sous Linux ?

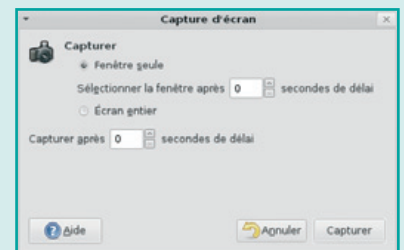
Plusieurs méthodes s'offrent à vous pour effectuer une capture d'écran. La première, et la plus simple, consiste à utiliser la touche [Impr écran] ou [Print Screen] de votre clavier. L'ensemble de l'écran est alors immédiatement capturé et une fenêtre de dialogue apparaît pour vous demander où enregistrer l'image. Pour ne capturer que la fenêtre active, utilisez la combinaison de touches [Alt]+[Impr écran] ou [Print Screen].

Dans certains cas, en fait selon la distribution utilisée, la touche [Impr écran] n'est pas fonctionnelle. Dans ce cas, vous pourrez utiliser l'une des solutions suivantes :

→ Le logiciel de retouche d'images **The Gimp** (installé par défaut sur la majorité des distributions) possède une fonction de capture d'écran. Dans la fenêtre principale de The Gimp, cliquez dans le menu **Fichier** -> **Acquisition** -> **Capture d'écran**. Une fenêtre apparaît alors à l'écran pour vous proposer de capturer l'écran entier ou une seule fenêtre, en précisant le nombre de secondes de délai avant la capture (le temps de sélectionner une fenêtre éventuellement ou d'arranger l'apparence de l'écran).

→ Par ailleurs, votre environnement de bureau (Gnome, KDE, etc.) possède certainement un accessoire de capture d'écran. KDE, par exemple, vous propose l'outil **Ksnapshot**. Vous le trouverez dans le menu **K** -> **Multimédia** -> **Graphisme**. Au lancement, une petite boîte de dialogue apparaît ; elle vous permet de choisir un mode de capture (plein écran, fenêtre unique ou zone de l'écran) ainsi que de fixer un délai (en secondes) avant la capture.

Idem chez Gnome : vous trouverez dans le menu **Applications** -> **Accessoires**, l'outil de capture d'écran. En revanche, vous ne pourrez ici que capturer la totalité de l'écran. En effet, dès que vous cliquez sur l'entrée **Prendre une capture d'écran**, c'est tout ce qu'il se passe ! Une boîte de dialogue vous permet ensuite de préciser l'emplacement où doit être sauvegardée la capture. Vous ne pourrez pas capturer une fenêtre en particulier, ni une zone quelconque de l'écran.





Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>  
<http://www.unixgarden.com>  
<http://www.editions-diamond.fr>



----> Pour finir, nous pouvons citer l'incontournable **Imagemagick** (<http://www.imagemagick.org/script/index.php>), un ensemble de commandes destinées à la manipulation d'images (découpage, redimensionnement, conversion dans un autre format, ajout de filtres, etc.). Bon nombre d'applications graphiques utilisent d'ailleurs les utilitaires d'Imagemagick, mais ce dernier peut également s'utiliser en ligne de commande. Ainsi pour capturer une portion de l'écran, on saisira la commande suivante dans un terminal :



`import -pause X nom_capture.ext`

Le **X** désigne un chiffre représentant le nombre de secondes de délai, avant que le curseur de la souris ne se transforme en petite croix ; cette dernière vous permet ensuite de sélectionner la zone de l'écran que vous souhaitez capturer. Vous pouvez nommer la capture comme vous le souhaitez, et lui donner l'extension de votre choix (à condition que le format soit supporté par Imagemagick ; cela dit, ce dernier supporte pas loin d'une centaine de formats !). Par exemple :

`import -pause 5 logo_linuxpratique.png`

Au bout de 5 secondes, je pourrai sélectionner à l'écran le logo que je souhaite capturer et ce dernier sera enregistré sous le nom de `logo_linuxpratique`, au format PNG, et dans le répertoire dans lequel je me situais lorsque j'ai saisi la commande.

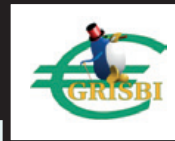
Pour capturer l'écran entier, la commande est quelque peu différente. Il faut ajouter l'option `-window root` :

`import -pause 5 -window root logo_linuxpratique.png`

Vous n'avez rien à faire, la capture de l'écran dans son ensemble s'effectue automatiquement (ici après 5 secondes). Ceci est notamment utile si vous devez au préalable dérouler un menu. En effet, la souris ne peut pas à la fois sélectionner une zone de l'écran et interagir avec l'interface graphique...

Fleur Brosseau

## Comment faire ma comptabilité personnelle sous Linux ?



On peut retenir trois principales applications de comptabilité sous Linux : il s'agit de **Grisbi** (<http://www.grisbi.org/>) – qui, au passage, est disponible en libre téléchargement pour Windows –, de **GnuCash** (<http://www.gnucash.org/>) et de **KmyMoney** (<http://kmymoney2.sourceforge.net/>) destiné à l'environnement de bureau KDE.

Cependant, **Grisbi** semble être le premier choix des utilisateurs. Son interface francisée, facile à prendre en main et intuitive fait de **Grisbi** l'outil idéal pour une utilisation régulière de base (comptes bancaires, comptes de dettes, échéancier, création de rapports, opérations de rapprochement, etc.). Il respecte tout à fait l'esprit de la comptabilité française. Par ailleurs, une documentation complète (et en français) concernant son utilisation est disponible sur le site officiel. Parallèlement, **KmyMoney** offre les mêmes fonctionnalités que **Grisbi**. Il est lui aussi clairement orienté comptabilité personnelle. Enfin, **GnuCash** est vu comme un outil légèrement plus complexe ou du moins plus difficile à appréhender, plutôt axé sur la comptabilité professionnelle.

Les utilisateurs d'**Intuit Quicken**, qui génère des fichiers au format QIF, pourront sans souci importer leurs anciens fichiers de comptabilité sous **Grisbi**, **GnuCash** ou **KMyMoney** (vous devez avoir autant de fichier \*.QIF que vous aviez de comptes.). Les utilisateurs de **MS Money**, en revanche, devront tout d'abord exporter leurs fichiers dans le format QIF avant l'importation.

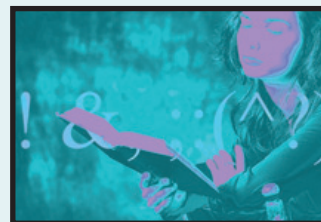
F. B.

## Comment lire mes fichiers PDF sous Linux ?

Quelle que soit votre distribution, vos fichiers PDF pourront être lus par différents programmes. Ceux-ci comportent, pour la plupart, les fonctionnalités habituelles qu'on attend d'un tel utilitaire, comme différents modes d'affichage, une fonction de recherche par mot-clé, des fonctions de zoom, navigation entre les pages, impression et enregistrement.

Le bureau **Gnome** intègre depuis quelque temps un lecteur PDF vraiment très agréable à utiliser, nommé **Evince**. Il peut intégrer également un lecteur nommé **gpdf**, moins abouti qu'**Evince** (dont il est le prédécesseur), mais tout à fait correct pour un usage de base. Le bureau **KDE** intègre, quant à lui, le lecteur **Kpdf**, tout aussi performant que ses homologues.

Il existe également une suite logicielle nommée **Xpdf**, puissante, mais peu séduisante aux yeux des utilisateurs à cause de son interface graphique pour le moins austère et pas vraiment intuitive. Elle inclut non seulement un visionneur PDF, mais aussi un extracteur de texte, un convertisseur PDF vers Postscript, et quelques autres utilitaires.



Sachez que l'éditeur **Adobe** propose également une version d'**Acrobat Reader** pour Linux (à récupérer à l'adresse : <http://www.adobe.com/fr/products/acrobat/readstep2.html>), quasi identique à la version pour les systèmes Windows.

F. B.

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Comment lire mes CD audio/DVD/DivX sous GNU/Linux ?

Pour être en mesure de lire des CD ou DVD sur son ordinateur, il y a quelques points de détail à maîtriser au préalable. Tout d'abord, le fichier `/etc/fstab`. En ce qui nous concerne, il se doit de comporter au moins une ligne dédiée au lecteur multimédia, du type :

```
/dev/hda /media/cdrom1 udf,iso9660 user,noauto
```

Où `/dev/hda` représente le fichier d'accès à votre lecteur. Typiquement, le nom de ce fichier correspond au branchement de votre lecteur sur le bus concerné de votre carte mère. Le motif de nommage est alors `/dev/hd` pour les lecteurs en IDE, et la dernière lettre, ici le « a » indique qu'il s'agit du premier lecteur, celui qui correspond normalement au port marqué « IDE1 » sur votre carte mère. Si vous avez, comme moi, un disque dur en Serial ATA, votre premier lecteur DVD est donc atteignable via `/dev/hda`, et le second, si vous en avez un, par `/dev/hdb`, et ainsi de suite.

Ensuite, le deuxième champ de cette ligne de configuration, correspond au point de montage, c'est-à-dire le répertoire de votre arborescence dont le contenu sera remplacé par celui des disques que vous placerez dans votre lecteur lors d'une opération de montage. Toutefois, s'il faut que votre `/etc/fstab` soit configuré de façon opérationnelle, il est bon de noter tout de suite que la lecture d'un CD ou d'un DVD ne requiert pas que vous vous occupiez plus avant des histoires de montage et d'arborescence.

Le champ suivant de la ligne précise le type de système de fichier que l'on peut s'attendre à trouver, ici `udf` et `iso9660` correspondent aux systèmes de fichiers communément trouvés sur les CD et DVD. Le dernier champ comporte, lui, des informations sur la politique de montage et de distribution des droits d'accès au contenu du périphérique. Puisque nous nous éloignons là de mon propos initial, je vous renvoie vers la page de manuel de la commande `mount` où vous pourrez trouver plus d'informations. Et maintenant, attaquons le vif du sujet !

Pour lire un film sur DVD, rien de plus simple ! Une fois validée l'étape précédente, insérez le DVD de votre choix dans votre lecteur multimédia, lancez une application libre de lecture de vidéo et cherchez l'option de lecture de DVD qu'il propose. Elle se trouve généralement dans le menu **Fichier** de l'application, sinon dans son interface graphique... Si des messages d'erreur viennent troubler le lancement de votre film, cherchez à préciser à votre logiciel quel périphérique utiliser comme lecteur. Dans notre cas, nous précisions `/dev/hda`. Sinon, c'est que vous avez un DVD protégé contre les lecteurs standards qui impose un logiciel de lecture précis, généralement fourni avec le DVD, mais incompatible avec votre système d'exploitation libre. Je vous recommanderais alors d'aller rapporter ledit DVD chez le marchand en faisant un esclandre !

Les Logiciels libres de lecture de vidéo les plus réputés sont : **VLC** (<http://www.videolan.org/vlc/>), **Xine**, décliné en plusieurs versions comme `xine-ui`, `gxine` et `totem-xine`, basé à chaque fois sur le même moteur (<http://xinehq.de/>), **MPlayer** (<http://www.mplayerhq.hu/>), **GStreamer** avec l'implémentation `totem-gstreamer`.

Dans la cas d'un CD audio, rappelons tout d'abord qu'il est inutile de vouloir monter dans son système de fichier un tel média, c'est tout simplement impossible à ma connaissance... Si vous souhaitez extraire les pistes musicales d'un CD audio de cette manière, il vous sera recommandé de passer par des logiciels tels que **Sound Juicer** (pour Gnome, <http://www.burtonini.com/blog/computers/sound-juicer/>), **Grip** (pour Gnome, <http://nostatic.org/grip/>) ou **Kaudiocreator** (pour KDE, <http://www.icefox.net/programs/?program=KAudioCreator>) dotés d'une interface graphique, ou encore par les commandes `cdda2wav` ou `cdparanoia` qui s'acquittent très bien de la tâche...

Ensuite, vous avez plusieurs choix pour lire votre CD. Le premier, le plus simple, est d'appuyer sur le bouton de lecture du lecteur. Vous ne vous étiez jamais demandé pourquoi il y avait deux boutons sur la façade de votre lecteur ? Un seul suffirait pour ouvrir et fermer le tiroir... Eh bien, maintenant vous savez ! Vous pouvez généralement lancer la lecture et passer à la piste suivante grâce à ce bouton « superflu ». L'avantage de cette technique est qu'elle ne consomme aucune ressource matérielle, son principal inconvénient est de n'offrir qu'un contrôle limité de l'opération.

Une deuxième technique est d'utiliser un logiciel dédié à la lecture de CD tel que **Gnome-CD** ou votre lecteur de vidéo ou encore des lecteurs élaborés de musique tel que **Beep Media Player** (BMP, <http://beep-media-player.org/>), **Rhythmbox** (intégré par défaut dans le bureau Gnome, <http://www.gnome.org/projects/rhythmbox/>) ou **Amarok** (<http://amarok.kde.org/>). Voilà une bonne occasion d'installer et de tester plein de logiciels pour trouver celui qui vous correspond le plus...

Simon Descarpentries



## Comment graver un CD/DVD sous GNU/Linux ?

Graver un CD sous GNU/Linux, rien de plus simple ! Et ce, quelle que soit votre distribution. Comme toujours dans le monde du Libre, il existe plusieurs moyens d'arriver à vos fins et, parmi ceux-ci, deux grandes catégories :

- > utiliser une commande en mode texte dans un terminal ;
- > utiliser une application en mode graphique.

Abordons de suite le deuxième cas. Si vous souhaitez graver des CD ou DVD via une interface graphique, il existe plusieurs programmes qui peuvent répondre à vos besoins. Si vous êtes un utilisateur de KDE, votre préférence se dirigera sûrement vers **K3b** (<http://www.k3b.org/>), un logiciel complet, si ce n'est LE logiciel de gravure, que les utilisateurs de logiciels propriétaires dans ce domaine n'auront aucun mal à prendre en main.

Si, au contraire, votre préférence penche pour GNOME, vous vous laisserez séduire par **Gnomebaker** (<http://gnomebaker.sourceforge.net/v2/>), le boulanger de GNOME, qui vous permettra de mettre vos galettes au four. Devant le succès quasi immédiat de K3b à sa sortie, il y a plus de trois ans, toute autre tentative de développement d'un utilitaire libre et graphique de gravure fut découragée, les ressources se concentrant en effet sur le premier projet. Mais cette époque est révolue et, un beau jour, un utilisateur de GNOME un peu perdu dans l'interface graphique chargée de K3b se décida à coder une application de gravure pour GNOME, plus épurée. Le résultat est simple et efficace !

Notons également que l'explorateur de fichier par défaut dans GNOME, à savoir **Nautilus**, comporte également un module de gravure de fichier, plus ou moins ergonomique. On pourra, par ailleurs, citer **Graveman** (<http://graveman.tuxfamily.org/>) et **Brasero** (anciennement Bonfire, <http://sourceforge.net/projects/bonfire/>), tous deux moins connus que leurs homologues mentionnés ci-dessus, mais néanmoins tout à fait fonctionnels et efficaces.

Ensuite, loin de la guerre des bibliothèques graphiques (GTK contre QT, GNOME contre KDE), il existe encore **X-CD-Roast** (<http://www.xcdroast.org/>), un utilitaire de gravure pour X, dont l'interface graphique définit sa propre notion de l'intuitivité... Blague à part, ce projet est maintenu depuis les premières heures de la gravure sous GNU/Linux et il s'agit donc d'une sérieuse alternative qu'il convient de considérer dans notre cas.

Mais toutes ces solutions graphiques sont toujours basées sur un même fondement plus sobre : les commandes en mode console. Et puisque c'est possible et même fondamental de pouvoir le faire, voyons maintenant comment graver à partir de la console.

Dans ce domaine, un logiciel a tenu le haut du pavé pendant des années. Une commande très complète, intuitive et vraiment pratique : **cdrecord**. Cette commande est toujours disponible, mais à la suite d'un changement de licence (GNU/GPL vers CDDL), le projet Debian a décidé de fournir une alternative aux outils développés par le sulfureux Jörg Schilling (sûrement en prévision d'un avenir plus orageux...). C'est ainsi qu'est né le projet « Debburn » fournissant un set d'applications de gravure en mode console, sous licence GNU/GPL. Via son CDR-Kit, ce projet propose donc désormais la commande **wodim** parfait remplaçant de **cdrecord**. Précisons tout de suite que **wodim** n'a pas pour objectif de rester un clone de l'application qu'elle remplace, et que, comme tout bon *fork*, elle évoluera selon son propre chemin.

Alors, comment graver une image ISO à partir d'une console ? Tout simplement via la commande :

```
wodim speed=32 dev=/dev/hdd mon_fichier.iso
```

en supposant que **/dev/hdd** soit bien le fichier d'accès à votre graveur (dans cette commande et les suivantes, remplacez simplement **wodim** par **cdrecord**, selon l'outil que vous utilisez). Et il est possible de paramétrer plus finement l'opération comme :

```
wodim fs=64m speed=32 dev=/dev/hdd -audio -overburn *.wav
```

qui grave toutes les pistes audio du répertoire courant sur un CD qui sera audio et compatible avec n'importe quelle chaîne Hi-fi, à l'aide d'une mémoire tampon réservée de 64 Mo et en précisant que l'on souhaite pouvoir graver plus que la capacité nominative du CD-ROM sur lequel on opère.

Pour effacer un CD-RW, on utilisera la commande :

```
wodim blank=all speed=12 dev=/dev/hdd
```

Pour effectuer une copie « à la volée » entre un lecteur et un graveur :

```
wodim -v -dao fs=32m -overburn speed=12 dev=/dev/hdd -isosize /dev/hdc (en supposant ici que /dev/hdc soit le fichier d'accès au lecteur source).
```

Voilà, avec ces bases et le fichier de manuel *man wodim* (ou *man cdrecord*), vous êtes désormais de taille à créer vos propres formules magiques et à déchaîner avec précision la foudre laser qui marque à jamais vos tranches de silice...

Simon Descarpentries



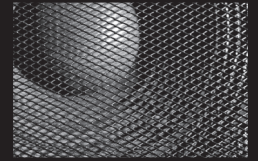
Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

# Pourquoi certaines distributions n'incluent-elles pas de logiciel permettant de lire ou de créer des fichiers audio MP3 ?



## Petit historique

Le codec MP3 est un algorithme de compression du son, permettant de réduire d'un facteur 10 la taille d'un fichier son 16 bits tel que le MS-Wav ou Sun-Au. Cette technique de compression du son fut achevée officiellement en 1992, suite à la collaboration de l'institut Fraunhofer-Gesellschaft et de l'entreprise Thomson, dans le but d'offrir une troisième couche de compression du son au codage du *Moving Picture Experts Group* aussi connu sous l'acronyme MPEG.

Alors que la performance semble dans un premier temps minime, le format MP3 acquiert ses lettres de noblesse grâce à l'internet, où il permet pour la première fois dans des temps raisonnables, avec les débits de l'époque, de transférer de la musique entre deux ordinateurs terminaux du grand réseau.

Devant la popularité du format, qui incite à la conversion numérique de ses vieux vinyles, et permet rapidement de se procurer sur le net, de manière illégale, les derniers morceaux à la mode, une nouvelle génération de baladeurs musicaux voit le jour, les fameux clés USB et lecteurs MP3. De plus, le format est choisi pour des projets de grande envergure, comme la mise à disposition d'archives sonores, ou encore la diffusion de radio sur Internet sous forme de flux MP3.



## Le hic

Or, il est un fait que tout le monde s'est empressé d'oublier dans l'euphorie de l'essor de cette nouvelle technologie, permettant dès 1998 aux lycéens du monde entier de détourner ces hautes technologies pour télécharger sur le net les génériques des dessins animés de leur enfance, c'est que le format MP3 n'a jamais été libre ! Son utilisation à tout va fut tolérée par ses pères fondateurs pendant les balbutiements de la technologie, et tant qu'ils n'avaient pas de cibles solvables à viser, mais dès que le marché des lecteurs MP3 prit son envol, il fut rapidement rappelé aux sociétés productrices que l'utilisation de techniques de compression MP3 était couverte par des brevets, et qu'il convenait de s'acquitter de redevances (et non des moindres : 5\$ par copie diffusée de logiciel pouvant lire des MP3 ou 60 000\$ à l'année) auprès des organismes détenteurs des brevets.

Dès lors, la chasse aux utilisations illégales du MP3 fut lancée. Et les encodeurs libres de musique au format MP3 durent d'abord se cacher, puis cesser toute activité, à l'instar des développeurs d'eMule, croulant sous les poursuites judiciaires (et gageons qu'en France, la loi DADVSI permettra à des organismes privés comme la SACEM de faire renoncer l'équipe de développement de VLC d'ici peu... Équipe dont les développeurs risquent 6 mois de prison et 30 000 euros d'amende pour avoir permis de lire des DVD avec VLC. Voir : <http://www.videolan.org/>).

C'est donc pour des raisons philosophiques, « ce n'était pas libre », puis des raisons pratiques, « finalement, c'est interdit », que, peu à peu, le codec MP3 a officiellement disparu des distributions GNU/Linux.

## Des remplaçants

Tout d'abord, concernant l'écoute de fichiers MP3, même si en effet la décompression de fichiers MP3 fait appel à des bibliothèques propriétaires, il reste possible, quelle que soit votre distribution, de bidouiller un peu pour obtenir de votre logiciel de lecture favori qu'il interprète vos morceaux.

Et concernant la compression, le candidat idéal pour remplacer le MP3 est le format Ogg Vorbis, qui obtient des résultats équivalents, voire meilleurs, au niveau du taux de compression. Ce format est entièrement libre, et certains lecteurs MP3 supportent également l'Ogg (comme les produits de la société iRiver). Voir : <http://www.vorbis.com/> et <http://ptaff.ca/ogg/>.

Mais l'Ogg Vorbis n'est pas le seul candidat. En effet, il existe aussi un format libre nommé « MPC », ou « MPEG+ », qui, partant de la même base libre que le MP3 (à savoir le MPEG-2), apporte, lui aussi, son lot d'améliorations au codage pour obtenir finalement quelque chose d'assez abouti, tant au niveau du taux de compression que de la qualité de son restitué (voir <http://www.musepack.net/>).

On peut également citer l'algorithme libre de compression audio sans perte FLAC, qui conviendra aux plus exigeants en matière de qualité sonore (voir <http://flac.sourceforge.net/>).

Comme toujours, le monde libre fourmille de solutions...

Simon Descarpentries

## Si je faisais tout en tant que root, ce serait bien plus pratique, non ?

Oui, dans une certaine mesure, c'est vrai : tout faire sous le compte root est plus « simple ». Mais en réalité, cela ne va pas sans poser de problèmes...

### Systèmes mono-utilisateur vs systèmes multi-utilisateurs

Tout faire sur un seul et même compte est le parti pris des systèmes mono-utilisateurs, qui n'ont pas implémenté la gestion des comptes. On se rappelle (pas toujours...) de Windows 95, par exemple, ou, plus vieux encore, de DOS (là on cherche vraiment loin...). Pourquoi fonctionnaient-ils ainsi ? Disons que tout est affaire de politique. Le système mono-utilisateur a certains avantages : sa simplicité fait qu'il est facile à prendre en main pour quelqu'un qui n'a pas envie de trop se casser la tête avec l'informatique ; tout se passe sous un seul compte, parfois aucun mot de passe n'est demandé... Bref, c'est facile. A côté de ça, il peut paraître fort contraignant à certains utilisateurs d'avoir des mots de passe à répétition...

Mais ce qui est simple pose parfois des problèmes ! Un exemple : si vous attrapez un virus sur votre système mono-utilisateur, c'est potentiellement l'ensemble du système qui pourra être détruit. Vous pouvez très bien utiliser votre Linux « comme si » c'était un système mono-utilisateur, mais c'est un peu à vos risques et périls...

### Des droits pour chaque utilisateur

L'idée dans l'utilisation d'un système multi-utilisateur, c'est de partager les tâches, et d'associer des droits particuliers à chaque utilisateur. Par exemple, si on souhaite monter un serveur de mail sur son système, on crée un compte « mail » auquel on accorde uniquement certains droits : celui d'écrire dans quelques répertoires du disque, d'utiliser tant d'espace mémoire, etc. Ainsi, si le compte venait à être piraté (ex : s'il y a une faille de sécurité dans votre serveur mail), le pirate ne pourra avoir que les droits initialement conférés au compte, et pas plus, car c'est root qui fixe les règles du jeu.

La nuance devient très importante dès lors que l'on souhaite avoir un peu de sécurité. Et il semble difficile aujourd'hui de se passer de sécurité sur son système. C'est un peu comme si on créait des compartiments étanches : si l'un des compartiments est détruit ou envahi, les autres continuent à être opérationnels.

### Pourquoi éviter certaines manipulations sous root ?

Même sans compter avec les virus, vous pouvez vous-même être un risque pour votre système. Et l'un des premiers enseignements, surtout pour les débutants qui utilisent Linux (ou tout autre système UNIX, puisque le principe est commun à tous ces systèmes), est de ne pas tout faire sous le compte root, justement. Pourquoi cela ? Vous allez vite comprendre. Le système tel qu'il a été conçu est segmenté. Il y a root qui peut intervenir sur tout le système, c'est-à-dire sur toute la partition /. Et il y a la partie réservée aux utilisateurs : en général, elle se trouve sous la partition /home, dans laquelle on trouve les espaces réservés à chaque utilisateur : par exemple, /home/Bill-G est le répertoire de Bill G. Lorsque notre ami Bill G. utilisera son système, l'ensemble de ses propres fichiers seront créés sous ce répertoire : par exemple, son bureau se trouve ici : /home/Bill-G/Desktop. Le compte de Bill G. est normalement restreint à ce répertoire (avec quelques exceptions, comme /tmp, le répertoire temporaire, sur lequel il a le droit d'écrire). Ainsi, si Bill G. décide par le plus grand des hasards de se rebeller contre son système, et fomenté le plan machiavélique de tout effacer sur le disque dur, il va essayer de saisir la commande suivante :

```
bill@g $ rm -rf /
```

et voilà ce qui se passe alors :

```
bill@g $ rm -rf /
```

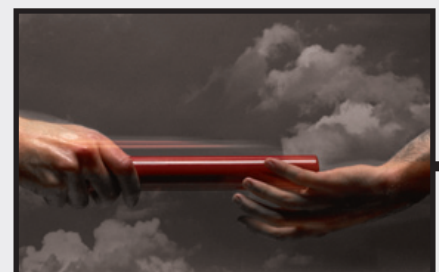
```
rm: cannot remove '/': permission denied
```

Eh oui ! La permission lui est refusée, car il n'a pas le droit d'intervenir directement sur le répertoire racine. Les fichiers situés sur la partition racine (/) sont la propriété exclusive du compte root, le super-utilisateur. Seul root pourrait les effacer.

### Des passe-droits

Mais que se passe-t-il si je dois simplement exécuter une commande sous root (par exemple installer un nouveau programme) ? Est-ce que je dois redémarrer le système en entier ? Bien sûr que non... Il existe des « ponts » pour passer d'un utilisateur à un autre. La manière la plus simple est d'utiliser la commande `su`. Elle permet de changer d'utilisateur.

PERMISSION DENIED



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>  
<http://www.unixgarden.com>  
<http://www.editions-diamond.fr>

Exemple, je suis toujours Bill G., et je voudrais être root pendant quelque temps :

```
bill@g $ su root
Password: *****
root@g #
```

Et voilà, je suis root, je peux installer un *package*, supprimer des logiciels, des fichiers système, etc. Attention tout de même, Bill G. logué en tant que root pourrait tout effacer ! Il ne le ferait d'ailleurs peut-être pas intentionnellement, mais parfois sur le clavier, il peut arriver qu'on glisse sur une touche...

Lucie Robin

## J'ai oublié mon mot de passe, comment faire ?

Si, par malchance, vous avez oublié votre mot de passe, il n'est pas vraiment question de le « récupérer », mais plutôt d'en créer un nouveau... Nous allons, dans un premier temps, considérer le cas de la perte du mot de passe de votre compte personnel, autrement dit, impossible d'ouvrir une session lorsque vous êtes face à l'interface graphique de connexion.

Pour remédier à ce problème, au démarrage de l'ordinateur, lorsque vous êtes face à votre *bootloader* (Grub ou Lilo, qui vous permet de sélectionner le

système d'exploitation à lancer), sélectionnez la ligne *recovery mode* ou *rescue mode*, autrement dit le « mode minimal », à l'aide des curseurs fléchés de votre clavier. Dans ce mode, vous serez connecté à votre système en mode texte, en tant qu'administrateur. Attendez l'invite de commande, puis saisissez : `passwd <votre_login>`. Une nouvelle ligne s'affiche alors : `Enter new Unix Password`. Saisissez alors votre nouveau mot de passe, puis une deuxième fois pour confirmer (rien ne s'inscrit à l'écran pendant la frappe, c'est normal). Si tout s'est bien déroulé, vous devriez obtenir le message : `passwd updated successfully`. Tapez `exit` pour quitter la console et redémarrer une session. Vous pouvez désormais vous connecter à l'aide votre nouveau mot de passe. Bien sûr, si l'oubli a lieu alors que vous avez déjà ouvert une session, effectuez la même manipulation dans votre terminal de commandes graphique.

En dehors de votre compte personnel, votre système d'exploitation a besoin d'un compte spécifique pour réaliser certaines tâches sensibles (souvent des tâches d'administration et de configuration). C'est pourquoi il existe un compte de « super-utilisateur » du système, communément appelé *root*. Si vous oubliez le mot de passe du compte root, le problème est déjà plus sérieux...

Les utilisateurs de K/X/Ubuntu noteront qu'il n'y a pas de mot de passe dédié au compte root. Ou plutôt, les développeurs de cette distribution ont fait le choix de désactiver le compte root par défaut. En effet, pour exécuter des tâches en tant que super-utilisateur, il faut précéder vos commandes par `sudo`, puis saisir le mot de passe de votre propre compte (le mot de passe du premier utilisateur que vous avez défini à l'installation). Si vous avez oublié ce mot de passe, il va falloir procéder comme ci-dessus.

Lors de l'installation d'autres distributions, on vous demandera de saisir à la fois un mot de passe pour root et un mot de passe pour votre compte personnel. En cas de perte du mot de passe root, le plus simple est de démarrer votre système sur un *live-CD* (type Knoppix, Ubuntu, Mandriva One, etc.). Lorsque vous êtes face à l'espace de travail, ouvrez un terminal de commandes. Il va falloir monter temporairement la partition racine de votre système sur un répertoire quelconque du système temporaire créé par le live-CD. Pour cela, créons par exemple un répertoire *racine/* dans le répertoire personnel de l'utilisateur. Bien sûr, vous êtes censé savoir ici sur quelle partition de votre disque dur se trouve la racine de votre système GNU/Linux... Dans notre cas, il s'agit de `/dev/hda3`. On saisit donc la commande suivante :

```
guest@machine:~$ mkdir racine
guest@machine:~$ su
guest@machine:~# mount /dev/hda3 racine/
guest@machine:~#
```

Notez que la commande `mount` requiert les droits de super-utilisateur, renseignez-vous auprès des concepteurs du live-CD utilisé pour connaître ce mot de passe (le plus souvent, il s'agit de « root » ou, parfois, aucun mot de passe n'est requis).

Ensuite, déplacez-vous dans le répertoire *racine* ; vous y trouverez logiquement le contenu de la partition racine de votre système. Éditez alors le fichier `/etc/passwd`, et éventuellement le fichier `/etc/shadow` selon le cas, qui listent les utilisateurs du système et leur mot de passe respectif. Dans ces deux fichiers, il va falloir supprimer le mot de passe associé à l'utilisateur root. Pour cela, à la ligne commençant par `root` (la première ligne en général), supprimez la série de caractères faisant office de deuxième champ (il s'agit du mot de passe chiffré). Par exemple :

```
root:X5Y12ef6z:0:0:root:/root:/bin/bash doit devenir root::0:0:root:/root:/bin/bash
```

Dans le fichier `/etc/shadow`, un unique caractère `x` se trouve dans le champ mot de passe. Cela signifie que le système utilise le paquetage *shadow-utils* mettant en œuvre le marquage des mots de passe en les déplaçant dans des fichiers annexes. Supprimez également le `x` de la ligne `root`.

Pour information, lorsque qu'un « ! » se trouve dans le champ mot de passe, cela signifie qu'aucun mot de passe n'a encore été défini pour le compte. Si une étoile « \* » se trouve dans le champ mot de passe, cela signifie que le compte ne permet pas de se connecter. Enfin, si le champ est vide, cela signifie qu'aucun mot de passe ne sera demandé lors de la connexion à ce compte (c'est d'ailleurs ce que l'on cherche à obtenir avec root).

Sauvegardez le fichier, puis redémarrez votre système (après avoir ôté le live-CD du lecteur). Ouvrez une session en utilisant votre *login* et votre mot de passe personnel. Puis, ouvrez un terminal de commandes et saisissez la commande `su`, puis [Entrée]. Aucun mot de passe ne vous est demandé (logique !), et vous devenez super-utilisateur du système. Sans plus attendre, définissez un nouveau mot de passe pour root à l'aide de la commande `passwd`. En effet, pendant le laps de temps où root n'a plus de mot de passe, votre système est en danger : n'importe qui pourrait se loguer en tant que root avec un mot de passe vide et corrompre votre système...

Fleur Brosseau



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Est-ce que Linux est sécurisé ?

Réponse surprenante : non ! Aucun système informatique n'est « sécurisé ». En fait, la question est mal posée : « Est-ce qu'il existe des mécanismes de sécurité sous GNU/Linux ? » serait préférable, d'autant qu'on peut alors répondre : oui. Est-ce que ces mécanismes sont efficaces ? Cette fois, notre réponse est sibylline : cela dépend... de la manière dont ils ont été mis en place, des risques encourus par le système, des objectifs de sécurité, etc.

Prenons un exemple : soit un système W qui sert à Monsieur X à naviguer sur le net et faire ses courses en ligne. Correction : soit un vieux système W, dont aucun patch de sécurité n'a été installé depuis des années (disons depuis 6 ans ; comme ça, on appellera notre système « W2000 »). Connectez ce système W2000 sur Internet. Qu'observe-t-on ? En une petite dizaine de minutes en moyenne, le système a été compromis par des virus, des vers, des pirates... Conclusion : notre système W2000 n'est (absolument) pas sécurisé !

Maintenant, si on prend exactement le même système et qu'on le place dans les locaux d'une enceinte militaire protégée, avec des gardes armés, et qu'on le conserve dans un coffre-fort de haute sécurité, accessible uniquement à des personnes habilitées dans une pièce ultra sécurisée, sans qu'il soit possible de connecter le système à un quelconque réseau informatique... On comprend bien que la problématique n'est pas la même, et que, malgré le fait que le système en lui-même puisse être une vraie passoire, les informations contenues à l'intérieur pourront parfaitement être sécurisées.

Donc, « sécurisé » ou « pas sécurisé » ne veut pas dire grand-chose. On peut se poser la question différemment : est-ce que mon système offre une sécurité adéquate pour l'usage que je compte en faire ?

### Quelle sécurité, pour quels besoins ?

Il est quasiment impossible de faire un tour exhaustif de l'ensemble des mécanismes de sécurité disponibles pour GNU/Linux tant ils sont nombreux. On citera néanmoins quelques outils :

----> Pour la maîtrise du trafic : à part l'utilisation de mots de passe solides (sur le sujet, on ne sera jamais assez préventif !), le b-a ba de la sécurité dans un environnement interconnecté est vraiment le *firewall* (ex. : **Netfilter**, <http://www.netfilter.org/>). Beaucoup de distributions Linux sont livrées aujourd'hui avec un firewall préconfiguré pour une utilisation standard du système, ce qui évite la problématique de la configuration du firewall pour quelqu'un qui n'est pas expert. Un firewall permet de maîtriser les flux entrants et sortants de votre machine, et c'est déjà beaucoup !

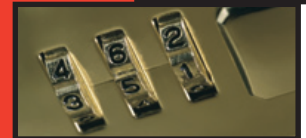
----> Pour l'e-mail : en pratique, même si ce n'est pas de la sécurité au sens strict du terme, on conseille généralement d'ajouter un antispam à son système, afin d'éviter de trier pendant des heures les vrais mails des annonces publicitaires. Ainsi, on risque moins de cliquer sur tous les programmes vérolés envoyés par les pirates. Jetez un œil à **Spamassassin** (<http://spamassassin.apache.org/>), c'est un excellent programme pour cela.

----> Pour la sauvegarde des données : si vous souhaitez conserver des données sensibles, pensez à programmer automatiquement des *backups* (**Zip**, **Amanda**, **Dump/restore**, etc.) ; mais attention cependant à la sécurité de ces informations (confidentialité, intégrité...).

Voilà déjà quelques mesures simples que l'on peut relativement facilement mettre en place et qui sont en pratique assez efficaces.

Vous êtes plutôt parano ? Pour les personnes qui sont plus soucieuses de leur sécurité, quelques outils supplémentaires s'imposent : **GnuPG** pour chiffrer ses mails, ses backups, etc., des outils de détection d'intrusion (**Snort**, **Aide**,...), et un scanner de vulnérabilités pour tester régulièrement sa machine (**Nessus**...) et vérifier que tout se passe comme on le voulait.

On pourra encore augmenter la sécurité de Linux en ajoutant des modules au noyau du système comme **Grsecurity**, qui, parmi une multitude de fonctionnalités de sécurité permet de rendre la pile non exécutable, et (sans entrer trop dans les détails) de rendre la vie beaucoup plus difficile à celui qui souhaiterait prendre le contrôle de votre système. A ce niveau, normalement, les contraintes de sécurité commencent à peser sur l'utilisation du système. Eh oui, plus vous restreindrez les éventuelles utilisations malveillantes de votre machine, plus vous serez vous-même contraint de veiller à votre propre utilisation de cette même machine ! Plus question de télécharger et d'exécuter n'importe quoi !



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Pourquoi n'y a-t-il pas de virus ou de spyware sous GNU/Linux ?

Tout d'abord, techniquement parlant, l'affirmation sous-entendue dans cette question est fautive. Il existe des virus pour les systèmes GNU/Linux. Ils sont simplement relativement rares et souvent de simples essais dits « *proof of concept* ».

Pour comprendre ce qui fait qu'un système est plus intéressant qu'un autre pour un développeur de virus ou de *spyware*, il faut surtout se pencher sur des considérations non techniques.

Pour que le développeur de virus puisse voir son œuvre faire les choux gras de la presse, des forums et des sites de news, il lui faut un terrain de jeu de taille respectable. Actuellement, la grande majorité des machines personnelles et d'entreprise utilise Windows (98/Me/2000/XP). C'est le facteur le plus à même d'expliquer la « richesse » de la faune virale pour ces systèmes. Ceci explique également que le système Mac OS X d'Apple soit moins affecté que Windows.

En second lieu, nous avons les habitudes des utilisateurs. Windows (du moins XP), comme Mac OS X et GNU/Linux, partage les privilèges entre utilisateur et super-utilisateur. Malheureusement, beaucoup d'utilisateurs Windows travaillent avec le compte d'administration. Celui-là même qui offre tous les pouvoirs aux applications exécutées (dont le navigateur Web, le client Mail, le logiciel de messagerie instantanée et toute application émettant et recevant des données via Internet). Mais ne vous y trompez pas, un code viral exécuté sous le compte *root* sur un système GNU/Linux fera sans doute autant de ravages que sous Windows.

Enfin, nous avons les considérations techniques. Les virus et autres *spywares* utilisent habituellement des failles de sécurité ou des erreurs de conceptions pour infecter un système. Dans le monde du Logiciel libre, la réactivité face à ces failles est très importante. Dès leur découverte, des centaines de développeurs tentent de les corriger et mettent à disposition des correctifs (ou *patches*) et/ou des versions mises à jour. Celles-ci sont ensuite reprises par les distributions qui les intègrent et les propagent aux utilisateurs sous la forme d'une mise à jour de la distribution ou du paquet concerné.

La propagation d'un code malicieux est donc rapidement enrayerée. Bien plus rapidement que ne pourrait le faire une société ne partageant pas les sources de ses programmes.

Enfin, le développement ouvert des applications et du système GNU/Linux permet une meilleure vérification du code source avant même qu'il n'arrive chez l'utilisateur. La communauté des développeurs est ainsi bien plus active et motivée que ne pourrait l'être une équipe, même importante, de développeurs travaillant en cercle fermé sur un projet.

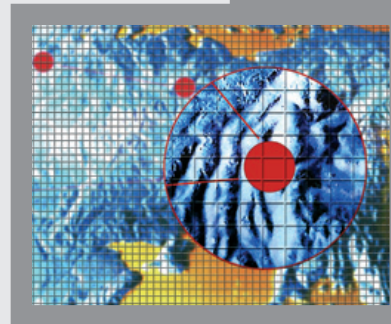
Mais si, un jour, GNU/Linux représente 90% des systèmes utilisés, que les utilisateurs y transposent leurs mauvaises habitudes et qu'on assiste à l'arrivée massive de logiciels propriétaires pour ce système... alors, GNU/Linux risque de voir sa collection de virus et de *spywares* s'étoffer de manière importante.

Denis Bodor

Mais on peut encore largement augmenter la sécurité en utilisant des modèles plus complexes de sécurité, comme le permettent par exemple **SELinux**, développé et utilisé par une célèbre agence de sécurité américaine. Ces modules permettent de définir une granularité extrêmement fine dans les permissions d'usage du système (des utilisateurs, des *process*, des *sockets*, etc.). A ce stade, vous devez avoir appliqué les principes fondamentaux de la sécurité : supprimer tous les programmes qui ne sont pas strictement nécessaires dans votre système, définir et formaliser une politique de sécurité tenue à jour et régulièrement auditée, etc. Autrement dit, une telle machine est bien loin de l'ordinateur « multimédia » dont on se sert presque tous !

La sécurité dans GNU/Linux, comme dans tout autre système, dépend de vos besoins, de vos moyens et des contraintes que vous serez prêt à supporter...

Lucie Robin





Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## C'est quoi un port IP ? Je bloque quoi et comment ?



### Transit conventionnel et transit personnalisé

Imaginez un téléphone. En fait, imaginez que l'on ait relié 65535 téléphones à votre ordinateur. C'est très pratique, car cela permet d'avoir 65535 conversations avec 65535 personnes différentes en même temps... Un port, sur une adresse IP, c'est un peu la même chose : disons pour simplifier les choses que c'est un canal de communication ; il y en a 65535 sur une adresse IP.

#### Quel port fait quoi ?

Supposons que je souhaite communiquer avec un serveur web. Normalement, tout est transparent pour l'utilisateur : je lance un navigateur web, puis, je vais sur mon moteur de recherche préféré. En fait, en coulisses, il se passe beaucoup de choses. Une fois mon navigateur lancé, mon système va chercher à qui correspond l'adresse IP du domaine **mon-moteur-de-recherche.com** ; une fois l'adresse IP trouvée (disons 123.456.78.9), mon navigateur va tenter d'établir une communication avec le serveur web. Pour cela, le navigateur utilise le port 80 de l'IP destinataire. Pourquoi le port 80 ? C'est une convention, un standard, on utilise le port 80 pour les communications HTTP, le port 443 pour les communications HTTPS, le port 25 pour le mail (SMTP), etc. Une fois la communication établie, les informations seront transmises. Le serveur web enverra le contenu de la page web au navigateur, qui l'interprétera pour faire apparaître la page **mon-moteur-de-recherche.com**.

#### Puis-je personnaliser mes ports ?

Il est parfaitement possible de lancer un serveur web sur un port autre que le port 80. Par exemple sur mon ordinateur personnel, si je crée un serveur web, je peux choisir n'importe quel port entre 1 et 65535. Le problème sera simplement d'indiquer au client web quel port il doit utiliser. On peut l'indiquer dans l'URL : au lieu de visiter **monsie.org**, il faudra taper **http://monsie.org:8080** si j'utilise le port 8080.

### Du bon usage du firewall et autres dispositifs bloquants

C'est un peu comme des portes : la plupart du temps, il est utile de les laisser ouverts, si on souhaite permettre l'établissement d'un trafic. Mais, parfois il peut être utile de les bloquer...

#### Est-ce que je laisse tout passer ?

Le firewall a cette fonction. Il permet, en définissant des règles, de réguler les communications établies ou reçues par un système informatique. Il existe différents types de firewalls : certains régulent le trafic des paquets IP directement, d'autres le régulent en observant le contenu de la communication.

Par exemple, Squid (**www.squid-cache.org**) est un proxy. Il permet entre autres de réguler les communications HTTP en fixant des règles : par exemple, telle adresse IP a le droit de contacter tel site, mais uniquement entre telle et telle heure ; ou encore, toute connexion à tel site est refusée, car son contenu est illicite.

La plupart des firewalls qu'on utilise aujourd'hui sont « de bas niveau ». Cela signifie qu'ils effectuent des contrôles directement sur la pile IP. Grâce à des règles, on peut contrôler le trafic entrant et sortant d'un système et s'assurer qu'il n'y a pas de trafic illicite (ex : que personne n'est en train de se connecter à votre système à votre insu). Presque toutes les distributions Linux sont aujourd'hui dotées de firewalls pré-configurés pour les utilisateurs non-initiés.

#### Alors je bloque quoi ?

La configuration d'un firewall dépend du niveau de contrainte que l'on est capable de s'imposer. L'idéal de sécurité serait d'interdire toute communication entrante et sortante, mais un tel système ne serait pas très utile pour quelqu'un souhaitant utiliser internet... Partant de ces besoins, on est donc obligé d'ouvrir certaines voies de communication : le web, le mail, le FTP... Cela revient à dire, si on verbalise les règles : j'autorise toute communication TCP partant de mon adresse vers n'importe quelle adresse, sur les ports 80, 25, 110, etc. Par contre, j'interdis tout autre trafic. Donc une personne qui tenterait d'initier une communication vers mon système sera bloquée par mon firewall.

Lucie Robin



Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>  
<http://www.unixgarden.com>  
<http://www.editions-diamond.fr>

## Comment tuer une application qui a planté, ou comment faire face à un écran « figé » ?

On vous avait pourtant dit que « Linux c'est super parce que ça plante jamais ! ». Oui, mais voilà, ça fait 5 minutes que vous êtes impuissant devant votre écran figé, que votre clavier ne répond plus et que vous secouez votre souris dans tous les sens... Pas de panique : il y a un moyen de remédier au problème, et ce, de façon beaucoup moins brutale que les boutons *Power* ou *Reset* de votre ordinateur !

Plusieurs cas de figure peuvent se présenter :

---> Si le programme planté a été lancé en ligne de commande, que vous avez encore accès au terminal de lancement et que votre clavier fonctionne, tapez simplement [Ctrl]+[C]. Vous récupérez la main, et toutes les fenêtres de l'application disparaîtront.

---> Si le programme a été lancé de manière graphique (via un menu ou un double-clic sur une icône), ouvrez un terminal et saisissez la commande : `~$ killall -9 nom_du_programme`.

---> Si la solution précédente ne fonctionne pas, vous pouvez remplacer le nom du programme par son PID (*Process Identifier*). Pour déterminer quel est le PID de l'application qui a planté, utilisez la commande :

```
~$ ps -ef | grep <motif>
```

Le motif correspond à une chaîne de caractères contenue dans le nom du programme incriminé. Par exemple, considérons que l'éditeur de Gnome, Gedit, ait planté, on entrera la commande :

```
~$ ps -ef | grep gedit
fleur  20665  1 7 15:51 ?        00:00:01 gedit
~$ kill -9 20665
~$
```

L'application se ferme et vous récupérez la main.

---> Si seule la souris vous a abandonné, n'oubliez pas que le raccourci clavier [Alt]+[F2] permet d'appeler le lanceur d'application. Vous pouvez saisir alors le nom de votre terminal de commandes (*gnome-terminal* sous Gnome ou *konsole* sous KDE), puis entrer une commande permettant de tuer l'application.

---> Si le serveur graphique X a planté et que votre clavier fonctionne toujours, il vous sera toutefois impossible de lancer ou d'utiliser une application graphique. Donc, impossible d'utiliser votre console graphique pour tuer l'application qui a provoqué le *freeze* de votre serveur graphique. Dans ce cas, il faut passer en mode texte *tty* (un mode texte absolu qui permet de communiquer directement avec le cœur de votre machine).

Pour cela, sous certaines distributions, un appui répété sur la touche [Windows] du clavier devrait faire l'affaire. Cette touche est parfois configurée pour permettre de passer de votre interface graphique aux différentes interfaces *tty*. Puis, après avoir saisi votre *login* et votre mot de passe, vous pourrez utiliser la commande *kill*, comme précisé ci-dessus.

Néanmoins, il se peut que la touche [Windows] ne soit pas configurée ainsi. Dans ce cas, les raccourcis [Ctrl]+[Alt]+[F1], [Ctrl]+[Alt]+[F2], [Ctrl]+[Alt]+[F3], [Ctrl]+[Alt]+[F4], [Ctrl]+[Alt]+[F5], etc. vous permettent de passer respectivement en mode *tty1*, *tty2*, *tty3*, etc. Pour revenir en mode graphique, utilisez [Alt]+[F7]. Ceci dépend en fait de ce qui est défini par défaut dans le fichier */etc/inittab* (lignes du type `1:2345:respawn:/sbin/getty 38400 tty1`, `2:23:respawn:/sbin/getty 38400 tty2`, etc.), mais c'est ce que l'on observe sur la plupart des distributions.

---> Si malgré l'arrêt des applications responsables, votre système est toujours figé, alors il ne vous reste plus qu'à saisir la commande suivante, en mode *tty*, commande qui permet de redémarrer « proprement » :

```
~$ shutdown -r now
```





Exemplaire offert par Les Éditions Diamond  
<http://www.ed-diamond.com>  
<http://www.unixgarden.com>  
<http://www.editions-diamond.fr>



---> Lorsque le problème est lié au serveur X, le plus simple consiste à redémarrer ce dernier. Pour ce faire, il suffit de taper au clavier la combinaison [Ctrl]+[Alt]+[Backspace]. Le serveur X va alors tuer toutes les applications graphiques et donc les sessions utilisateurs en cours, puis va se relancer. Vous revenez ainsi à l'écran de connexion.

---> Une autre solution consiste à utiliser une série de combinaisons clavier, basée sur la touche [Syst] ou [SysRq] (même touche que [Impr écran] ou [Print screen]), à taper dans un ordre bien défini :

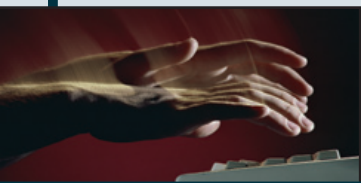
- 1 [Alt]+[Syst]+[R] : place le clavier en mode « brut » (*raw mode*). Puis, essayez de presser [Ctrl]+[Alt]+[Backspace] pour tuer le serveur X. Si ça ne fonctionne pas, continuez avec ce qui suit.
- 2 [Alt]+[Syst]+[S] : ceci permet d'écrire toutes les données non sauvegardées sur le disque (on parle de « synchronisation » du disque).
- 3 [Alt]+[Syst]+[E] : pour envoyer un signal de terminaison à tous les processus, sauf à *init*.
- 4 [Alt]+[Syst]+[I] : pour tuer tous les processus actifs, sauf *init*.
- 5 [Alt]+[Syst]+[U] : pour démonter, puis remonter toutes les partitions en lecture seule (ceci évitera une vérification du système de fichiers au redémarrage).
- 6 [Alt]+[Syst]+[B] : pour redémarrer le système. Vous pouvez aussi presser le bouton *reset* de votre machine.

Ces combinaisons de touches permettent d'envoyer des commandes directement au noyau, commandes qui permettront d'enregistrer les fichiers ouverts malgré l'absence d'interface graphique, puisque cette dernière est gelée.

Attention, pour que cela fonctionne, il faut que votre noyau ait été compilé avec le support des « clefs magiques » (l'option `CONFIG_MAGIC_SYSRQ` doit être à « y »), et que celui-ci soit activé dans `/proc` (ce qui est pratiquement toujours le cas dans les distributions courantes du type Ubuntu ou Mandriva). Pour le vérifier :

```
~$ grep CONFIG_MAGIC_SYSRQ /boot/config-2.6.15-27-386
CONFIG_MAGIC_SYSRQ=y
~$ cat /proc/sys/kernel/sysrq
1
```

Fleur Brosseau



## Pourquoi Linux utilise-t-il toute la mémoire vive alors que je n'ai quasiment rien lancé comme programme ? Pourquoi n'ai-je jamais de mémoire disponible ?

À quoi sert d'avoir de la mémoire qui ne sert à rien ? En fait, pour optimiser au maximum la vitesse du système, Linux utilise tout l'espace mémoire disponible pour accélérer les fonctionnalités dont le noyau a la charge.

L'élément le plus lent d'un ordinateur est son disque dur. Il n'y a pas de formule magique pour le faire tourner plus vite, mais on peut faire en sorte que les données déjà utilisées soient conservées en mémoire (ce qui éviterait une nouvelle recherche lente sur le disque dur). Linux gère ainsi de gros caches disque. Chaque accès au disque dur y est conservé dans le but d'accélérer une future demande.

Bien sûr, si une application demande de la mémoire, Linux réduit ces caches et fournit la mémoire nécessaire à l'application. Il s'ajuste automatiquement.

Pour connaître l'état de la mémoire, on tape la commande `free` dans une console. On voit ainsi la mémoire utilisée, avec des valeurs pour `buffers` et `cached` qui représentent la mémoire gérée par Linux pour accélérer des tâches qui lui sont propres. L'option `-m` permet d'avoir les résultats en mégaoctets.

```
~$ free -m
              total        used        free     shared    buffers     cached
Mem:           250          247           2          0           2          78
-/+ buffers/cache: 166          83
Swap:          486          152          333
```

Notez également que la commande `top` vous donne la consommation mémoire et de CPU de chaque tâche en cours d'exécution.

Nicolas Boulay

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Pourquoi certains fichiers bureautiques (traitement de texte, feuille de tableur, vidéo...) ne sont-ils pas lisibles (ou mal) avec les logiciels disponibles sous GNU/Linux ?

Ne vous est-il jamais arrivé de chercher à ouvrir sous GNU/Linux un fichier humoristique attaché à un courriel envoyé par un ami et de n'y parvenir qu'à moitié ? Et ce, en perdant au passage la petite musique de fond ou 80% de la mise en page ? Il est facile alors de se laisser emporter par la déception de l'instant. Mais non, le problème ne vient pas du fait que les développeurs de logiciels bureautiques libres n'ont pas d'humour et tentent d'en priver le reste de la planète ! Relativisons un peu...

Tout d'abord, vous n'avez pas perdu grand-chose ! Vous en avez de toute façon assez vu pour comprendre que même avec une musique de fond, la blague n'était pas à la hauteur des efforts consentis pour la voir, ni pour la transmettre d'ailleurs, ni même pour la fixer sur un support...

Bon, mais quand même, au nom de la liberté d'expression, je refuse de me contenter de fichiers à moitié décryptés ! Eh bien oui, on s'approche là, si vous n'avez pu récupérer que des bribes de fichier, c'est bien parce que vous êtes en train d'essayer d'ouvrir un fichier d'un format propriétaire, fermé, tenu secret et crypté, appartenant à une firme qui ne souhaite pas que ses fichiers puissent être ouverts à partir d'autres applications que celles qu'elle vend. Alors, c'est déjà pas si mal de pouvoir extraire d'un de ces fichiers la substance du message qu'il contient à partir d'un Logiciel libre ! Vous profitez là d'un long et laborieux travail de rétro-conception, que des lois comme DADSVI rendent désormais illégal et passible de prison...

Utiliser des fichiers au format propriétaire s'avère donc hasardeux, voire de mauvais goût lorsqu'il s'agit de partager des informations avec d'autres utilisateurs. Et si l'on pousse le raisonnement un peu plus loin, on se rend compte alors qu'utiliser des fichiers au format fermé se révèle même dangereux pour vos données. En effet, si vous sauvegardez de grandes quantités de travail dans un format que seul le vendeur de vos logiciels peut lire, il y a fort à parier que dans cinq ans, quand toutes les machines et les logiciels de travail auront changé, vous vous retrouveriez dans l'incapacité de consulter encore vos fichiers... Surtout si l'entité qui vous fournissait (de gré, et parfois de force) en logiciels a fermé boutique entre-temps, en emportant ses secrets et ses brevets ! Et dans le meilleur des cas, si elle survit, au moins se sera-t-elle assurée de votre infaillible fidélité durant toutes ces années...

Simon Descarpentries

## Pourquoi certains périphériques ne fonctionnent-ils pas (ou mal) sous Linux ?

Vous venez d'acheter la dernière merveille en vente chez votre spécialiste ou grande surface favorite et voilà que vous découvrez que ça ne marche pas (ou mal, c'est-à-dire que certaines fonctions ne sont pas disponibles) sur votre système GNU/Linux ! Et il vous faut passer outre les sarcasmes de certains qui vous expliquent que vous feriez mieux de changer d'outils...

Premièrement, êtes-vous allé voir sur la Toile s'il n'y a pas un site qui peut vous renseigner sur le support ? Pour une certaine imprimante, je me suis retrouvé sur un site en japonais pour disposer des pilotes (par ailleurs excellents). Alors, ne désespérez pas et commencez par faire un peu de lecture. Maintenant, je vous invite à me suivre afin de comprendre le pourquoi de cette situation (le périphérique est livré avec un CD, mais ce dernier est quasiment vide et entièrement pour certains systèmes propriétaires).

En fait, le support des périphériques sous Linux a toujours été laborieux. On peut au passage remercier des sociétés comme HP qui auront fait beaucoup pour faciliter ledit support. Mais le fait est que, souvent, le support des périphériques se heurte à plusieurs problèmes rédhibitoires :

---> L'absence de volonté du constructeur du périphérique. C'est une équipe de volontaires, non liée au constructeur, qui avec les moyens du bord et un travail patient d'ingénierie, écrit peu à peu les pilotes pour le périphérique. Le constructeur ne s'y oppose pas, mais ne fait rien pour aider. Alors, on dispose des fonctions fondamentales, mais forcément le reste, c'est pour plus tard...





Exemplaire offert par Les Éditions Diamond  
<http://www.ed-diamond.com>  
<http://www.unixgarden.com>  
<http://www.editions-diamond.fr>



---> L'obstruction du constructeur qui bride certaines fonctions ou interdit le développement. Il arrive que le constructeur n'autorise pas le développement de tout ou partie d'un pilote en GPL, car il considère que ce dernier pourrait violer ses secrets professionnels. Cependant, dans bien des cas, il ne fournit aucun pilote en échange. L'utilisateur qui dispose d'autre chose que d'un système d'exploitation propriétaire est prié de faire sans ! Une variante est la fourniture de pilotes fermés et non maintenus (bonjour la galère pour trouver la bonne version de noyau, la bonne version des outils annexes, etc.). En outre, la documentation d'un tel objet est plus que succincte...

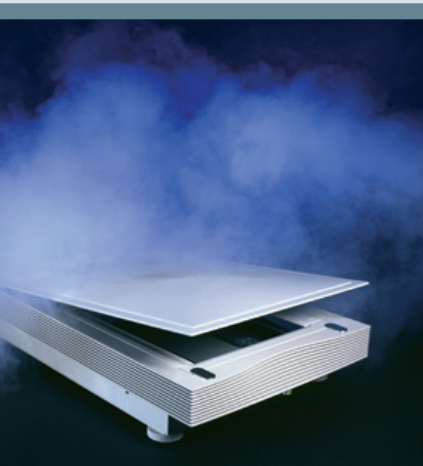
---> Le soutien du bout des lèvres. Le constructeur veut bien aider, mais il ne faut pas que ça lui coûte ! Alors, il fournit un peu de documentation, oriente les développeurs, bride souvent (pas de secrets industriels !) et on obtient un pilote moyen, mais qui fait presque tout.

Toutes ces situations ne valent que s'il y a des volontaires. Or, dans certains cas, le périphérique est trop confidentiel, trop vieux ou simplement n'a intéressé personne disposant des compétences requises. Ce qui peut paraître ironique, c'est que les constructeurs développent eux-mêmes leurs pilotes pour les systèmes propriétaires et ils y investissent beaucoup d'argent. La seule exception, c'est le Logiciel libre. Là, ce sont les utilisateurs qui investissent ! Et forcément, tout le monde n'est pas logé à la même enseigne !

Sauf que certains industriels ont fini par comprendre que cela n'était finalement pas une bonne idée. Pourquoi ? Un périphérique qui fonctionne mal est un très mauvais ambassadeur pour la marque et la société. Or, les utilisateurs de Linux et des Logiciels libres ne sont plus des cas isolés. Ils sont dans les bureaux du constructeur et certains aimeraient bien voir le support amélioré ou simplement disponible ! Les restrictions à ce support sont alors clairement inverses aux devoirs que les constructeurs ont acceptés de la part des éditeurs de systèmes propriétaires. Ici, pas de support direct, sauf exception. Par contre, les pilotes sont développés souvent très bien et sont excellents en termes de stabilité. La documentation est aux normes industrielles. Le constructeur met les moyens à la hauteur de sa vision du marché Linux. On sent encore qu'il y a une différence de traitement, mais cela devient marginal lorsque le constructeur fait ce qu'il faut pour être dans le noyau officiel ou simplement respecter les règles des communautés avec lesquelles il doit cohabiter.

Alors, comment faire pour ne pas avoir de soucis ? Eh bien, avant d'acheter, il faut se renseigner. En fonction de l'achat, il existe pour pratiquement tous les périphériques des sites qui vous expliqueront si telle ou telle technologie, périphérique, etc. fonctionne et comment sous GNU/Linux ou d'autres systèmes libres. Certes, cela tue « l'achat passion ». Mais l'achat passion, c'est aussi le risque de ne pas être supporté... A moins que le constructeur ait clairement indiqué que le produit est compatible Linux ou OS libres. Cela arrive de plus en plus, en particulier pour les périphériques réseau et les imprimantes. Comme quoi le lobbying, ça paie ! Et puis, disons le franchement, il y a des marques plus « amicales » que d'autres, à vous de vous renseigner...

Sidoine Pierrel



## Un programme que j'ai lancé s'est terminé sur un « segmentation fault ». Qu'est-ce que c'est ?



Aux débuts de l'informatique, les ordinateurs n'étaient capables d'exécuter qu'un programme à la fois. L'intégralité des ressources – notamment la mémoire – était allouée à une seule tâche.

Puis sont arrivés des systèmes d'exploitation permettant d'exécuter plusieurs tâches, chacune d'entre elles disposant d'une zone de mémoire dans laquelle elle peut travailler. Or, malgré cette délimitation virtuelle, chaque tâche pouvait, sans aucun contrôle, aller écrire où bon lui semble dans la mémoire de la machine ! Et ainsi aller perturber le travail des autres

Exemplaire offert par Les Éditions Diamond

<http://www.ed-diamond.com>

<http://www.unixgarden.com>

<http://www.editions-diamond.fr>

## Je n'arrive pas à booter sur le CD, comment faire ?

Lorsque vous testez une distribution *live* ou, tout bonnement, lorsque vous souhaitez installer une distribution à partir du CD d'installation, il faut absolument que votre système démarre via le CD-ROM qui se trouve dans le lecteur. Mais voilà, rien à faire, votre système démarre à chaque fois sur votre disque dur !

Au démarrage de votre ordinateur, un programme simple enregistré sur la carte mère, appelé BIOS, se charge d'initialiser votre ordinateur, puis de donner la main à un chargeur d'amorçage (ex. : Grub ou Lilo) qui s'occupe de charger un système d'exploitation. Le BIOS passe ainsi en revue le disque dur, le lecteur de CD-ROM, le lecteur de disquette, etc. jusqu'à trouver le programme de démarrage (le programme de *boot*). Mais pas toujours dans cet ordre ! C'est à vous, utilisateur, de fixer l'ordre de passage en revue des différents périphériques de démarrage, grâce au paramètre du BIOS appelé *boot order* (l'ordre d'amorçage). Comprenez que, dans le cas présent, le lecteur de CD-ROM doit être vérifié avant tout le reste. Donc, il vous suffit de modifier le paramétrage du BIOS de votre machine de façon à placer le lecteur de CD en première position.

Lorsque vous allumez votre ordinateur, le BIOS se lance. Il effectue certaines vérifications que vous pouvez d'ailleurs suivre à l'écran : fréquence du processeur, test de la mémoire vive, etc. Vous pouvez généralement entrer dans la configuration du BIOS dans les premières secondes qui suivent la mise sous tension de l'ordinateur en pressant une touche ou une combinaison de touches indiquée à l'écran. (Ex. : *DEL* -> *Setup*). Pressez alors rapidement la touche [Suppr] (cas le plus fréquent, mais il peut s'agir d'une autre touche).

Vous vous retrouvez alors face à l'écran de configuration de votre BIOS. Cherchez la section qui permet d'accéder aux différents paramètres (ex. : *Advanced BIOS Features*), puis recherchez le boot order (ex. : *Boot Device Select*). Vous disposez de trois lignes (1<sup>st</sup>, 2<sup>nd</sup>, 3<sup>rd</sup>) vous permettant d'organiser les différents périphériques de démarrage ; c'est ici que votre lecteur de CD/DVD doit se trouver en première position. Notons que le boot order des ordinateurs récents est en général correctement configuré : 1) CD-ROM 2) disque dur 3) disquette.

Lorsque cela est fait, enregistrez vos modifications, puis quittez l'interface de configuration du BIOS. Votre ordinateur redémarre et si votre CD se trouve dans le lecteur, vous devriez enfin parvenir à booter sur ce dernier...

Fleur Brosseau



tâches, voire du système lui-même. Une telle possibilité est inadmissible, car posant de lourds problèmes de sécurité et de confidentialité, chaque tâche pouvant à volonté faire planter le système tout entier ou espionner ce que font les autres.

Il fallait donc trouver une solution à ce problème. Elle s'avérera être matérielle. À partir des années 1985/1986, les nouveaux processeurs ont intégré un circuit supplémentaire : l'unité de gestion mémoire (*Memory Management Unit* ou MMU).

Cette unité est programmée par le système d'exploitation de telle sorte que chacun des programmes qui s'y exécute se voit assigner une zone mémoire précise. La MMU peut alors contrôler chacun des accès mémoire effectués et vérifier leur validité. Si un programme a le malheur de vouloir accéder à une zone mémoire qui ne lui appartient pas, la MMU envoie un signal au système d'exploitation qui peut alors décider de tuer le programme fautif afin de préserver la confidentialité ou la sécurité du système : le programme reçoit alors un signal *SIGSEGV* et est tué. C'est la fameuse « *segmentation fault* » ou erreur de segmentation : le programme a violé la segmentation mémoire et est terminé pour des raisons de sécurité.

Cette erreur peut être le résultat d'un code malicieux, mais, dans la grande majorité des cas, il s'agit d'une erreur de programmation venant soit du programme lui-même, soit d'une des bibliothèques qu'il utilise : typiquement, le programmeur a voulu se servir d'un pointeur non-initialisé qui, par infortune, pointe vers une zone qu'il n'est pas autorisé à accéder.

Alexandre Courbot





Exemplaire offert par Les Éditions Diamond  
<http://www.ed-diamond.com>  
<http://www.unixgarden.com>  
<http://www.editions-diamond.fr>



## Êtes-vous incollable sur le système GNU/Linux ?



( Réponses au quiz page 31 )

**1. FAUX** : Linux a vu le jour en 1991, mais Unix est né bien avant, en 1969 (p. 20). **2. VRAI** et **FAUX** : certes, R. Stallman est l'initiateur du projet GNU et de la licence GPL, mais n'oublions pas que la création du noyau Linux revient à Linus Torvalds (p. 13). **3. FAUX** : Tux est le nom de la mascotte du système GNU/Linux (p. 19). **4. FAUX** : les systèmes GNU/Linux sont aujourd'hui tout à fait accessibles pour l'utilisateur lambda, d'autant plus que toute la configuration et l'administration du système peuvent être effectuées via des utilitaires graphiques. En outre, les distributions les plus répandues (Ubuntu, Mandriva, etc.) offrent aujourd'hui une procédure d'installation automatisée, qui devrait convaincre les plus réticents... **5. FAUX** : un *live-CD* comporte une distribution GNU/Linux autonome, que vous pouvez découvrir et tester en toute liberté, sans installer quoi que ce soit sur votre disque dur et aussi souvent que vous le souhaitez (p. 35). **6. FAUX** : Windows et Linux peuvent tout à fait cohabiter sur un même disque dur, du moment que chacun dispose de la place qu'il lui faut et que vous veillez à ne pas écraser la partition dédiée à Windows lors de votre installation (p. 60). **7. FAUX** : GNU/Linux est justement le système d'exploitation idéal pour les configurations matérielles les plus modestes (p. 24). **8. FAUX** : peu de constructeurs pensent à fournir les pilotes d'installation pour les systèmes GNU/Linux (p. 25 et 77). **9. FAUX** : on en parle peu (voire pas) dans les médias, on ne l'utilise pas par défaut dans la majorité des entreprises ni à l'école, mais GNU/Linux fait tranquillement son chemin et gagne, mine de rien, de plus en plus d'adeptes, et ce, dans le monde entier. Les TPE/PME, les universités, les administrations publiques, tous y voient surtout une formidable économie et pour les mêmes performances ! (p. 7 et 14). **10. VRAI** et **FAUX** : c'est fortement recommandé pour faciliter installations et mises à jour, mais ce n'est pas obligatoire. **11. FAUX** : techniquement parlant, il existe des virus pour les systèmes GNU/Linux, mais ils sont simplement beaucoup plus rares que ceux destinés aux autres systèmes d'exploitation (p. 73). **12. VRAI** et **FAUX** : de par leur conception, les systèmes de fichiers de Linux ont une tendance naturelle à n'écrire que là où il y aura la place de le faire et d'optimiser ainsi l'espace disque consommé. Cependant, lorsque le disque commence à être plein et que de gros volumes de données sont manipulés, le système n'aura plus beaucoup le choix et on pourra observer une fragmentation des fichiers (p. 26). **13. FAUX** : un logiciel est dit « libre » lorsqu'il est conforme à tous les critères définis par la licence libre qui le régit (p. 9). **14. FAUX** : le terme « libre » ne signifie pas forcément « gratuit » (p. 5 et 10). **15. FAUX** : même si les Logiciels libres bénéficient d'une meilleure réputation à ce sujet que leurs consorts propriétaires, il peut arriver qu'une application plante (bug d'une version instable d'un logiciel, saturation du processeur, manque de mémoire, etc.) (p. 75). **16. VRAI** : une multitude de sites Web recensent tous les Logiciels libres existants, classés par domaine d'applications, vous n'avez qu'à faire votre choix et à l'installer en quelques clics de souris (p. 17 et 52). **17. FAUX** : une donation est en effet toujours la bienvenue chez les développeurs de Logiciels libres et gratuits, mais elle n'est en rien obligatoire ! **18. VRAI** : selon les termes de la licence GPL, un logiciel régi par celle-ci peut être exécuté, copié, modifié et distribué librement (p. 9). **19. VRAI** (p. 39). **20. FAUX** : tous les développeurs font en sorte de respecter la norme LSB (p. 33). **21. FAUX** : pour pouvoir utiliser les applications de l'environnement que vous n'avez pas, il vous suffira d'installer conjointement les bibliothèques de base (mais ceci est entièrement pris en charge par votre gestionnaire de paquets de logiciels) (p. 23). **22. VRAI** (p. 58). **23. VRAI** et **FAUX** : même si la plupart des logiciels propriétaires ont un équivalent libre, ce n'est pas encore une généralité (p. 62). **24. FAUX** : il vous faudra simplement être vigilant vis-à-vis des formats de vos fichiers audio et vidéo (p. 67 et 69). **25. FAUX** : il existe une multitude de jeux libres pour votre système Linux, de toute catégorie : arcade, stratégie, aventure, simulation, *shoot em up*, RPG, etc. Deux adresses incontournables : <http://jeuxlibres.net/> [fr] et <http://www.happypenguin.org/> [en]. **26. VRAI** : diverses applications (des programmes d'émulation) vous permettent de « faire croire » à un logiciel qu'il est exécuté sur un système Windows (p. 61). **27. FAUX** : un troll désigne un sujet de conversation qui amène assurément à la discorde les participants à la discussion (p. 31). **28. FAUX** : un démon désigne un programme qui tourne en arrière-plan de votre système, chargé d'exécuter une tâche prédéterminée (p. 42). **29. VRAI** : ce codage permet de représenter les milliers de caractères d'Unicode, incluant tous les caractères de n'importe quel système d'écriture (p. 41). **30. FAUX** : au contraire, l'utilisation intempestive du compte root peut être dangereuse pour votre système. L'utilisateur root ayant tous les droits d'administration, une mauvaise manipulation de votre part serait fatale et irréversible pour votre système (p. 70).

# **ABONNEZ-VOUS ET RETROUVEZ L'ENSEMBLE DE NOS PUBLICATIONS**



**CLIQUEZ ICI !**

**[www.ed-diamond.com](http://www.ed-diamond.com)**

**SUR NOTRE  
SITE MARCHAND**



(Re)Découvrez

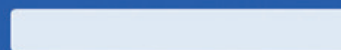
# UnixGarden v3

Le site éditorial des Éditions Diamond

GNU/Linux

Mac OS X

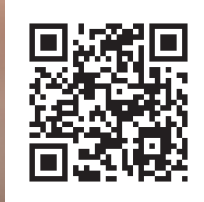
+BSD



Rechercher

Recherche avancée

## Retrouvez une sélection d'articles publiés par Les Éditions Diamond !



# www.unixgarden.com

LE RENDEZ-VOUS DE TOUS LES INTERNAUTES AVIDES DE CONNAISSANCES TECHNIQUES CONCERNANT L'OPEN SOURCE !