

Kiki Novak

# Administration LINUX par la pratique

Tome 1 : les fondamentaux  
de l'administration système



● Éditions  
EYROLLES

# Administration LINUX par la pratique

Tome 1

## Une introduction aux bases de l'administration Linux

Ce guide pratique et ludique répond à toutes les questions que vous vous posez sur les bases de l'administration Linux, en explorant les distributions GNU/Linux les plus populaires, notamment CentOS 7 et la famille Red Hat Enterprise Linux.

Il vous permet d'installer, de configurer et de gérer des serveurs sous Linux. Conçu étape par étape, il suit une progression pédagogique cohérente, en détaillant tout ce qu'il faut savoir sur les bases de l'administration système Linux (shell et ligne de commande, droits d'accès, processus, etc.).

## 100 % pratique et compatible RHEL/CentOS

Les configurations fournies en exemple sont immédiatement utilisables, telles quelles. Aucun prérequis externe n'est nécessaire puisque les techniques enseignées permettent à un administrateur novice de RHEL/CentOS de maîtriser les bases du système, tout en lui montrant les bons réflexes pour se documenter et apprendre en autonomie par la suite.

## À qui s'adresse cet ouvrage ?

- Aux administrateurs de serveurs qui souhaitent maîtriser leurs systèmes Linux au quotidien.
- Aux adeptes de Linux désireux d'approfondir leurs connaissances.
- Aux professionnels qui préparent la certification RHCSA ou RHCE.

## Au sommaire

Si Linux m'était conté • Avant de mettre la main à la pâte • Linux installé par une poule • Linux en mode texte : consolez-vous ! • Gérer les utilisateurs • Gérer les droits d'accès • Chercher une aiguille dans une botte de foin • Créer et manipuler des liens • Gérer les processus • Gérer les services • Accéder aux périphériques amovibles • Partitionner et formater un disque dur • Gérer les archives compressées • Premiers pas sur le réseau • Installer et gérer des logiciels • Manipuler le chargeur de démarrage et le noyau • Installer et gérer un RAID logiciel • Gérer les volumes logiques avec LVM • Gérer les logs du système • Annexe : installer un poste de travail Linux

Féru d'informatique depuis les jours du processeur monoplatine 8080 et du Commodore VC-20, **Kiki Novak** est gérant de la société Microlinux, spécialisée dans l'élaboration de solutions informatiques durables reposant à 100 % sur GNU/Linux et les logiciels libres. Microlinux est également un organisme de formation s'adressant aux utilisateurs et aux administrateurs de Linux et du libre. Aucun stagiaire n'a été maltraité durant la conception de cet ouvrage. <https://www.microlinux.fr>

[www.editions-eyrolles.com](http://www.editions-eyrolles.com)

# **Administration Linux par la pratique**

DANS LA MÊME COLLECTION

S. RINGUEDÉ. – **SAS.**

N°67631, 4<sup>e</sup> édition, 2019, 688 pages.

C. BLAESS. – **Solutions temps réel sous Linux.**

N°67711, 3<sup>e</sup> édition, 2019, 318 pages.

T. PARISOT. – **Node.js.**

N°13993, 2018, 472 pages.

C. PIERRE DE GEYER, J. PAULI, P. MARTIN, E. DASPET. – **PHP 7 avancé.**

N°67720, 2<sup>e</sup> édition, 2018, 736 pages.

H. WICKHAM, G. GROLEMUND. – **R pour les data sciences.**

N°67571, 2018, 496 pages.

F. PROVOST, T. FAWCETT. – **Data science pour l'entreprise.**

N°67570, 2018, 370 pages.

J. CHOKOGOUE. – **Maîtrisez l'utilisation des technologies Hadoop.**

N°67478, 2018, 432 pages.

R. GOETTER. – **CSS 3 Flexbox.**

N°14363, 2016, 152 pages.

H. BEN REBAH, B. MARIAT. – **API HTML 5 : maîtrisez le web moderne !**

N°67554, 2018, 294 pages.

W. MCKINNEY. – **Analyse de données en Python.**

N°14109, 2015, 488 pages.

E. BIERNAT, M. LUTZ. – **Data science : fondamentaux et études de cas.**

N°14243, 2015, 312 pages.

SUR LE MÊME THÈME

K. NOVAK. – **Débuter avec Linux.**

N°13793, 2017, 522 pages.

R. HERTZOG, R. MAS. – **Debian Jessie.**

N°14203, 2016, 538 pages.

J.-F. BOUCHAUDY. – **Linux Administration – Tome 1.**

N°14082, 2014, 690 pages.

Retrouvez nos bundles (livres papier + e-book) et livres numériques sur  
<http://izibook.eyrolles.com>

Kiki Novak

# Administration Linux par la pratique

Tome 1 : les fondamentaux de l'administration système

● Éditions  
EYROLLES

ÉDITIONS EYROLLES  
61, bd Saint-Germain  
75240 Paris Cedex 05  
[www.editions-eyrolles.com](http://www.editions-eyrolles.com)

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans l'autorisation de l'Éditeur ou du Centre Français d'exploitation du droit de copie, 20, rue des Grands Augustins, 75006 Paris.

© Éditions Eyrolles, 2019, ISBN : 978-2-212-67738-6

# Avant-propos

---

Vous voilà prêt à acheter ce livre, ou du moins suffisamment curieux pour jeter un œil aux premières lignes de l'avant-propos. Cet ouvrage est un guide pratique pour les administrateurs en herbe qui veulent installer, configurer et gérer des serveurs sous CentOS et Red Hat Enterprise Linux.

Si vous pensiez vaguement que Linux est une marque de mouchoirs ou de lessive, ce livre s'adresse également à vous. Aucun savoir, aucune compétence spécifique n'est présumée de votre côté. Si vous êtes doté d'une dose saine de curiosité et d'un certain plaisir à expérimenter, considérez-vous comme le lecteur idéal de ces pages.

**Figure 1**

Linux, ce n'est pas seulement une marque de lessive. C'est aussi le système qui fait fonctionner les serveurs de la planète entière.



*Administration Linux par la pratique* est un véritable manuel de formation, élaboré et peaufiné dans le cadre des cours d'introduction à l'administration des systèmes Linux que je dispense entre Nîmes, Alès et Montpellier. Il est le fruit de quelques années de terrain, qui a mûri lentement sous le soleil de la garrigue gardoise, à la lumière des nombreuses questions de mes stagiaires.

Certes, il existe déjà plusieurs ouvrages sur Linux en général et l'administration des serveurs CentOS et Red Hat Enterprise Linux en particulier. Il est même réjouissant de voir que les publications sur ce sujet se multiplient. En quoi ce livre est-il différent, alors ? Plutôt que réinventer la roue, j'ai choisi une approche pragmatique : **vous faire découvrir les bases de Linux en plongeant les mains dans le cambouis, sans prérequis externes, en suivant une progression pédagogique cohérente.**

Ce premier ouvrage de la collection se limite à l'administration de base : l'installation, les bases du shell et de la ligne de commande, la gestion des utilisateurs, les droits d'accès, les processus, la gestion des disques et des applications, etc. Les aspects avancés de la configuration – le serveur web Apache, le serveur de messagerie Postfix, le serveur de fichiers Samba, etc. – feront l'objet de tomes ultérieurs. Songez à la célèbre devise du grand philosophe Friedrich Nietzsche, selon laquelle celui qui un jour veut apprendre à voler doit d'abord apprendre à se tenir debout et à marcher. Le tome 1 que vous tenez entre les mains est donc entièrement dédié à la marche ; la danse et le vol plané seront abordés à partir du deuxième tome.

Toutes les configurations fournies en exemple sont immédiatement utilisables, telles quelles. Elles sont basées sur CentOS 7, une distribution dérivée de Red Hat Enterprise Linux 7. D'après mon expérience, les formations qui essaient de danser à toutes les noces (comme on dit dans mon pays natal) finissent par embrouiller les novices. En règle générale, je conseille à mes stagiaires de s'initier à Linux (ou Unix) en optant pour un seul système – CentOS en l'occurrence – et de découvrir les autres, comme Debian, Ubuntu ou FreeBSD, par la suite.

CentOS est probablement le système Linux le plus ennuyeux qui existe. En informatique, « ennuyeux » est une qualité très recherchée. Pas de drame, pas de mauvaises surprises, juste un système qui fonctionne bien au quotidien. La distribution CentOS n'inclut pas les technologies dernier cri tout juste aptes à faire exploser vos systèmes en plein vol. Au lieu de cela, l'ambition de cette distribution – et de sa jumelle commerciale Red Hat Enterprise Linux – consiste à fournir le système *open source* le plus stable et le plus fiable qui existe, et ce pour une durée de dix ans pour chaque version. Il n'est donc pas étonnant que ces deux distributions soient omniprésentes dans les entreprises et les *datacenters* du monde entier.

Un effet secondaire non négligeable du focus de ce livre, c'est que les administrateurs un peu aguerris pourront l'utiliser pour préparer leur certification RHCE (*Red Hat Certified Engineer*) ou RHCSA (*Red Hat Certified System Administrator*). Les ouvrages spécialisés dédiés à ce sujet (on ne citera pas de noms) sont particulièrement soporifiques et ce n'est peut-être pas une mauvaise idée de les compléter par une approche plus ludique.

*No profit grows where is no pleasure taken*, disait Shakespeare. Pour vous donner une idée de l'ambiance générale qui peut régner dans les formations Linux que j'ai l'occasion d'animer, vous pouvez imaginer une bande de grands enfants qui s'amusent avec l'équivalent numérique d'une boîte de Lego, vous n'en serez pas très loin. J'ai fait de mon mieux pour essayer de conserver cet état d'esprit dans la rédaction de ce livre. Et si jamais vous trouvez que tout cela



ne fait pas très sérieux, rappelez-vous cet autre adage de Nietzsche selon lequel il n'y a rien de plus sérieux qu'un élève qui joue.

## Que contient ce livre ?

Ce livre est conçu pour être lu dans l'ordre, page par page, chapitre par chapitre. Lisez-le comme un roman, tout en sachant que le chapitre sur lequel vous ferez l'impasse reviendra vous mordre les fesses.

Le **chapitre 1** raconte l'histoire d'Unix, de GNU et de Linux depuis les origines. Il effectue un tour d'horizon sur l'informatique en général et Linux en particulier. Si vous ne savez pas trop ce qu'est un « système d'exploitation », une « distribution » ou un « logiciel libre », ce chapitre fournit des réponses à vos questions.

Le **chapitre 2** présente CentOS, le système (ou la distribution) utilisé pour mettre en pratique les exemples de ce livre, ainsi que des considérations sur le choix du matériel informatique sur lequel vous pourrez l'installer. Nous y présentons également VirtualBox, un logiciel libre de virtualisation qui vous permettra de vous entraîner sur votre ordinateur personnel sans pour autant mettre en péril votre système et vos données.

Dans le **chapitre 3**, vous pourrez vous mettre aux fourneaux en apprenant comment effectuer l'installation standard d'un système Linux, de la manière la plus simple possible. Le but de l'opération est que vous disposiez d'un système fonctionnel, qui vous permette de découvrir Linux par la pratique, chez vous, sur votre ordinateur.

Les **chapitres 4 à 19** constituent sans aucun doute le « plat de résistance » de l'ouvrage. Ils vous initient progressivement à l'administration d'un système Linux en commençant par des opérations de base (**chapitre 4**), pour vous amener petit à petit à réaliser des opérations plus avancées comme la gestion des utilisateurs (**chapitre 5**) et des droits d'accès (**chapitre 6**), la recherche de fichiers (**chapitre 7**), la création et la manipulation de liens (**chapitre 8**), la gestion des processus (**chapitre 9**) et des services (**chapitre 10**), l'accès aux périphériques (**chapitre 11**), le partitionnement et le formatage des disques (**chapitre 12**) ou encore la gestion des archives compressées (**chapitre 13**). Le **chapitre 14** présente les bases du réseau sous Linux. La gestion des logiciels est abordée au **chapitre 15**. Le **chapitre 16** nous plonge au cœur du chargeur de démarrage et nous fait changer de kernel comme de chemise. Les **chapitres 17 et 18** présentent des concepts avancés pour la gestion des disques comme le RAID et LVM. Enfin, le **chapitre 19** vous montre tout ce qu'il faut savoir pour exploiter les logs du système.

Et, s'il vous reste un petit creux, vous trouverez en **annexe** des informations supplémentaires pour l'installation d'un poste de travail basé sur la distribution OpenSUSE Leap et l'environnement de bureau KDE.

## Remerciements

Merci à Alexandre Habian pour avoir cru en ce projet et surtout pour sa patience. Sans son professionnalisme – le vrai, celui qui ne fait pas l’impasse sur les qualités relationnelles et la bonne humeur – ce livre n’aurait jamais vu le jour.

Un grand merci à Clothilde, ma tendre moitié, pour son soutien, ses bons conseils, ses bons petits plats et les hectolitres de thé au citron.

### À propos de l’auteur

Féru d’informatique depuis les jours du processeur 8080 et du Commodore VC-20, Kiki Novak est gérant de la société Microlinux, spécialisée dans l’élaboration de solutions informatiques durables basées à cent pour cent sur Linux et les logiciels libres. Microlinux est également un organisme de formation s’adressant aux utilisateurs et aux administrateurs de Linux.

► <https://www.microlinux.fr>

# Table des matières

---

## CHAPITRE 1

<b>Si Linux m'était conté</b> .....	<b>1</b>
Au commencement était Unix . . . . .	1
Richard Stallman et le projet GNU . . . . .	5
Linus Torvalds et le noyau Linux. . . . .	9
Les ancêtres Slackware, Red Hat et Debian . . . . .	14
Slackware Linux. . . . .	15
Les distributions de l'entreprise Red Hat . . . . .	16
Debian GNU/Linux . . . . .	17
Linux pour tous : la famille Ubuntu . . . . .	18
SUSE : un drôle de caméléon. . . . .	20
Les systèmes BSD, une autre famille d'Unix libres. . . . .	21
La foire aux distributions . . . . .	22
Quelle est la « meilleure » distribution Linux ? . . . . .	23
La panoplie logicielle . . . . .	23
L'actualité des paquets . . . . .	24
Les outils d'administration . . . . .	25
Le caractère commercial . . . . .	26
La qualité « entreprise » . . . . .	26
La configuration par défaut . . . . .	28
La qualité de la documentation . . . . .	28
Qui utilise Linux ? . . . . .	29

## CHAPITRE 2

<b>Avant de mettre la main à la pâte</b> .....	<b>31</b>
Petite introduction culinaire . . . . .	31
Se former à Linux avec CentOS. . . . .	32
Le matériel : usine à gaz ou simple gazinière ? . . . . .	33
Acheter un PC faisant office de serveur . . . . .	33
Acheter un vrai serveur . . . . .	34
Comment obtenir la distribution CentOS ? . . . . .	36
Graver le fichier ISO . . . . .	37
Et si je n'ai pas de lecteur optique ? . . . . .	38
S'entraîner sur un système virtualisé. . . . .	39
Créer et configurer une machine virtuelle . . . . .	40

## CHAPITRE 3

**Linux installé par une poule ..... 45**

L'œuf ou la poule ?	45
Au commencement était le BIOS	46
Testez votre mémoire !	48
Installation de CentOS	49
Premiers pas	49
Choisir la disposition du clavier	51
Partitionner le disque dur	52
Désactiver le service Kdump	54
Activer le réseau et définir le nom d'hôte	54
Démarrer l'installation	56
Définir les paramètres utilisateur	56
Créer un utilisateur	58
Fin de l'installation et redémarrage initial	59

## CHAPITRE 4

**Linux en mode texte : consolez-vous ! ..... 61**

Introduction à la ligne de commande	61
Faut-il avoir peur du mode texte ?	61
Console ? Terminal ? Et quoi encore ?	63
Se connecter à un serveur Linux en SSH	63
<i>Utiliser PuTTY sous Windows</i>	64
<i>Utiliser OpenSSH sous Windows</i>	65
<i>Utiliser OpenSSH sous Mac OS X</i>	66
<i>Utiliser OpenSSH sous Linux ou BSD</i>	67
Basculer entre les consoles virtuelles	67
Quitter la console	69
Premiers pas en ligne de commande	69
Navigation : ls, pwd et cd	70
Afficher le contenu d'un répertoire avec ls	70
Décrypter les résultats de votre ordinateur	71
<i>Qu'est-ce qui est quoi là-dedans ?</i>	71
<i>Mais encore ?</i>	72
<i>Humain, pas trop humain ?</i>	73
<i>Splendeur et misère des fichiers cachés</i>	73
<i>Cachez cette configuration que je ne saurais voir</i>	74
Afficher les informations détaillées d'un répertoire	74
pwd : « Vous êtes ici ! »	75
On bouge avec cd !	75
<i>Chemin relatif ou absolu ?</i>	75
<i>À court d'arguments</i>	76
« Ici » et « à l'étage »	76
Deux commandes de sortie simples : echo et cat	79
echo : afficher une ligne de texte	79
cat : afficher et concaténer	80
La structure des répertoires sous Linux	82
Où suis-je ?	82
Une structure en arborescence	82

Visite guidée du système en dix minutes . . . . .	83
<i>Home sweet home</i> . . . . .	83
<i>Remonter à la racine : /</i> . . . . .	83
<i>Les répertoires /bin et /boot</i> . . . . .	83
<i>Les répertoires /dev et /etc</i> . . . . .	84
<i>Le répertoire /lib</i> . . . . .	85
<i>Les répertoires /mnt, /media et /run</i> . . . . .	86
<i>Les répertoires /proc et /sys</i> . . . . .	86
<i>Les répertoires /root et /sbin</i> . . . . .	87
<i>Le répertoire /usr</i> . . . . .	88
<i>Les répertoires /tmp et /var</i> . . . . .	89
<b>Visualiser : more et less</b> . . . . .	89
Gérer l’affichage de fichiers longs . . . . .	89
Visualiser avec more . . . . .	90
Less is more : moins, c’est plus ! . . . . .	91
<b>Créer : touch et mkdir</b> . . . . .	92
Modifier l’horodatage d’un fichier avec touch . . . . .	92
Créer un fichier vide avec touch . . . . .	92
Créer un fichier texte sans éditeur de texte . . . . .	93
<i>Avec cat</i> . . . . .	93
<i>Avec echo</i> . . . . .	93
Créer des répertoires avec mkdir . . . . .	94
Créer une série de répertoires . . . . .	94
Gare aux espaces ! . . . . .	95
Les arborescences en un coup d’œil avec tree . . . . .	96
Créer une arborescence de répertoires . . . . .	98
<b>Copier, déplacer et renommer : cp et mv</b> . . . . .	99
Copier des fichiers et des répertoires avec cp . . . . .	99
Sauvegarder un répertoire . . . . .	101
Déplacer des fichiers et des répertoires avec mv . . . . .	102
Renommer des fichiers et des répertoires avec mv . . . . .	103
<b>Supprimer : rm et rmdir</b> . . . . .	103
Gare aux armes de destruction massive . . . . .	103
Travailler avec ou sans filet ? . . . . .	104
Supprimer des répertoires avec rmdir . . . . .	105
Un coup d’essuie-glace avec clear . . . . .	106
<b>Éditer des fichiers texte : Vi</b> . . . . .	106
Une réputation problématique . . . . .	106
L’éditeur de texte installé sur tous les systèmes Linux . . . . .	107
Vi amélioré : Vim . . . . .	107
Vimtutor . . . . .	108
<b>Travailler moins pour taper plus</b> . . . . .	110
La complétion automatique . . . . .	110
La flemme devient un gage de qualité . . . . .	113
Répéter une commande . . . . .	114
Utiliser l’historique des commandes . . . . .	115
Invoquer une commande en utilisant la recherche inversée . . . . .	115
<b>Consulter l’aide en ligne : man et info</b> . . . . .	116
Unix est long et la vie est brève . . . . .	116
On appelle les secours ? . . . . .	117

Le bonheur est dans le PC . . . . .	117
Afficher le manuel en ligne : man . . . . .	118
Linux, Shakespeare et Molière . . . . .	119
<i>Comment lire une page man ?</i> . . . . .	120
Afficher le manuel en ligne : info . . . . .	121

## CHAPITRE 5

### Gérer les utilisateurs . . . . . 125

Systèmes mono-utilisateurs et systèmes multi-utilisateurs . . . . .	125
Ajouter de nouveaux utilisateurs : useradd . . . . .	126
Utiliser n'est pas administrer . . . . .	128
Changer d'identité et devenir root . . . . .	128
Savoir qui l'on est . . . . .	129
En savoir un peu plus sur les utilisateurs : id, groups, finger . . . . .	130
La gestion des utilisateurs sous le capot . . . . .	132
Comprendre le fichier /etc/passwd . . . . .	132
Les utilisateurs système . . . . .	133
Comprendre le fichier /etc/shadow . . . . .	133
Trouver les utilisateurs physiques du système . . . . .	135
Vue détaillée sur l'ajout d'un utilisateur . . . . .	137
Les données GECOS des utilisateurs . . . . .	137
Le profil par défaut des nouveaux utilisateurs . . . . .	139
<i>Personnaliser le profil par défaut des nouveaux utilisateurs</i> . . . . .	140
Modifier le mot de passe d'un utilisateur . . . . .	141
Associer les utilisateurs aux groupes . . . . .	142
Comprendre le fichier /etc/group . . . . .	142
Supprimer un utilisateur d'un groupe . . . . .	143
Créer et supprimer des groupes . . . . .	144
Supprimer un utilisateur . . . . .	144
Vider l'eau du bain avec le bébé . . . . .	145
Administrer le système avec sudo . . . . .	146

## CHAPITRE 6

### Gérer les droits d'accès . . . . . 149

Qui a le droit de faire quoi ? . . . . .	149
Un exemple pratique . . . . .	150
Comprendre les permissions dans l'affichage détaillé . . . . .	151
Rendre un fichier exécutable . . . . .	152
Ajouter et retirer les droits de lecture et d'écriture . . . . .	153
Une autre approche : la notation numérique . . . . .	154
Attribuer tous les droits à tout le monde . . . . .	154
Retirer et ajouter des droits . . . . .	155
Les permissions par défaut : umask . . . . .	156
Gérer les droits d'accès aux répertoires . . . . .	158
Changer le propriétaire et le groupe d'un fichier . . . . .	161
Pas de cadeaux ! . . . . .	162

## CHAPITRE 7

<b>Chercher une aiguille dans une botte de foin .....</b>	<b>163</b>
Chercher l'emplacement d'un fichier dont on connaît le nom .....	163
Limiter la recherche à certains répertoires .....	164
Zone interdite .....	164
Faire fi des avertissements .....	165
Interrompre une recherche .....	166
Chercher des fichiers dont on ne connaît pas le nom exact .....	166
Chercher selon d'autres critères que le nom. ....	167
Chercher en fonction de la taille .....	167
Appliquer une commande sur les fichiers trouvés. ....	168
Chercher par type. ....	168
Chercher selon les droits d'accès .....	169
Cas pratique : attribuer des permissions à un ensemble de fichiers .....	169
Chercher du texte à l'intérieur d'un fichier .....	170
Chercher du texte dans une série de fichiers .....	172
Affiner la recherche .....	174
Obtenir des informations sur son matériel .....	175

## CHAPITRE 8

<b>Créer et manipuler des liens .....</b>	<b>177</b>
Les liens symboliques .....	177
Créer des liens symboliques .....	178
À quoi servent les liens symboliques ? .....	179
<i>Exemple n°1 : awk</i> .....	179
<i>Exemple n°2 : ex</i> .....	179
<i>Exemple n°3 : gpg</i> .....	180
<i>Conclusion</i> .....	180
Lequel est le vrai ? .....	180
Casser un lien symbolique .....	181
Effacer un lien symbolique .....	182
Les liens physiques .....	183
Un nom de fichier alternatif .....	183
Un lien physique a la vie dure .....	184

## CHAPITRE 9

<b>Gérer les processus.....</b>	<b>185</b>
Linux, un système multitâche .....	185
Surveiller un processus : ps .....	186
Afficher les processus en cours .....	187
Compter les processus. ....	188
Envoyer un signal à un processus : kill .....	188
Arrêter et relancer un processus .....	189
Terminer et tuer un processus .....	190
S'adresser à un processus par son nom : killall .....	191
Utiliser kill ou killall ? .....	192
Les processus dans tous leurs états .....	192

Un lien de parenté : pstree . . . . .	193
Massacre familial . . . . .	194
Tirer à vue sur tout ce qui bouge ? . . . . .	195
Surveiller l'utilisation des processus de la machine : top . . . . .	196

## CHAPITRE 10

### Gérer les services..... 199

Une confusion babylonienne . . . . .	199
Qu'est-ce qu'un service ? . . . . .	200
Arrêter et (re)démarrer un service . . . . .	200
Recharger la configuration d'un service . . . . .	201
Activer un service au démarrage de la machine . . . . .	202
Les cibles systemd . . . . .	204
Démarrer en mode de secours . . . . .	205
Changer de cible à la volée . . . . .	206
Afficher l'état du système . . . . .	207
Définir la cible par défaut . . . . .	208
Éteindre et redémarrer la machine . . . . .	209

## CHAPITRE 11

### Accéder aux périphériques amovibles..... 211

Les fichiers de périphérique . . . . .	211
La notion de montage/démontage . . . . .	213
Monter manuellement un CD ou un DVD . . . . .	213
Démonter le CD ou le DVD . . . . .	215
Monter et démonter manuellement une clé USB . . . . .	215
Scier la branche sur laquelle on est assis . . . . .	218
Éjecter un DVD ou un CD-Rom : eject . . . . .	219
Monter plusieurs périphériques en même temps . . . . .	220
Obtenir des informations sur les périphériques montés . . . . .	220

## CHAPITRE 12

### Partitionner et formater un disque dur..... 223

Le partitionnement automatique . . . . .	223
Les tables de partitionnement MBR et GPT . . . . .	224
Identifier le type de partitionnement sur mon système . . . . .	224
Le schéma de partitionnement le plus simple . . . . .	225
La partition d'échange . . . . .	225
Les systèmes de fichiers pour la partition principale . . . . .	225
Retour à la case départ . . . . .	226
Supprimer les partitions existantes . . . . .	227
Créer la partition d'échange . . . . .	228
Créer la partition principale . . . . .	230
État des lieux . . . . .	233
<i>Comprendre les UUID</i> . . . . .	233
Un schéma de partitionnement un peu plus élaboré . . . . .	234
Préparation du disque . . . . .	235



Créer la partition /boot . . . . .	235
Créer la partition d'échange . . . . .	235
Créer la partition principale . . . . .	235
État des lieux . . . . .	236
Comprendre les étiquettes . . . . .	237
Afficher l'espace occupé et l'espace disponible . . . . .	237
<b>Ajouter un disque à une installation existante . . . . .</b>	<b>238</b>
Insérer et connecter le deuxième disque dur . . . . .	238
Ajouter un disque dans une machine virtuelle . . . . .	239
Vérifier la prise en charge du deuxième disque . . . . .	240
Partitionner le deuxième disque . . . . .	241
Formater le deuxième disque . . . . .	242
<b>Partitionnement manuel d'un système UEFI . . . . .</b>	<b>245</b>
<b>Ajouter un disque avec une table de partitions GPT . . . . .</b>	<b>247</b>
<b>Supprimer une table de partitions GPT . . . . .</b>	<b>250</b>
<b>Formater un disque dur externe ou une clé USB . . . . .</b>	<b>251</b>

## CHAPITRE 13

### **Gérer les archives compressées . . . . . 255**

Les différents formats d'archivage . . . . .	255
<b>Compresser et décompresser un fichier . . . . .</b>	<b>256</b>
Compresser et décompresser un fichier avec gzip . . . . .	256
Compresser et décompresser un fichier avec bzip2 . . . . .	257
<b>Manipuler les archives avec tar . . . . .</b>	<b>258</b>
Créer une archive avec tar . . . . .	258
Extraire les fichiers d'une archive . . . . .	259
Combiner l'archivage et la compression . . . . .	259
Extraire une archive compressée . . . . .	260

## CHAPITRE 14

### **Premiers pas sur le réseau . . . . . 261**

<b>Une confusion babylonienne . . . . .</b>	<b>261</b>
<b>Prérequis matériel et logiciel . . . . .</b>	<b>262</b>
Le modem routeur ADSL . . . . .	262
Questions de câblage . . . . .	263
<i>Choisir le bon type de câble . . . . .</i>	<i>263</i>
<b>La carte Ethernet . . . . .</b>	<b>264</b>
<b>Les pilotes sous Linux . . . . .</b>	<b>264</b>
<b>Principe de fonctionnement des réseaux . . . . .</b>	<b>266</b>
Afficher la configuration des interfaces réseau . . . . .	266
<i>L'adresse MAC de votre carte . . . . .</i>	<i>267</i>
<i>L'adresse IP et le réseau . . . . .</i>	<i>267</i>
<i>IPv4 et IPv6 . . . . .</i>	<i>267</i>
Les adresses IP . . . . .	268
<i>Un peu de pratique . . . . .</i>	<i>268</i>
<i>Un peu de théorie . . . . .</i>	<i>269</i>
<i>Peaufiner l'affichage des adresses IP . . . . .</i>	<i>270</i>
Établir un contact avec une machine distante : ping . . . . .	271

Le routeur : un centre de tri pour paquets numériques . . . . .	272
<i>Sous les pavés numériques, la plage d'adresses IP privées</i> . . . . .	272
<i>Relier le public et le privé.</i> . . . . .	273
<i>Le système de noms de domaine : l'annuaire d'Internet</i> . . . . .	273
<b>Configurer une connexion à Internet</b> . . . . .	274
Configuration dynamique : DHCP . . . . .	274
<i>Paramétrer le serveur DHCP intégré dans le modem routeur</i> . . . . .	275
<i>La configuration DHCP en détail.</i> . . . . .	276
À quoi sert NetworkManager ? . . . . .	278
Configuration statique . . . . .	279
<i>Renseigner les serveurs DNS</i> . . . . .	280
<b>Diagnostiquer le réseau en cas de panne</b> . . . . .	281

## CHAPITRE 15

### **Installer et gérer des logiciels** . . . . . **285**

Petit aperçu pour les utilisateurs de Windows . . . . .	285
<b>Installer des logiciels depuis le code source</b> . . . . .	286
Atelier pratique n° 1 : Joe's Own Editor . . . . .	286
Atelier pratique n° 2 : TCPdump . . . . .	289
Inconvénients d'une installation depuis le code source . . . . .	291
<b>Anatomie d'un paquet RPM</b> . . . . .	293
<b>Installer et gérer des paquets logiciels avec RPM</b> . . . . .	294
Trouver des paquets RPM . . . . .	295
Installer un paquet RPM . . . . .	295
Supprimer un paquet RPM . . . . .	295
Mode bavard ou laconique ? . . . . .	296
Gérer les dépendances entre les paquets RPM . . . . .	296
Mettre à jour des paquets avec RPM . . . . .	298
La suppression des paquets RPM revisitée . . . . .	300
Obtenir des informations sur les paquets RPM . . . . .	300
<i>Sur les paquets installés</i> . . . . .	300
<i>Sur des paquets non installés.</i> . . . . .	303
Avantages et inconvénients d'une installation avec RPM . . . . .	303
<b>Installer et gérer des paquets logiciels avec Yum</b> . . . . .	304
Installer un paquet avec Yum . . . . .	304
Supprimer des paquets avec Yum . . . . .	307
<i>Gérer les dépendances orphelines</i> . . . . .	308
Effectuer une mise à jour . . . . .	308
Afficher les listes de paquets par états . . . . .	310
Gérer les groupes de paquets . . . . .	311
Obtenir des informations sur les paquets . . . . .	313
Rechercher un paquet . . . . .	314
<b>Configuration des dépôts pour Yum</b> . . . . .	315
Où est-ce que ça se configure ? . . . . .	315
Les dépôts officiels de la distribution . . . . .	315
CentOS, le parent pauvre des distributions ? . . . . .	317
Paquets à gogo ! . . . . .	317
Protéger le système de base avec Yum-Priorities . . . . .	317
Configurer les dépôts de paquets officiels . . . . .	318

Configurer le dépôt CR .....	318
Configurer le dépôt tiers EPEL .....	319
Configurer le dépôt tiers ELRepo .....	320
Installer un paquet RPM téléchargé avec Yum .....	321
<b>CHAPITRE 16</b>	
<b>Manipuler le chargeur de démarrage et le noyau .....</b>	<b>325</b>
Le chargeur de démarrage GRUB .....	325
Comprendre la configuration par défaut .....	326
Les mises à jour et le noyau .....	326
Les fichiers de configuration de GRUB .....	327
Afficher les messages de démarrage .....	329
Modifier la résolution de la console .....	329
Faire le ménage dans les kernels .....	330
Installer un kernel récent .....	332
<b>CHAPITRE 17</b>	
<b>Installer et gérer un RAID logiciel .....</b>	<b>337</b>
La fiabilité toute relative des disques durs .....	337
Qu'est-ce que le RAID ? .....	338
RAID matériel ou RAID logiciel ? .....	338
Les niveaux de RAID .....	339
Le niveau RAID 0 .....	339
Le niveau RAID 1 .....	339
Le niveau RAID 10 .....	339
Le niveau RAID 5 .....	340
Le niveau RAID 6 .....	340
Installation de CentOS 7 en RAID 1 .....	340
Simuler une panne .....	344
Simuler une panne plus conséquente .....	347
Installation de CentOS 7 en RAID 6 .....	349
<b>CHAPITRE 18</b>	
<b>Gérer les volumes logiques avec LVM .....</b>	<b>355</b>
Concepts de base .....	355
LVM dans la configuration par défaut de CentOS .....	356
Une installation personnalisée basée sur LVM .....	357
Redimensionnement à chaud de la partition principale .....	361
<b>CHAPITRE 19</b>	
<b>Gérer les logs du système .....</b>	<b>367</b>
Un peu de pratique avant la théorie .....	367
Les logs au quotidien .....	368
Aperçu sur l'évolution des logs sous Linux .....	369
Gérer le journal de systemd .....	369
Rendre le journal persistant .....	370
Définir les permissions nécessaires pour lire les logs .....	371

Quelques options pratiques au quotidien . . . . .	371
La configuration des logs avec rsyslog . . . . .	373
La rotation des logs . . . . .	374

**ANNEXE A****Installer un poste de travail Linux . . . . . 377**

Le choix de la distribution . . . . .	377
Confectionner les supports d'installation . . . . .	378
Réinitialiser le disque dur . . . . .	379
Installer OpenSUSE Leap 15.0 . . . . .	379
Démarrer l'installation . . . . .	379
Partitionner le disque dur . . . . .	380
Terminer la configuration des paramètres . . . . .	382
Redémarrage initial . . . . .	384

**Index . . . . . 387**

# 1

## Si Linux m'était conté

---

*Le 16 septembre 2004, Linux est sorti tout ficelé du crâne de son inventeur Mark Shuttleworth, revêtu de son armure et prêt à conquérir le monde.  
Je plaisante.*

### Au commencement était Unix

Linux ? Unix ? Logiciels libres ? Faisons donc un petit voyage dans le temps pour y voir un peu plus clair. Pas forcément dans la préhistoire, car les tables à calcul de l'ère babylonienne datant de plus de quatre mille ans ne sont que d'un intérêt limité pour nous. Arrêtons-nous plutôt vers le milieu du XX<sup>e</sup> siècle et situons là le début de « notre » préhistoire.

Les premiers ordinateurs construits à l'époque de la guerre froide étaient de véritables monstres bibliques. Leur force de calcul était constituée de dizaines de milliers de tubes électroniques ou « lampes ». Ils occupaient l'équivalent d'un terrain de foot ou d'un hangar, consommaient autant d'électricité qu'un village entier et dégageaient plus de chaleur qu'une mare de geysers en activité.

#### **CITATION Retour vers le futur**

« Je crois qu'il y a un marché mondial pour à peu près cinq ordinateurs. »  
Thomas Watson, président d'IBM, 1958

Vers le début des années 1960, la taille des machines a pu être réduite de façon considérable avec l'avènement des semi-conducteurs. Les tubes électroniques ont été successivement remplacés par des transistors, puis par des circuits intégrés. Cette nouvelle génération de machines occupait à peine l'espace d'une collection d'armoires normandes. Malheureusement, toutes souffraient toujours du même défaut majeur.

Chacun de ces ordinateurs disposait en effet de son propre système d'exploitation, conçu en même temps que la machine et taillé sur mesure par la force des choses. Un ingénieur qui avait appris à se servir de l'une d'entre elles et qui souhaitait travailler sur un autre type de machine était contraint de jeter par-dessus bord tout son savoir-faire pour revenir à la case départ et tout réapprendre depuis le début. Si seulement toutes ces machines pouvaient parler le même langage... Il fallait trouver une solution.

### B.A.-BA Système d'exploitation

Un système d'exploitation ou OS (*Operating System*) est un ensemble de logiciels qui gère les fonctions les plus élémentaires d'une machine. D'une part, il contrôle les périphériques entrée/sortie comme le clavier et l'écran, ce qui permet à un humain de communiquer avec l'ordinateur. D'autre part, il s'occupe de la répartition intelligente des ressources de la machine comme le processeur et la mémoire.

Une machine dépourvue de système d'exploitation ne sera donc même pas capable de démarrer un programme. Sans système installé, même un PC moderne vous servira tout au plus à caler la porte de la terrasse un jour de mistral.

Le projet Multics (*Multiplexed Information and Computing Service*) a été initié en 1964 pour apporter précisément cette solution. L'ambition de Multics consistait à fournir un système d'exploitation **portable**, c'est-à-dire capable d'être porté sur la plupart des machines existantes ; ambition pharaonique car, si le projet réussissait, il mettrait fin à la confusion babylonienne des systèmes d'exploitation.

Multics n'a connu qu'un succès modeste, comme cela arrive parfois avec les projets pharaoniques. Hormis quelques thésards en informatique et une poignée de vétérans, l'humanité a même fini par l'oublier. Ce qui nous est resté de Multics, c'est une série de bonnes idées, mais surtout sa descendance totalement imprévue : un rejeton qui n'était pas au programme, mais qui a fini par connaître un succès incroyable.

1969, l'année où l'astronaute Neil Armstrong se promène sur la Lune, deux ingénieurs des laboratoires Bell<sup>1</sup>, Dennis Ritchie et Ken Thompson, décident d'écrire un système d'exploitation pour l'ordinateur dont ils disposent plus ou moins librement dans leur bureau. Cette machine, un DEC PDP-7, est considérée comme un « mini-ordinateur » à l'époque. Pour avoir une vague idée de la taille de l'engin, imaginez une batterie de quatre ou cinq réfrigérateurs de taille familiale posés les uns à côté des autres. Ritchie et Thompson se servent des bouts de code du projet Multics, mais leur ambition est bien plus modeste, pour ne pas dire purement ludique. Ce qui les motive dans l'immédiat, c'est de disposer d'une machine suffi-

---

1. L'équivalent de France Télécom, aux États-Unis

samment fonctionnelle pour jouer à un jeu tout à fait dans l'air du temps : *Space Travel*, un jeu interactif en mode texte, où il s'agit de poser une capsule spatiale sur la Lune.

**Figure 1-1**

Ken Thompson et Dennis Ritchie,  
les fondateurs d'Unix.



Leur projet est un succès, comme cela arrive souvent lorsqu'on fait quelque chose uniquement pour s'amuser et que l'ambition ne vient pas gâcher le plaisir. Étant donné que leur nouveau système tourne sur une seule machine, celle qu'ils ont à portée de main, ils s'amuse à faire un jeu de mots sur le système qui les a inspirés et décident de le baptiser « Unics ».

#### ÉTYMOLOGIE Le « X » de Unix

Le nom « Unics » est contracté par la suite en « Unix ». Cette consonne finale sera caractéristique d'un certain nombre de variantes, dérivées et clones d'Unix : XENIX, AIX, HP-UX, Ultrix, IRIX, Minix... sans oublier Linux et Mac OS X.

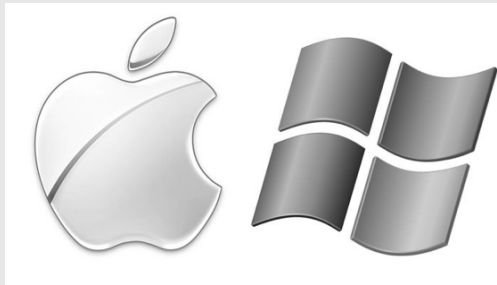
La fin des années 1960, ce n'est pas seulement la conquête spatiale, mais également la révolution de mai 1968, Woodstock, le *flower power* et la culture hippie. Les campus des universités et les entreprises pullulent de barbus à sandales qui ont une idée en tête : contribuer au code d'Unix en vue de l'améliorer. Certes, la propriété intellectuelle et les brevets existent déjà, mais cela n'empêche personne de vivre pour autant. Les *hackers* – au sens noble du terme – échangent entre eux leurs meilleures idées et les bouts de code source qui vont avec, aussi naturellement que leurs grands-mères échangent entre elles leurs meilleures recettes de cuisine. Les entreprises et les facultés ne payent pas de frais de licence pour utiliser Unix et, lorsqu'elles réclament le code source à Ken Thompson, celui-ci a l'habitude d'ajouter un petit mot au colis de bandes magnétiques et de disquettes : « *Love, Ken* ».

**CULTURE Code source et programme exécutable**

Les sources d'un programme, c'est l'ensemble des fichiers qui contiennent du code et que l'on compile pour obtenir un programme exécutable. Lorsqu'on distribue un programme sous forme binaire, il est prêt à l'emploi, mais on ne peut pas le modifier. De façon analogue, une recette de tarte aux pommes ainsi que tous les ingrédients nécessaires comme la farine, les œufs, les pommes, la cannelle et le sucre permettent de cuisiner une tarte aux pommes. Une fois qu'elle est sortie du four, il est difficile de revenir en arrière pour changer les ingrédients de base et la recette.

**L'AIR DU TEMPS L'informatique avant Microsoft et Apple**

Imaginez-vous en 1970 : Microsoft et Apple n'existent pas encore et personne n'aurait l'idée d'associer des mots de tous les jours comme « windows » ou « apple » à de l'informatique. Vous jetez un coup d'œil dans votre boule de cristal. Vous apercevez le futur pas trop lointain, quelques décennies plus tard. Vous annoncez solennellement qu'un jour viendra où les systèmes et les applications se vendront **sans** le code source qui va avec. Dans des cartons au graphisme léché. Des boîtes remplies majoritairement de vide comme les *cornflakes*. Ornées de fenêtres multicolores ou d'une pomme stylisée. Le prix sera conséquent, les gens devront acheter les boîtes avec le matériel et les ventes feront de vous l'homme le plus riche de la terre. On vous prend probablement pour un fou.



**Figure 1-2** À quoi ressemblait l'informatique en 1970, avant Microsoft et Apple ?

Durant les années 1970 et le début des années 1980, les universités utilisent à peu près exclusivement Unix. Les entreprises décident d'emboîter le pas et l'adoptent également à grande échelle. Après tout, les étudiants d'aujourd'hui font les ingénieurs de demain. Techniquement, Unix est à la pointe des systèmes d'exploitation. C'est un vrai système multitâche et multi-utilisateur, robuste et transparent. Il définit clairement les droits d'accès aux fichiers, il sépare les processus bien proprement et il est conçu dès le départ pour fonctionner en réseau. Petit à petit, Unix est en bonne voie de faire tourner les ordinateurs du monde entier.

L'âge d'or d'Unix connaît une fin abrupte et quelque peu absurde en 1983. Dans le cadre de la lutte antitrust du gouvernement de Ronald Reagan, les laboratoires Bell sont séparés de leur maison mère, l'entreprise de télécommunications AT & T (*American Telephone and Telegraph*). Dans la foulée des actions judiciaires qui s'ensuivent, un décret qui empêchait la commercialisation d'Unix jusque-là est rendu caduc. AT & T décide de sauter dans la brèche ouverte par la nouvelle législation et de s'approprier le système Unix et tout le code qui va



avec, en faisant fi des nombreuses contributions externes. L'émoi causé par cette mainmise – qui a failli sonner le glas du système – est considérable dans la communauté des *hackers*.

Les étudiants qui ont contribué au code d'Unix s'estiment doublement lésés. D'une part, AT & T « oublie » de les rémunérer alors que les licences sont monnayées au prix fort. D'autre part, ils n'ont plus accès à leur propre code ou – situation plus ubuesque encore – n'ont plus le droit de l'utiliser pour de sombres raisons de propriété intellectuelle.

Certes, AT & T essaie de calmer le jeu en annonçant que les universités pourront désormais bénéficier de tarifs préférentiels pour les licences. Il n'empêche que l'accès au code source est dorénavant restreint. Du jour au lendemain, Unix est devenu un système d'exploitation rigoureusement propriétaire et commercial.

#### MARKETING Les systèmes propriétaires dans l'éducation

Quelques décennies plus tard, le tarif préférentiel pour les élèves et les étudiants demeure une stratégie de fidélisation populaire auprès des éditeurs de systèmes et de logiciels propriétaires.

## Richard Stallman et le projet GNU

La commercialisation d'Unix marque l'avènement d'un véritable âge de fer en informatique. La « culture *hacker* » des premières années cède la place à une logique restrictive, commerciale et propriétaire. Cette transition ne s'est pourtant pas faite en un jour. Elle a été marquée par une série de signes avant-coureurs.

**Figure 1–3**  
Richard Stallman,  
l'initiateur du projet GNU.



Revenons un peu en arrière, en 1980, et rendons visite à Richard Stallman dans son laboratoire d'Intelligence artificielle au *Massachusetts Institute of Technology* (MIT). Richard est

confronté à un problème qu'il n'arrive pas à résoudre. La nouvelle imprimante laser du laboratoire, une Xerox 9700, se bloque régulièrement et refuse d'imprimer suite à des erreurs de bourrage papier. En bon *hacker* qui se respecte, Richard aime relever les défis techniques et le dysfonctionnement d'un périphérique tombe dans cette catégorie.

#### ENFER Les imprimantes

Les imprimantes récalcitrantes nous viennent droit de l'enfer. Lisez la BD géniale de Matthew Inman à ce sujet : *Why I believe printers were sent from hell to make us miserable*.

► <http://theoatmeal.com/comics/printers>

La précédente imprimante, une Xerox XGP, avait connu exactement le même problème de bourrage papier et Richard l'avait résolu comme un informaticien de l'époque pouvait le faire : il avait réclamé le code source du pilote à Xerox et s'était plongé dans sa lecture. Après avoir identifié l'erreur, il lui avait suffi de modifier et recompiler le code pour que l'imprimante fonctionne correctement.

#### GLOSSAIRE Pilote/driver

Un pilote de périphérique ou *driver* est le code qui permet au système d'exploitation de communiquer avec le périphérique en question.

Or, le problème auquel Richard se heurte cette fois-ci n'est pas d'ordre technique. Le fabricant Xerox vient en effet d'opposer un refus à sa demande, estimant que le code source est désormais un secret de fabrication. Richard ne peut donc pas y accéder, encore moins l'étudier ou le corriger. En revanche, Xerox l'invite à « envoyer un rapport d'erreurs », afin que les ingénieurs de l'entreprise étudient le problème à sa place et mettent à disposition une mise à jour qui corrigera éventuellement le dysfonctionnement.

#### BUG Le fin mot de l'histoire

Richard Stallman a effectivement soumis le rapport d'erreurs suggéré par Xerox. Il n'a jamais reçu de réponse.

Richard sent naître en lui un mélange de colère et d'impuissance. Les constructeurs de matériel informatique tendent visiblement à ne plus livrer que des pilotes au format binaire, sans le code source qui va avec. L'utilisation de licences logicielles restrictives s'impose manifestement comme une nouvelle norme. C'est une véritable gangrène qui touche le monde de l'informatique et le pourrit de l'intérieur. Il faut donc trouver une solution, une force nouvelle qui puisse contrecarrer cette tendance funeste.

Le 27 septembre 1983, quelques mois après la mainmise d'AT & T sur Unix, Richard Stallman poste un message sur Usenet pour annoncer la naissance du projet GNU, un système d'exploitation libre compatible avec Unix.

**BIBLIOGRAPHIE Richard Stallman et la révolution du logiciel libre**

Pour en savoir plus sur la vie et l'œuvre de Richard Stallman, lisez sa biographie autorisée publiée chez Eyrolles.

📖 *Richard Stallman et la révolution du logiciel libre*, ISBN : 978-2-212-12609-9

**CULTURE HACKER Acronyme récursif**

GNU signifie *GNU's Not Unix*, c'est-à-dire « GNU n'est pas Unix ». C'est un acronyme récursif, l'équivalent linguistique d'un chat qui se mord la queue. L'acronymie récursive est assez répandue en informatique.



**Figure 1-4** Le logo du projet GNU.

Pour Richard comme pour beaucoup d'autres, Unix reste le système d'exploitation de référence, pour toutes les raisons évoquées plus haut. Son seul défaut, c'est qu'il n'est pas libre. L'ambition du projet GNU consiste ni plus ni moins qu'à réinventer la roue et proposer un système d'exploitation libre 100 % compatible Unix mais qui, justement, n'est pas Unix, c'est-à-dire qu'il n'en contient aucune ligne de code.

**URL Le projet GNU**

► <http://www.gnu.org>

Un système d'exploitation comme Unix n'est pas un bloc monolithique. Il est composé d'une multitude de petits programmes, dont chacun s'acquitte d'une tâche bien définie. Cette modularité va considérablement faciliter la tâche au projet GNU, qui se pose comme but concret de remplacer l'un après l'autre chacun des composants d'Unix par un équivalent libre.

C'est donc un projet d'envergure, une vaste mosaïque qu'il s'agit de compléter avec beaucoup de patience, morceau par morceau. Richard Stallman lui-même démissionne de son poste au MIT en janvier 1984 pour se consacrer entièrement au projet GNU et développer quelques logiciels significatifs : un compilateur, un débogueur, une collection d'outils basiques et l'éditeur de texte Emacs.

Richard comprend très vite que le projet GNU a besoin d'une infrastructure légale pour assurer sa pérennité et lui éviter d'être cannibalisé par les éditeurs de logiciels propriétaires. En 1985, il crée la FSF (*Free Software Foundation*), une organisation à but non lucratif pour la défense et la

promotion du logiciel libre. Cette même année, il publie le *Manifeste GNU*, un texte fondateur qui porte aussi bien sur l'aspect technique et social du projet que sur sa philosophie.

#### URL La Free Software Foundation

► <https://www.fsf.org>

Le terme de « logiciel libre » est également clarifié ; en effet, le terme anglais *free* comporte une ambiguïté et il s'agit de distinguer *free as in speech* (libre, dans le sens de « liberté de la parole ») et *free as in beer* (gratuit, dans le sens de « bière à volonté »). Selon la définition proposée par Richard, un logiciel est libre s'il respecte les quatre conditions fondamentales suivantes :

- liberté d'utiliser le logiciel ;
- liberté de le copier ;
- liberté d'en étudier le fonctionnement ;
- liberté de le modifier et de redistribuer cette version modifiée.

Une seule obligation permet de préserver ces quatre libertés : toute personne qui souhaite apporter des modifications au code source d'un logiciel – en vue de l'améliorer ou d'en modifier le comportement – est tenue de publier ces modifications sous les mêmes conditions, en respectant à son tour les quatre libertés fondamentales. C'est l'application du principe du *copyleft* (un jeu de mots sur *copyright*, « droit d'auteur ») qui évite notamment l'appropriation du code source libre par une entreprise. Ce principe est entériné dans la licence publique générale GNU (ou licence GPL), que Stallman publie en 1989.

#### CULTURE La viralité de la GPL

Un logiciel libre publié sous licence GPL restera libre indépendamment de toutes les modifications qu'il subira. Cette préservation des quatre libertés fondamentales est garantie par le caractère viral de la GPL. Dans un entretien avec le *Chicago Sun-Times* le 1<sup>er</sup> juin 2001, le PDG de Microsoft, Steve Ballmer, associera la GPL à « un cancer qui s'accroche à la propriété intellectuelle ». (*sic*).

Durant la seconde moitié des années 1980, le projet GNU progresse lentement, mais sûrement. En 1990, l'ensemble des composants est réalisé et il ne manque plus que le noyau du système. Or, le noyau – ou *kernel* – constitue également la partie la plus importante du code, la pièce maîtresse, celle qui se situe le plus près du matériel et qui contrôle les fonctions élémentaires comme la gestion de la mémoire et des processus, le contrôle des périphériques, etc.

Pour compléter son système d'exploitation libre, la FSF (*Free Software Foundation*) lance Hurd, un projet de noyau libre pour les systèmes Unix en général et GNU en particulier. Malheureusement, le développement s'avère long et fastidieux. Le projet va battre de l'aile dès les débuts et Hurd fera progressivement sa descente vers les limbes de ces grands projets informatiques voués à rester éternellement en chantier.

## Linus Torvalds et le noyau Linux

Début janvier 1991, de l'autre côté de l'Atlantique, en Finlande, le jeune étudiant Linus Torvalds décide d'investir dans du matériel informatique. Il n'hésite pas à s'endetter sur trois ans pour acheter ce qui se fait de mieux à l'époque en matière d'ordinateur personnel : un IBM PC 30 386 flambant neuf, équipé d'un processeur 32-bits.

Il commande un jeu de disquettes d'installation du système Minix, une variante pédagogique d'Unix développée par le professeur Andrew Tanenbaum, qui lui servira pour les cours sur l'architecture des systèmes d'exploitation qu'il suit à l'université d'Helsinki. Tanenbaum est l'auteur de l'ouvrage de référence en la matière, *Operating Systems Design and Implementation*, un pavé de plus de sept cents pages. Selon ses propres dires, Linus Torvalds passe les premiers mois de l'année 1991 à faire deux choses principalement :

- 1 Rien.
- 2 Lire *Operating Systems Design and Implementation*.

### L'AIR DU TEMPS Les ordinateurs personnels

Si vous êtes né vers la fin des années 1960 ou avant, vous vous souvenez peut-être de cette époque glorieuse où les micro-ordinateurs ont fait leur apparition dans les foyers. Les machines 8-bits de la première génération – comme l'Apple II, le TRS-80 de Radio Shack, le Commodore VC-20 et le Commodore 64, le ZX-81 et le ZX Spectrum de Sinclair – étaient généralement livrées avec un système d'exploitation rudimentaire – néanmoins propriétaire – comprenant un interpréteur BASIC. Comme Linus Torvalds, j'ai fait mes premiers pas en programmation au début des années 1980, sur un VC-20, une machine sans disque dur, dotée de 3,5 kilo-octets (!) de RAM et d'un lecteur spécial qui utilisait les cassettes audio bon marché comme bandes magnétiques.

Par la suite et jusqu'à ce jour, les ordinateurs sont majoritairement *compatibles PC*, c'est-à-dire issus en ligne directe de l'IBM PC (*Personal Computer*) de 1981. Au début des années 1990, ces machines sont à peu près exclusivement équipées du système d'exploitation propriétaire et commercial MS-DOS (*Microsoft Disk Operating System*), livré sous forme d'une série de disquettes souples (*floppy*, en anglais). Quant aux ordinateurs de la marque Apple, ils tournent sous Mac OS depuis le lancement du Macintosh, le fameux petit cube beige présenté au public en 1984. Mac OS est un système d'exploitation propriétaire et commercial spécifiquement développé par Apple pour son matériel.

Au début des années 1990, l'installation d'un Unix commercial sur un ordinateur personnel reste hors de portée pour les particuliers, pour deux raisons. D'une part, le prix d'une licence Unix est tout bonnement prohibitif, de l'ordre de quelques milliers de dollars. D'autre part, Unix est relativement gourmand en ressources et son utilisation nécessite l'acquisition d'une station de travail puissante et tout aussi onéreuse.

Au bout d'un mois d'attente, Linus Torvalds finit par trouver le jeu de disquettes Minix dans sa boîte aux lettres, ce qui lui permet de remplacer le système MS-DOS livré en version réduite avec son matériel par quelque chose qui ressemble à Unix. Malgré sa visée pédagogique, Minix reste un système Unix sous le capot et son code source est disponible. Linus va passer les mois suivants à se faire les dents sur son nouveau système et à l'explorer dans ses moindres recoins pour en découvrir les possibilités.

**URL Minix**

Le système d'exploitation Minix (*Minimal Unix*) est toujours utilisé à des fins pédagogiques. Il en est actuellement à sa version 3.3.0.

► <https://www.minix3.org/>

Les limitations de Minix ne tardent pas à frustrer Linus, notamment l'émulateur de terminal qui est censé gérer la connexion à distance à l'ordinateur de l'université. Les hivers sont rudes en Finlande et, lorsqu'un étudiant en informatique veut éviter de sortir de chez lui, la possibilité de se connecter à distance pour travailler devient une nécessité. Certes, l'émulateur de terminal inclus dans Minix permet à Linus de lire ses courriels et de poster des messages sur Usenet, mais le téléchargement de fichiers à distance ne fonctionne pas. Ajoutons à cela le fait que l'architecture 16-bits de Minix ne tire pas pleinement profit des ressources de son processeur. Enfin, même si le code source est disponible, le système n'est pas vraiment libre, au vu des restrictions imposées par Andrew Tanenbaum quant à la modification et à la redistribution du code.

**CULTURE INTERNET Usenet**

Usenet est un système en réseau de forums – ou newsgroups – qui a existé bien avant le World Wide Web. Même s'il est rendu obsolète par les forums sur le Web, beaucoup de gens s'en servent encore. Usenet s'utilise avec un lecteur de nouvelles graphiques comme Pan, KNode ou Thunderbird, ou avec un client en ligne de commandes comme tin ou slrn. Alternativement, les groupes Google permettent d'y accéder à l'aide d'un simple navigateur web.

► <https://groups.google.com>

Linus a bientôt terminé la lecture d'*Operating Systems Design and Implementation*. Il sent que les cours sur Minix ne lui apportent plus grand-chose. Il décide donc de les sécher et de mettre à profit le temps ainsi gagné pour coder son propre émulateur de terminal. Il part littéralement de zéro, c'est-à-dire qu'il écrit son code directement au niveau du matériel, sans passer par Minix.

Au bout de quelques semaines de travail acharné, il réussit à se connecter à l'ordinateur de l'université pour lire ses courriels et poster dans les *newsgroups*. Il résout le problème du chargement des fichiers distants en écrivant un pilote pour son disque dur et un autre pour le système de fichiers utilisé par Minix. Il ajoute d'autres fonctionnalités au gré de son inspiration, dans l'effervescence de la création. Petit à petit, son émulateur de terminal se transforme en véritable petit système d'exploitation à part entière.

Le 3 juillet, Linus poste un message sur [news://comp.os.minix](https://news://comp.os.minix) pour se renseigner sur les standards POSIX. Le message en lui-même ne compte que quatre lignes et la question semble anodine en apparence, mais elle attire l'attention des lecteurs du groupe. Si ce Linus Torvalds a besoin des standards POSIX, c'est qu'il doit vraisemblablement concocter son propre système d'exploitation. Ari Lemmke, l'administrateur système de l'université, lui crée spontanément un compte sur le serveur FTP pour lui permettre de ranger les fichiers de son nouveau système et pour que le public y ait accès.

**GLOSSAIRE POSIX**

POSIX désigne la famille des standards valables pour les systèmes d'exploitation Unix. L'abréviation signifie Portable Operating System Interface et le « X » final désigne l'héritage Unix.

Le 25 août 1991, Linus Torvalds poste un message désormais célèbre sur `news://comp.os.minix`. J'essaie de le traduire aussi fidèlement que possible :

```
De      : torvalds@klaava.helsinki.fi (Linus Benedict Torvalds)
Groupe  : comp.os.minix
Sujet   : Qu'est-ce que vous aimeriez bien voir dans Minix ?
Date    : 25 Août 91
```

Salut à tous les utilisateurs de Minix -

Je construis un système d'exploitation (libre) pour les clones 386(486)AT (juste pour le fun, rien de gros et pro genre gnu). C'est en train de mijoter depuis avril et ça commence à être prêt. J'aimerais avoir des retours sur ce que les gens aiment/n'aiment pas dans Minix, vu que mon OS y ressemble plus ou moins (même organisation physique du système de fichiers [pour des raisons pratiques] entre autres choses).

Pour l'instant, j'ai porté bash(1.08) et gcc(1.40) ; ça a l'air de marcher. Ce qui signifie que j'aurai quelque chose d'utilisable d'ici quelques mois et j'aimerais savoir ce que les gens voudraient comme fonctionnalités. Toutes les suggestions sont les bienvenues, mais je ne promets pas de les implémenter :-)

Linus (torvalds@kruuna.helsinki.fi)

PS. Oui - il est débarrassé de tout code Minix et il a un système de fichiers multithread. Il n'est PAS portable et il ne supportera probablement jamais autre chose que les disques durs AT, étant donné que c'est tout ce que j'ai :-)

Quelques semaines plus tard, le 17 septembre 1991, Linus décide de publier la version 0.01 de son système sur le serveur FTP de l'université. Il choisit le nom de Freax, une contraction de *free* (libre/gratuit), de *freak* (obsédé) et du « X » final caractéristique des Unix. Ari Lemmke n'apprécie pas du tout ce nom et décide de son propre chef de nommer le répertoire de téléchargement `/pub/OS/Linux`.

Linus n'envoie pas d'annonce officielle pour la version 0.01 et se contente d'en informer quelques amis et collègues par courriel. En revanche, la publication de la version 0.02 est annoncée solennellement sur `news://comp.os.minix` et beaucoup d'utilisateurs de Linux considèrent que le message correspondant marque la « véritable » naissance de Linux :

Dans le contexte de l'époque, ce message est un véritable pavé dans la mare. Voilà quelqu'un qui a réussi à faire tourner un « vrai » Unix sur son ordinateur personnel. Un Unix libre de surcroît, avec son propre noyau libre et les outils GNU qui vont avec : le compilateur GCC, l'interpréteur de commandes Bash, l'éditeur de flux sed, etc.

Cette fois-ci, les contraintes matérielles et les lacunes du projet constituent juste autant de défis à surmonter. Les réactions de la communauté ne tardent pas à venir et elles sont aussi

De : torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)  
Groupe : comp.os.minix  
Sujet : Sources d'un noyau libre de type Minix pour 386-AT  
Date : 5 octobre 91

Vous regrettez le bon vieux temps de Minix-1.1, lorsque les hommes étaient des hommes et écrivaient eux-mêmes les pilotes pour leur matériel ? Vous n'avez pas de projet intéressant en cours et vous crevez d'envie de mettre les mains dans le cambouis d'un OS que vous pouvez adapter à vos besoins ? Ça vous agace quand tout marche bien sous Minix ? Finies les nuits blanches pour faire fonctionner un petit logiciel bien pratique ? Alors ce message est sûrement pour vous.

Comme je l'ai dit il y a un mois (?), je travaille en ce moment sur une version libre d'un minixoïde pour les machines de type AT-386. J'en suis à un stade où c'est même utilisable (ou pas, ça dépend de ce que vous voulez) et je suis prêt à distribuer les sources à plus grande échelle. C'est juste la version 0.02 (+1 [très petit] patch déjà), mais j'ai réussi à faire tourner bash/gcc/gnu-make/gnu-sed/compress etc. avec.

Les sources de mon petit projet sont disponibles à nic.funet.fi (128.214.6.100) dans le répertoire /pub/OS/Linux. Ce répertoire contient également un fichier README et une poignée de binaires pour travailler sous linux (bash, update et gcc, que demander de plus :-)). Les sources complètes du noyau sont fournies, vu qu'aucune ligne de code Minix n'a été utilisée. Les sources des bibliothèques ne sont que partiellement libres, elles ne peuvent donc pas être redistribuées pour le moment. Le système peut être compilé tel quel et il est censé fonctionner. Heh. Les sources des binaires (bash et gcc) sont disponibles au même endroit, dans /pub/gnu.

ALERTE ! AVERTISSEMENT ! ATTENTION ! Ces sources ont tout de même besoin de minix-386 pour être compilées (et gcc-1.40, peut-être même 1.37.1, pas testé) et il vous faut Minix pour le configurer si vous voulez le faire tourner, donc ce n'est pas encore un système indépendant pour ceux qui n'auraient pas Minix. J'y travaille. Il vous faut également avoir l'esprit bidouilleur pour le configurer (?), donc ceux qui cherchent une alternative à minix-386, passez votre chemin. Pour l'instant, ça s'adresse aux bidouilleurs qui s'intéressent aux systèmes d'exploitation et aux 386 avec un accès à Minix.

Le système a besoin d'un disque dur compatible AT (l'IDE c'est très bien) et EGA/VGA. Si vous êtes toujours intéressé, procurez-vous le README/RELNOTES sur le ftp et/ou envoyez-moi un mail pour des infos supplémentaires.

Je vous entends déjà (enfin, presque) vous demander : "Pourquoi?". Hurd va sortir d'ici un an (ou deux, ou dans un mois, qui sait) et j'ai déjà Minix. C'est un système fait par un bidouilleur pour les bidouilleurs. Je me suis bien amusé à l'écrire et ça fera peut-être plaisir à quelqu'un d'y jeter un coup d'œil et même de l'adapter à ses propres besoins. Il est suffisamment petit pour qu'on puisse le comprendre, l'utiliser et le modifier. J'attends les commentaires que vous pourrez y apporter.

[...]

Linus



enthousiastes que nombreuses. C'est une véritable avalanche que Linus a déclenchée. Les *hackers* du monde entier lui envoient une foule de suggestions et de demandes, mais également des rapports de bogues.

#### CULTURE Le bogue informatique

Un *bug* (ou bogue dans la version francisée), c'est une erreur de conception dans un logiciel, qui entraîne son dysfonctionnement. Le mot s'explique par le dysfonctionnement historique d'un des premiers ordinateurs électromécaniques, dû à un insecte (*bug*) coincé dans un relais.

Durant les mois à venir, Linus va peaufiner le code de son système en interaction constante avec la communauté. Celle-ci se montre d'ailleurs reconnaissante en organisant spontanément une collecte grâce à laquelle Linus s'acquittera des mensualités restantes pour sa machine de développement. Fin 1991, il décide d'aller voir une conférence de Richard Stallman sur le projet GNU à l'université d'Helsinki. Il est impressionné par la présentation et prend une décision d'une importance capitale pour la suite du projet.

**Figure 1-5**  
Linus Torvalds, créateur et  
mainteneur du noyau Linux.



Les toutes premières versions de Linux étaient certes libres, avec une restriction cruciale toutefois : l'utilisation commerciale du code était interdite. En janvier 1992, Linus Torvalds annonce son intention de libérer complètement le code de son noyau en le publiant dorénavant sous licence *GNU General Public License*. Ce changement de licence va encourager de nombreux utilisateurs dans le monde à migrer de Minix vers Linux.

## Les ancêtres Slackware, Red Hat et Debian

En 1992, l'installation d'un système Linux – ou plus exactement d'un système GNU et d'un noyau Linux – reste une aventure réservée à une élite de bidouilleurs avec un certain sens de l'aventure, comme Linus Torvalds l'a précisé dans son message détaillé sur Usenet : *this is a program for hackers by a hacker*.

Les toutes premières versions de Linux sont distribuées sous forme de deux disquettes qui fournissent le minimum syndical du système :

- la première disquette est amorçable et contient le noyau ;
- la deuxième disquette fournit une panoplie d'outils GNU qui servent entre autres choses à créer un système de fichiers.

Un tel système minimal ne permet pas de faire grand-chose. Le nombre de logiciels libres disponibles au téléchargement augmente certes de façon exponentielle, mais leur installation reste une procédure longue et pénible. Le code source de chaque composant d'un logiciel doit d'abord être téléchargé au compte-gouttes via une connexion téléphonique. Une fois qu'on a récupéré le code source, il faut le configurer manuellement pour l'adapter au système avant de lancer la compilation et l'installation. Chacune de ces étapes est hautement chronophage et engloutit facilement des après-midi ensoleillés, voire des week-ends entiers.

Les premières distributions Linux apparaissent dans le but de simplifier la procédure d'installation : Boot Root, Yggdrasil Linux/GNU/X, SLS Softlanding Linux Systems et d'autres encore.

#### **CULTURE Qu'est-ce qu'une distribution ?**

Pour vous faire une idée plus précise de ce que peut être une distribution Linux, pensez au mot *pack*, un peu comme ce qu'on vous vend dans les magasins de sport pour faire du badminton ou de la plongée sous-marine : deux raquettes, un volant, un filet et deux piquets, ou alors la paire de palmes, le masque et le tuba, et vous voilà opérationnel. Une distribution Linux sera donc idéalement constituée d'un ensemble cohérent composé en règle générale :

- du système de base ;
- d'une série d'outils d'administration ;
- d'une panoplie logicielle ;
- d'un installateur.

Ces premières distributions souffrent toutes d'une série de défauts prohibitifs. Soit elles sont pathologiquement minimalistes et n'offrent guère de confort, soit la panoplie de logiciels fournis est déjà plus complète, mais l'installateur est truffé de bogues à tel point que l'ensemble reste à peu près inutilisable.

## Slackware Linux

Le 16 juillet 1993, Patrick Volkerding, étudiant en informatique à la Minnesota State University Moorhead, annonce la publication de Slackware Linux 1.00 sur [news://comp.os.linux](mailto:news://comp.os.linux) :

```
De      : Patrick J. Volkerding (bf703@cleveland.Freenet.Edu)
Sujet   : ANNONCE: Slackware Linux 1.00
Groupes : comp.os.linux
Date    : 1993-07-16 17:21:20 PST
```

La distribution Slackware Linux (v. 1.00) est disponible dès à présent sur FTP anonyme. Il s'agit d'un système d'installation complet, conçu pour les systèmes avec une disquette floppy 3.5". Il a été abondamment testé avec un système 386/IDE. Le noyau par défaut n'offre pas de support SCSI, mais s'il y a suffisamment de demande, ça pourrait me motiver pour compiler une poignée de noyaux spécifiques que je mettrais sur FTP.

Cette version est largement basée sur le système SLS, avec une série de modifications et d'améliorations substantielles. Elle est constituée de deux principaux groupes de disquettes, A (13 disquettes) et X (11 disquettes).

[...]

```
--
Patrick Volkerding
volkerdi@mhd1.moorhead.msus.edu
bf703@cleveland.freenet.edu
```

### HISTOIRE **Slackware Linux 1.0**

Le texte complet de l'annonce de Slackware Linux 1.0 est toujours en ligne sur le site de Slackware.

► <http://www.slackware.com/announce/1.0.php>

On peut considérer que Slackware est bien la première distribution Linux, dans la mesure où il s'agit du premier système Linux réellement utilisable. Contrairement à l'ancêtre SLS Softlanding Linux Systems sur lequel elle se base, Slackware ne contraint plus les utilisateurs à sauter à travers des cerceaux en feu pour obtenir un système Linux raisonnablement complet sur leur machine.

Vous serez peut-être surpris d'apprendre que la distribution Slackware est encore activement maintenue et qu'elle se porte bien. Elle a fêté ses vingt-cinq ans le 16 juillet 2018.

### URL **Slackware Linux**

► <http://www.slackware.com>