

Kiki Novak

Débuter avec **LINUX**

Maîtrisez votre système
aux petits oignons



EYROLLES

Débuter avec LINUX

UN OUVRAGE DE RÉFÉRENCE POUR LES (FUTURS) LINUXIENS

Sans doute avez-vous déjà entendu parler de Linux. Peut-être avez-vous tenté de l'installer chez vous. Peut-être même l'utilisez-vous déjà et vous êtes-vous déjà rendu compte que tout ce que vous faisiez, tant bien que mal, avec votre système Windows, vous le faites mieux avec un système Linux.

Cet ouvrage va vous permettre de démarrer pour de bon avec Linux, en partant des fondamentaux et en progressant au fur et à mesure, jusqu'à devenir un pro – ou presque – de la ligne de commande. Dans le détail, vous découvrirez Linux et les logiciels libres par la pratique, vous installerez Linux et prendrez en main son interface graphique, avant d'administrer le système d'exploitation de A à Z. Vous configurerez enfin le poste de travail « aux petits oignons ».

UNE VIE APRÈS WINDOWS

Marre des virus et des antivirus ? Marre des systèmes qui rament et qui plantent ? Des mises à jour forcées ? Des publicités intempestives ? Des logiciels qui coûtent une fortune ? Des systèmes qui vous espionnent ? Vous tenez entre les mains l'ouvrage qui va vous permettre de débiter pas à pas avec Linux en suivant nos conseils et recommandations.

À qui s'adresse cet ouvrage ?

- Aux amateurs de logiciel libre et adeptes de Linux souhaitant maîtriser leur système au quotidien
- Aux étudiants et professionnels devant passer à Linux sans peine
- Aux utilisateurs Windows qui cherchent une alternative robuste, libre et gratuite
- À tous ceux qui veulent donner une seconde vie à leur vieil ordinateur, souvent incapable de faire tourner Windows correctement

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 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.

Sommaire

- Si Linux m'était conté
- Avant de mettre la main à la pâte
- Linux installé par une poule
- Premier contact avec le plan de travail
- 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 systèmes de fichiers
- Gérer les archives compressées
- Premiers pas sur le réseau
- Installer et gérer des logiciels
- Manipuler le chargeur de démarrage et le noyau
- Configurer le serveur graphique X.Org
- Linux installé par un chef
- Xfce bien mitonné
- Une sélection d'applications aux petits oignons
- Faites bonne impression sous Linux
- Larguez les amarres grâce au Wi-Fi
- Confectionner une clé USB d'installation
- Les différentes procédures de partitionnement
- Installer Linux sur un ordinateur avec EFI

Débuter avec LINUX

DANS LA MÊME COLLECTION

- P. MARTIN, J. PAULI, C. PIERRE DE GEYER. – **PHP 7 avancé.**
N°14357, 2016, 732 pages.
- L. BLOCH, C. WOLFHUGEL, A. KOKOS, G. BILLOIS, A. SOULLIÉ, T. DEBIZE. – **Sécurité informatique.**
N°11849, 5^e édition, 2016, 648 pages.
- R. GOETTER. – **CSS 3 Flexbox.**
N°14363, 2016, 152 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.
- B. PHILIBERT. – **Bootstrap 3 : le framework 100 % web design.**
N°14132, 2015, 318 pages.
- C. CAMIN. – **Développer avec Symfony2.**
N°14131, 2015, 474 pages.
- S. PITTION, B. SIEBMAN. – **Applications mobiles avec Cordova et PhoneGap.**
N°14052, 2015, 184 pages.
- H. GIRADEL, R. GOETTER. – **CSS 3 : pratique du design web.**
N°14023, 2015, 372 pages.
- C. DELANNOY. – **Le guide complet du langage C.**
N°14012, 2014, 844 pages.
- K. AYARI. – **Scripting avancé avec Windows PowerShell.**
N°13788, 2013, 358 pages.
- W. BORIES, O. MIRIAL, S. PAPP. – **Déploiement et migration Windows 8.**
N°13645, 2013, 480 pages.
- W. BORIES, A. LAACHIR, D. THIBLEMONT, P. LAFEIL, F.-X. VITRANT. – **Virtualisation du poste de travail Windows 7 et 8 avec Windows Server 2012.**
N°13644, 2013, 218 pages.
- J.-M. DEFRANCE. – **jQuery-Ajax avec PHP.**
N°13720, 4^e édition, 2013, 488 pages.

SUR LE MÊME THÈME

- C. BLAESS. – **Développement système sous Linux.**
N°14207, 4^e édition, 2016, 1040 pages.
- R. HERTZOG, R. MAS. – **Debian Jessie.**
N°14203, 2016, 538 pages.
- R. HERTZOG, R. MAS. – **Debian Wheezy.**
N°13799, 2014, 516 pages.
- J.-F. BOUCHAUDY. – **Linux Administration – Tome 1.**
N°14082, 3^e édition, 2014, 690 pages.
- C. BLAESS. – **Scripts Shell Linux et Unix.**
N°13579, 2^e édition, 2012, 296 pages.

Kiki Novak

Débuter avec LINUX

Maîtrisez votre système aux petits oignons

EYROLLES

The logo for EYROLLES features the word "EYROLLES" in a bold, sans-serif font. Below the text is a horizontal line with a small grey circle centered underneath it.

ÉDITIONS EYROLLES
61, bd Saint-Germain
75240 Paris Cedex 05
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.

© Groupe Eyrolles, 2017, ISBN : 978-2-212-13793-4

À la mémoire d'Albert Bruc...

The White Rabbit put on his spectacles. « Where shall I begin, please your Majesty ? » he asked.

« Begin at the beginning », the King said, very gravely, « and go on till you come to the end : then stop. »¹

Lewis Carroll, *Alice in Wonderland*

Wir werden sie ausführlich erzählen, genau und gründlich, – denn wann wäre je die Kurz- oder Langweiligkeit einer Geschichte abhängig gewesen von dem Raum und der Zeit, die sie in Anspruch nahm ? Ohne Furcht vor dem Odium der Peinlichkeit, neigen wir vielmehr der Ansicht zu, daß nur das Gründliche wahrhaft unterhaltend ist.²

Thomas Mann, *Der Zauberberg*

*Grau, lieber Freund, ist alle Theorie,
Und grün des Lebens goldner Baum...³*

Goethe, *Faust I*

-
1. *Le Lapin Blanc mit ses lunettes. « Plaise à Votre Majesté, où dois-je commencer ? » demanda-t-il. « Commencez au commencement », dit le Roi d'un ton grave, « et continuez jusqu'à ce que vous arriviez à la fin ; ensuite, arrêtez-vous. »* Lewis Carroll, *Alice au pays des merveilles* (trad. H. Bué, 1972).
 2. *Nous la raconterons en détail, exactement et minutieusement. En effet, l'intérêt d'une histoire ou l'ennui qu'elle nous cause ont-ils jamais dépendu de l'espace et du temps qu'elle a exigé ? Sans craindre de nous exposer au reproche d'avoir été méticuleux à l'excès, nous inclinons au contraire à penser que seul est vraiment divertissant ce qui est minutieusement élaboré.* Thomas Mann, *La montagne magique* (trad. M. Betz, Fayard, 1981).
 3. *Grise, cher ami, est toute théorie / Et vert l'arbre doré de la vie.* Goethe, *Faust I* (trad. Jean Malaplate, GF, 1984).

Avant-propos

Le présent ouvrage s'adresse en premier lieu à tous ceux qui utilisent l'informatique au quotidien et qui cherchent une alternative...

Linux, un système écologique et performant

Vous rappelez-vous ce qu'étaient les Trabant, ces petites voitures en plastique produites par millions en ex-RDA, censées révolutionner le confort routier non seulement de l'ouvrier est-allemand, mais plus généralement des habitants de tous les pays membres du pacte de Varsovie vers lesquels on les exportait massivement ? Propulsée par un moteur à deux temps monté dans une carrosserie en résine artificielle, la Trabant peuplait les routes est-allemandes, hongroises, tchèques et polonaises.

Les gens qui roulaient en Trabant n'avaient pas vraiment le choix ; ils n'avaient pas opté pour ce modèle par conviction. La Trabant tenait aussi peu la route qu'une Lada, s'avérait tout aussi gloutonne que polluante et bruyante et, au vu de ses performances routières modestes, le coût d'achat et d'entretien était relativement élevé.

CLIN D'ŒIL

Mon grand-père en Hongrie roulait en Trabant. Il en était fier comme un pape et je me souviens encore du jour où il m'a copieusement fâché parce qu'en m'allumant une Gauloise sans trop faire attention, j'ai réussi à faire un trou dans la portière en phénoplaste en la brûlant.

Imaginez donc un monde où tout le monde roulerait en Trabant. Et imaginez que, dans ce même monde, les Clio, 206, Jaguar XS et autres Porsche 911 seraient *gratuites*. Elles pousseraient sur les arbres et aussi dans les champs. Il suffirait de les déterrer ou de les cueillir. Les Clio et les 206 seraient prêtes à démarrer, clé en main ou presque, et consommeraient moins de trois litres au cent. Quant aux Jaguar et aux Porsche, elles viendraient en pièces détachées. Quiconque s'aventurerait à récolter et assembler les pièces aurait une belle surprise, car il se

Figure 1

Et si on roulait tous en Trabant ?



retrouverait avec un bolide distillant de belles sensations, un troupeau de deux cents chevaux sous les fesses, consommant moins d'un litre sur cent kilomètres. Un litre d'huile d'olive.

Belle utopie, direz-vous. Trop belle pour être vraie. Et pourtant, il est un domaine dans notre quotidien à tous, où dix-neuf personnes sur vingt « roulent en Trabant » et pensent qu'elles n'ont pas le choix parce qu'elles ignorent l'existence des autres Clio, 206 et Porsche, qui poussent à côté. Je parle de l'informatique.

Dans notre monde imaginaire, les conducteurs de Trabant regarderaient d'un œil mauvais ces Jaguar qui poussent dans les champs. Forcément, si ça pousse dans les champs, ce n'est pas sérieux ; si ça ne coûte rien, ça ne vaut rien. Et bien évidemment, si ça consomme peu, les performances doivent être décevantes, voire carrément minables. Le conducteur de Trabant est plein de mépris envers le conducteur de Jaguar, mépris qui se mue en rancœur lorsqu'il apprend que les Jaguar tombent très rarement en panne – en tout cas, pas aussi régulièrement que sa Trabant, mais bon. Il doit certainement s'agir d'un mythe, car n'oublions pas que si la Trabant est la voiture la plus populaire, c'est qu'il doit bien y avoir une bonne raison à cela... Or tout cela a lieu dans notre monde, dans le domaine de l'informatique...

En informatique, la Trabant serait le système d'exploitation Microsoft Windows que les vendeurs vous obligent à acquérir lors de l'achat d'un PC neuf, en vous faisant croire que vous n'avez pas le choix. C'est cher ; vous payez la licence et les options sont toutes facturées à part. Ça ne tient pas la route, au nombre élevé de plantages, et c'est vulnérable à toutes sortes d'attaques de virus et autres vers. C'est gourmand en ressources, ce qui vous oblige à investir dans du matériel suffisamment performant pour avoir un système fonctionnel. Ça vous espionne de partout, sans le moindre respect pour la confidentialité de vos données. Et c'est polluant, parce que votre vieux matériel aux performances désormais trop modestes se retrouve tout bonnement relégué à la casse.

Pourquoi ce livre ? Et pour qui ?

Le présent ouvrage s'adresse donc à tous les conducteurs de Trabant qui aimeraient bien savoir ce que peuvent valoir ces Jaguar et ces Porsche qui poussent sur les arbres. **Cette alternative existe : elle s'appelle Linux.**

Sans doute avez-vous déjà entendu parler de Linux. Peut-être avez-vous tenté de l'installer chez vous. Peut-être même l'utilisez-vous déjà et vous êtes-vous déjà rendu compte que tout ce que vous faisiez, tant bien que mal, avec votre système Windows, vous le faites mieux avec un système Linux. Sur votre vieux PC, sans être obligé d'acheter du matériel plus performant. Gratuitement, sans être obligé de vous procurer une licence pour le système, ni pour les logiciels. Et en toute liberté, en l'installant sur autant de machines qu'il vous chante et autant de fois que vous le voulez... Et sans que votre propre système vous espionne derrière votre dos !

Il existe déjà des livres sur Linux et il est même réjouissant de voir que les publications sur ce sujet se multiplient. Plutôt que réinventer la roue, j'ai choisi pour cet ouvrage une approche pragmatique : **vous faire découvrir Linux en plongeant les mains dans le cambouis.**

Aucune connaissance, aucune compétence spécifique n'est présumée de votre côté. Si vous êtes muni d'une dose saine de curiosité et d'un certain plaisir à expérimenter, considérez-vous comme le lecteur idéal de ces pages.

PÉDAGOGIE **Les mains dans le cambouis**

Si nous nous adressons certes à tous les débutants, nous n'hésiterons pas cependant à sortir des interfaces fenêtrées et à vous présenter petit à petit ce qui tourne sous le capot de votre système... afin de satisfaire l'appétit des plus aguerris !

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 **premier chapitre** 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 **deuxième chapitre** introduit Slackware Linux, le système (ou la distribution) utilisée pour mettre en pratique les exemples de ce livre, ainsi que des considérations sur le choix du matériel informatique sur lequel vous pourrez installer Linux.

Dans le **troisième chapitre**, 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 étant que vous disposiez d'un système fonctionnel, qui vous permette de découvrir Linux par la pratique, chez vous, sur votre ordinateur.

Le **chapitre 4** est une sorte de visite guidée de votre nouveau système, un peu comme si vous vous installiez dans une voiture que vous venez d'acheter. Où sont les pédales ? Où sont les leviers ? Comment fait-on pour changer les vitesses ? Et où sont donc passés les clignotants ? Ceux d'entre vous qui ont l'habitude d'un autre système d'exploitation découvriront probablement une série de similitudes rassurantes... et quelques différences vaguement inquiétantes. Ce chapitre vous prend par la main et vous montre ce qu'il y a de nouveau.

Les **chapitres 5 à 17** constituent sans aucun doute le « plat de résistance » de l'ouvrage. Ils vous introduisent progressivement, petit à petit, à l'administration d'un système Linux, en commençant par des opérations de base (**chapitre 5**) pour vous amener petit à petit à réaliser des opérations plus avancées comme la gestion des utilisateurs (**chapitre 6**) et des droits d'accès (**chapitre 7**), la recherche de fichiers (**chapitre 8**), la création et la manipulation de liens (**chapitre 9**), la gestion des processus (**chapitre 10**) et des services (**chapitre 11**), l'accès aux périphériques (**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 le **chapitre 17** nous apprend à configurer le serveur graphique.

Les **chapitres 18 à 20** décrivent l'installation d'un système Linux taillé sur mesure et configuré « aux petits oignons », en utilisant les compétences acquises dans les chapitres précédents. Une fois que le système de base est mis en place au **chapitre 18**, le **chapitre 19** est entièrement consacré au peaufinage de l'environnement de bureau, pour le rendre plus esthétique et plus fonctionnel. Le **chapitre 20** passe en revue et en détail l'installation et la configuration d'une série d'applications populaires pour être plus productif dans les domaines de l'Internet, de la bureautique, du graphisme et du multimédia : tout ce qu'il vous faut pour un poste de travail complet.

Pour le dessert, les **chapitres 21 et 22** abordent deux points un peu plus délicats, le système d'impression et la configuration d'un réseau sans fil. À ce stade, en effet, le lecteur a acquis suffisamment de compétences pour s'attaquer à des tâches techniquement plus exigeantes.

Et s'il vous reste un petit creux, vous trouverez **en annexe** des informations supplémentaires pour l'installation de Linux sur du matériel plus récent : clé USB, partitionnement GPT et installation UEFI.

À propos de cette édition

Débuter avec Linux est une réécriture complète de mon ouvrage *Linux aux petits oignons* paru chez Eyrolles en 2009. Cette première version de mes cours sur les bases de l'administration Linux était basée sur CentOS 5.3. Si je ne me suis pas contenté d'une simple mise à jour de ce premier livre, c'est pour plusieurs raisons.

Plus de sept ans se sont écoulés depuis la parution de *Linux aux petits oignons*, sept ans durant lesquels le monde de Linux a bien changé. Je me suis retrouvé confronté à un dilemme quant au choix de la distribution. CentOS 7.3 n'a plus grand-chose en commun avec l'ancêtre 5.3 désormais obsolète. Certes, CentOS reste un excellent choix, et ce n'est pas un hasard si cette

distribution est toujours aussi répandue en entreprise. Malheureusement, elle a pour ainsi dire perdu un certain nombre de vertus pédagogiques en cours de route. Non content de cela, la version 7 est bien plus gourmande en ressources que les versions précédentes, et les processeurs 32-bits ne sont plus supportés.

J'ai donc décidé de me baser dorénavant sur Slackware Linux, une distribution « brute de décoffrage » bien plus adaptée en simples termes pédagogiques, même si elle ne jouit pas d'une popularité excessive. Un bénéfice majeur de ce choix, c'est que je ne pénalise pas mes nombreux lecteurs qui possèdent du vieux matériel, car Slackware est idéal pour refaire une jeunesse au matériel obsolète. Pas la peine de casser la tirelire pour apprendre les bases de Linux.

Enfin, les chapitres ajoutés ou réécrits depuis le premier ouvrage sont le fruit de quelques années de travail de terrain. Ils ont été élaborés et peaufinés dans le cadre des formations Linux avec mes stagiaires, en essayant de fournir des réponses à leurs nombreuses questions.

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 et les hectolitres de café espresso.

À propos de l'auteur

Féru d'informatique depuis les jours du processeur monoplatine 8080 et du Commodore VC-20, j'ai découvert le monde de GNU/Linux relativement tard. En 2001, je développais du PHP pour un éditeur et regrettais de passer plus de temps derrière ma machine que devant. « T'as qu'à installer Linux », me suggéra alors un message posté sur un forum. Quelques jours plus tard, j'avais remplacé mon système propriétaire par Slackware Linux dans sa version 7.1 et j'étais inscrit sur la liste de diffusion de basiclinux.net, un groupe de vétérans chevronnés qui prodiguaient gratuitement des cours détaillés sur l'administration d'un système Linux, mais qui a malheureusement cessé toute activité.

Établi dans le Gard à mi-chemin entre Nîmes et les Cévennes, j'ai créé ma propre entreprise en 2009. Microlinux est spécialisée dans l'élaboration de solutions informatiques durables basées à cent pour cent 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. En dehors de mes publications chez Eyrolles, j'ai écrit de nombreux articles pour la presse écrite, notamment *Linux Pratique* et *Planète Linux*.

► <http://www.microlinux.fr>

Table des matières

CHAPITRE 1

Si Linux m'était conté 1

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 19

Les systèmes BSD, une autre famille

d'Unix libres 20

La foire aux distributions 21

Quelle est la « meilleure »

distribution Linux ? 22

 La panoplie logicielle 22

 L'actualité des paquets 23

 Les outils d'administration 23

 Le caractère commercial 24

 La qualité « entreprise » 25

 La configuration par défaut 26

 La qualité de la documentation 27

Qui utilise Linux ? 27

CHAPITRE 2

Avant de mettre la main

à la pâte 29

Petite introduction culinaire 29

Se former à Linux avec Slackware 31

 Une distribution robuste sans fioritures ... 31

Le matériel : usine à gaz ou simple gazinière ? . 34

 Quelques scénarios à éviter 34

 Un PC pour Linux, mais lequel ? 35

Soyez branché ! 37

Comment obtenir la distribution

Slackware ? 38

 Utiliser un client BitTorrent pour récupérer

 les images ISO 40

 Acheter un support d'installation

 Slackware ? 40

 Graver les fichiers ISO 40

 Et si je n'ai pas de lecteur optique ? 41

CHAPITRE 3

Linux installé par une poule.... 43

L'œuf ou la poule ? 43

Au commencement était le BIOS 44

Testez votre mémoire ! 46

Installation de Slackware 48

 Premiers pas 48

 Choisir la disposition du clavier 49

 Se connecter en tant que root 50

 Partitionner le disque dur 51

Choisir le schéma de partitionnement 51

Partitionner le disque dur avec fdisk 52

 Lancer l'installateur Slackware 57

 Formater la partition d'échange 58

 Formater la partition principale 59

 Décider ce que l'on souhaite installer 61

 Configurer et installer le chargeur

 de démarrage 63

Configurer la souris	66
Configurer le réseau	67
Configurer les services au démarrage et la police de la console	69
Configurer l'horloge et le fuseau horaire . .	70
Choisir le gestionnaire de fenêtres	70
Définir un mot de passe root	71
Fin de l'installation et redémarrage initial .	72
Configuration post-installation sommaire .	74
Définir les utilisateurs de la machine	74
Derniers réglages et lancement de Xfce . . .	77

CHAPITRE 4

Premier contact

avec le plan de travail 79

Découvrir le bureau Xfce	79
Ouvrir et fermer une session	79
Premières impressions	80
Quelques retouches initiales	81
Le menu Applications	85
Repérer les applications de base	86
Manipuler les fenêtres	86
Le navigateur de fichiers Thunar	88
Configuration sommaire de Thunar	88
Où suis-je ?	90
Une structure en arborescence	90
Naviguer dans les répertoires	90
Home sweet home	91
Naviguer avec les boutons	92
Visite guidée du système en dix minutes . .	93
Remonter à la racine : /	93
Les répertoires /bin et /boot	93
Les répertoires /dev et /etc	95
Le répertoire /lib	96
Les répertoires /mnt, /media et /run	97
Les répertoires /proc et /sys	98
Les répertoires /root et /sbin	98
Le répertoire /usr	99
Les répertoires /tmp et /var	101
La nausée devant l'abondance ?	101

CHAPITRE 5

Linux en mode texte :

consolez-vous ! 103

Introduction à la ligne de commande	103
Faut-il avoir peur du mode texte ?	103
Console ? Terminal ? Et quoi encore ? . . .	105
<i>Jouer avec les consoles virtuelles</i>	<i>105</i>
<i>Utiliser un émulateur de terminal</i>	<i>106</i>
L'émulateur de terminal Xfce Terminal . . .	106
<i>Configurer Terminal pour une utilisation</i> <i>agréable</i>	<i>107</i>
<i>Premiers pas en ligne de commande</i>	<i>110</i>
Naviguer : ls, pwd et cd	111
Afficher le contenu d'un répertoire avec ls .	111
Décrypter les résultats de votre ordinateur	111
<i>Qu'est-ce qui est quoi là-dedans ?</i>	<i>111</i>
<i>Mais encore ?</i>	<i>113</i>
<i>Humain, pas trop humain ?</i>	<i>113</i>
<i>Splendeur et misère des fichiers cachés . . .</i>	<i>114</i>
<i>Cachez cette configuration que je ne saurais</i> <i>voir</i>	<i>115</i>
Afficher les informations détaillées d'un répertoire	116
pwd : « Vous êtes ici ! »	117
On bouge avec cd !	117
<i>Chemin relatif ou absolu ?</i>	<i>117</i>
<i>À court d'arguments</i>	<i>118</i>
« Ici » et « à l'étage »	118
Deux commandes de sortie simples :	
echo et cat	122
echo : afficher une ligne de texte	122
cat : afficher et concaténer	122
Visualiser : more et less	124
Gérer l'affichage de fichiers longs	124
Visualiser avec more	125
Less is more : moins, c'est plus !	126
Créer : touch et mkdir	127
Modifier l'horodatage d'un fichier avec touch	127
Créer un fichier vide avec touch	128

Une autre approche : la notation numérique	189	À quoi servent les liens symboliques ?	215
Attribuer tous les droits à tout le monde	190	Lequel est le vrai ?	216
Retirer et ajouter des droits	190	Les liens symboliques en mode graphique	217
Les permissions par défaut : umask	191	Casser un lien symbolique	218
Gérer les droits d'accès des répertoires	193	Effacer un lien symbolique	219
Changer le propriétaire et le groupe		Les permissions des liens symboliques	220
d'un fichier	196	Les liens physiques	220
Pas de cadeaux !	197	Un nom de fichier alternatif	220
		Un lien physique a la vie dure	221
CHAPITRE 8		CHAPITRE 10	
Chercher une aiguille		Gérer les processus 223	
dans une botte de foin..... 199		Linux, un système multitâche	223
Chercher l'emplacement d'un fichier		Lancer et surveiller un processus : ps	224
dont on connaît le nom	199	Afficher les processus en cours	225
limiter la recherche à certains répertoires	200	Compter les processus en cours	226
Zone interdite	200	Envoyer un signal à un processus : kill	227
Faire fi des avertissements	201	Arrêter et relancer un processus	228
Interrompre une recherche	202	Terminer et tuer un processus	228
Chercher des fichiers dont on ne connaît		S'adresser à un processus par son nom :	
pas le nom exact	202	killall	230
Chercher selon d'autres critères		Utiliser kill ou killall ?	231
que le nom	203	Les processus dans tous leurs états	232
Chercher en fonction de la taille	203	Un lien de parenté : pstree	232
Appliquer une commande sur		Massacre familial	233
les fichiers trouvés	204	Tirer à vue sur tout ce qui bouge ?	235
Chercher par type	204	Surveiller l'utilisation des ressources	
Chercher selon les droits d'accès	205	de la machine : top	235
Cas pratique : attribuer des permissions			
à un ensemble de fichiers	206	CHAPITRE 11	
Chercher du texte à l'intérieur d'un fichier	207	Gérer les services..... 237	
Chercher du texte dans une série		Une confusion babylonienne	237
de fichiers	209	Qu'est-ce qu'un service ?	238
Affiner la recherche	210	Arrêter et (re)démarrer un service	238
Quelques applications pratiques :		Activer un service au démarrage	
obtenir des informations sur son matériel	211	de la machine	239
		Les services au démarrage et les niveaux	
		d'exécution	242
		Deux niveaux d'exécution	
		un peu spéciaux : 0 et 6	246
		Éteindre et redémarrer la machine	247
CHAPITRE 9			
Créer et manipuler			
des liens 213			
Les liens symboliques	213		
Créer des liens symboliques	214		

CHAPITRE 12

Accéder aux systèmes de fichiers 249

- La gestion des périphériques amovibles en mode graphique 249
 - Insérer un DVD ou un CD 250
 - Insérer une clé USB 251
 - Première conclusion 252
 - Petit retour à l'âge de pierre 252
- Les fichiers de périphérique 253
- La notion de montage/démontage 254
 - Monter manuellement un CD ou un DVD 255
 - Démonter le CD ou le DVD 256
 - Monter et démonter manuellement une clé USB 256
 - Scier la branche sur laquelle on est assis .. 259
 - Le blocage des lecteurs 259
- Éjecter un DVD ou un CD-Rom : eject . 260
- Obtenir des informations sur les périphériques montés 260

CHAPITRE 13

Gérer les archives compressées..... 263

- Les différents formats d'archivage 263
- Compresser et décompresser un fichier .. 264
 - Compresser et décompresser un fichier avec gzip 264
 - Compresser et décompresser un fichier avec bzip2 265
- Manipuler les archives avec tar 266
 - Créer une archive avec tar 266
 - Extraire les fichiers d'une archive 267
 - Combiner l'archivage et la compression .. 267
 - Extraire une archive compressée 268

CHAPITRE 14

Premiers pas sur le réseau.. 269

- Une grande confusion 269
- Prérequis matériel et logiciel 270

- Le modem routeur ADSL 270
- Questions de câblage 271
- La carte Ethernet 272
- Les pilotes sous Linux 273

Principe de fonctionnement des réseaux . . 274

- Afficher la configuration des interfaces réseau : ifconfig 274
 - L'adresse MAC de votre carte* 275
 - L'adresse IP et le masque de sous-réseau* . . 276
- Les adresses IP 276
 - Un peu de pratique* 276
 - Un peu de théorie* 277

Établir un contact avec une machine distante : ping 278**Le routeur : un centre de tri pour paquets numériques 279**

- Sous les pavés numériques, la plage d'adresses IP privées* 280
- Relier le public et le privé* 280
- Le système de noms de domaine : l'annuaire d'Internet* 280

Configurer une connexion à Internet . . . 281

- Configuration dynamique : DHCP 281
 - Paramétrer le serveur DHCP intégré dans le modem routeur* 283
 - La configuration DHCP en détail* 284
- Configuration statique 285
 - Renseigner les serveurs DNS* 287

Diagnostiquer le réseau en cas de panne . . 288**Et comment faire avec plusieurs PC ? . . . 289**

CHAPITRE 15

Installer et gérer des logiciels 291

- Petit aperçu pour les utilisateurs de Windows 291
- Anatomie d'un paquet Slackware 292
- Gérer les paquets Slackware avec les outils traditionnels 293
 - Installer un paquet 294
 - Vérifier si un paquet est installé 295

Supprimer un paquet	296		
Mettre à jour un paquet	296		
En savoir plus sur le contenu d'un paquet	299		
Gérer les paquets Slackware avec slackpkg	299		
Configuration initiale	300		
Installer des paquets	301		
Supprimer des paquets	302		
Mettre à jour des paquets	302		
Rechercher des paquets ou des fichiers individuels	303		
Deux options fort pratiques	304		
Recompiler des paquets officiels	305		
Fabriquer un paquet à partir du code source	305		
Modifier un paquet officiel	307		
Compiler des paquets tiers depuis les sources	308		
Construire un paquet à partir d'un script de SlackBuilds.org	309		
Gérer les dépendances des paquets	312		
Gérer les paquets tiers confortablement avec sbopkg	312		
<i>Installer sbopkg</i>	313		
<i>Utiliser sbopkg</i>	313		
Installer des paquets tiers précompilés	315		
Les paquets fournis par Eric Hameleers	315		
Autres ressources de paquets binaires	316		
Installer les paquets multilib pour Slackware64	316		
Installer l'outil slackpkg+	317		
Configurer slackpkg+ pour l'utilisation du dépôt multilib	317		
Installer les paquets multilib	318		
CHAPITRE 16			
Manipuler le chargeur de démarrage et le noyau... 319			
Le chargeur de démarrage LILO	319		
Modifier le délai d'attente au démarrage	321		
		Options globales et paramètres de démarrage	322
		Modifier la résolution de la console	323
		Récupérer un système qui ne démarre plus	325
		Configurer le système et le noyau à charger	327
		Basculer vers le noyau GENERIC	328
		Le noyau GENERIC et les mises à jour	331
CHAPITRE 17			
Configurer le serveur graphique X.Org 335			
		Le système X Window et X.Org	335
		Quelques préparatifs	336
		Le gestionnaire de fenêtres	
		WindowMaker	337
		Définir la disposition du clavier	338
		Personnaliser l'aspect du terminal XTerm	340
		Quel pilote pour quelle carte graphique ?	340
		Tester l'accélération graphique	342
		Corriger la résolution avec le framebuffer activé	343
		Corriger le comportement du pavé tactile	344
		Utiliser le pilote propriétaire Nvidia	345
		Utiliser le pilote propriétaire AMD/ATi	349
		Le gestionnaire de fenêtres Fluxbox	350
CHAPITRE 18			
Linux installé par un chef ... 353			
		Les options de démarrage	353
		Partitionnement et formatage	354
		Installer et configurer Slackware	359
		Configuration post-installation	360
		Créer un utilisateur	360
		Peaufiner l'éditeur Vim	360
		Personnaliser l'invite de commande	362
		<i>Avec un peu de couleur, c'est encore mieux</i>	364
		<i>Quelques alias pratiques pour la console</i>	365
		<i>Définir Vim comme l'éditeur principal</i>	366

<i>Compléter le PATH pour les utilisateurs</i>	366	Supprimer les icônes du bureau	392
Effectuer la mise à jour initiale	366	Choisir un économiseur d'écran	393
Basculer vers le noyau GENERIC et peaufiner LILO	368		
Configurer et personnaliser le serveur graphique X.Org	370		
Définir le français comme langue principale du système	371		
Installer et configurer sbopkg	372		
CHAPITRE 19		CHAPITRE 20	
Xfce bien mitonné	373	Une sélection d'applications aux petits oignons	395
Installer le jeu d'icônes Elementary-Xfce	373	Naviguer sur Internet avec	
Installer le thème GTK Clearlooks Phenix	374	Mozilla Firefox	395
Installer une police mieux adaptée à l'interface	376	Franciser Firefox	396
Installer une collection de fonds d'écran	376	Se documenter	399
Remplacer le menu Applications par Whisker Menu	378	Installer le plug-in Macromedia Flash	399
Remplacer le tableau de bord inférieur par le dock Plank	382	Installer le module Adblock Plus	400
Installer Plank	383	Gérer le courrier électronique	
Tester Plank et configurer le lancement automatique	383	avec Mozilla Thunderbird	401
Peaufiner le compositeur d'affichage	385	Franciser Thunderbird	402
Configurer Plank	386	Configurer un compte mail	404
Utiliser Plank au quotidien	386	Configurer le format d'envoi des courriels	405
Remplacer le gestionnaire de connexion		Le bon grain et l'ivraie : filtrer ses courriels	406
XDM par LXDM	387	L'alternative Seamonkey	407
Installer LXDM	387	Franciser Seamonkey	408
Configurer LXDM	389	Utiliser Seamonkey	408
<i>Activer le verrouillage numérique</i>	389	Utiliser des outils bureautiques	409
<i>Relancer le serveur graphique à la déconnexion</i>	389	La suite bureautique LibreOffice	409
<i>Définir un fond d'écran personnalisé</i>	390	<i>Installer JDK</i>	410
<i>Désactiver le sélecteur de langues</i>	390	<i>Installer LibreOffice</i>	413
<i>Désactiver l'affichage des utilisateurs</i>	390	<i>Désactiver l'insertion automatique</i>	416
<i>Corriger la couleur de la police d'affichage</i>	391	<i>Supprimer les polices exotiques</i>	416
Touches finales	392	<i>Installer les jeux de polices Microsoft</i>	417
		<i>Apprendre à se servir de LibreOffice</i>	419
		Afficher des documents PDF avec Evince	420
		Éditer des textes simples avec Gedit	422
		Gérer les fichiers	423
		Rechercher des fichiers avec Recoll	423
		Gérer les archives compressées avec File Roller	425
		Graver des CD et des DVD avec Brasero	426
		Manipuler les images	427
		Visionner des images avec gThumb	427

Retoucher des photos et dessiner avec Gimp	428	Numériser des documents avec XSane	465
Gérer les graphismes vectoriels avec Inkscape	430	Numériser des documents avec Simple Scan	466
Faire de la PAO avec Scribus	431		
Manipuler les fichiers audio	432	CHAPITRE 22	
Lire des fichiers et des flux audio		Larguez les amarres	
avec Audacious	432	grâce au Wi-Fi	469
Extraire des CD audio avec Asunder	434	Le point d'accès Wi-Fi	469
Éditer des fichiers audio avec Audacity	435	Configurer le Wi-Fi sur un ordinateur	
Manipuler les fichiers vidéo	437	portable	472
Regarder des fichiers vidéo avec MPlayer	437	Gérer les connexions réseau	
<i>Utiliser MPlayer</i>	440	avec NetworkManager	473
<i>Quelques astuces</i>	442	Ajouter le Wi-Fi à une station de travail	477
<i>Une interface graphique pour MPlayer ?</i>	443		
Regarder des fichiers vidéo		ANNEXE A	
et des DVD avec VLC	443	Confectionner une clé USB	
Télécharger des torrents		d'installation	479
avec Transmission	444		
Aller plus loin	446	ANNEXE B	
		Les différentes procédures	
CHAPITRE 21		de partitionnement	483
Faites bonne impression		MBR et GPT	483
sous Linux	447	Master Boot Record (MBR)	484
Imprimer avec CUPS	447	GUID Partition Tables (GPT)	484
Activer et démarrer le serveur CUPS	448	Migrer vers GPT	485
Votre imprimante et Linux	448	Revenir au partitionnement MBR	486
Configurer une imprimante		ANNEXE C	
HP Officejet Pro 8600	450	Installer Slackware	
Administrer CUPS	455	sur un ordinateur avec EFI..	487
Configurer une imprimante mal prise		EFI et BIOS	487
en charge	456	Un exemple pratique : l'Asus S300	488
<i>Configurer une imprimante</i>		Configuration du BIOS et démarrage	488
<i>Brother HL-2140</i>	456	Partitionnement	488
<i>Configurer une imprimante</i>		Le chargeur de démarrage GRUB	489
<i>Canon PLXMA MG2250</i>	459	Peaufiner l'aspect de GRUB	490
<i>Configurer une imprimante Canon</i>			
<i>PLXMA iP4200</i>	461	Index	493
Numériser des documents	465		

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^e 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*), c'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 Bell¹, 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 grand-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 retrouver 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 viendrait où les systèmes et les applications se vendraient *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 serait conséquent, les gens seraient obligés d'acheter les boîtes avec le matériel et les ventes feraient de vous l'homme le plus riche de la terre.

On vous prend probablement pour un fou.

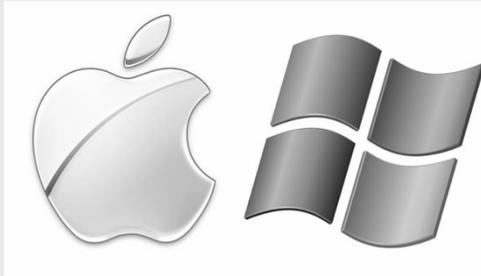


Figure 1-2 À quoi ressemblait l'informatique en 1970, avant Microsoft et Apple ?

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 *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 résoudre ce genre de problème : 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 driver ou « pilote » de périphérique, c'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 puissent étudier le problème à sa place et mettre 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 à 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.

📖 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 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 ne contient aucune ligne de code d'Unix.

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

► <http://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 gogo »). Selon la définition proposée par Richard, un logiciel est libre s'il respecte les quatre conditions fondamentales :

- la liberté d'utiliser le logiciel ;
- la liberté de le copier ;
- la liberté d'en étudier le fonctionnement ;
- la 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^{er} 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.

Figure 1-5

À ce jour, le noyau GNU/Hurd n'est toujours pas finalisé. La version 0.8 a été publiée le 18 mai 2016.



Linus Torvalds et le noyau Linux

Début janvier 1991, de l'autre côté de l'Atlantique, en Finlande, le jeune étudiant en informatique 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 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 de 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*.

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.

L'AIR DU TEMPS Les ordinateurs personnels

Si vous êtes né vers la fin des années soixante 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 – mais 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.

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.

► <http://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.

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 sécher ses cours 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.

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>

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` 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 de 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 :

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 intéressés, 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

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 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 qui permettra à Linus de s'acquitter 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 de Helsinki. Il est impressionné par la présentation de Stallman et prend une décision d'une importance capitale pour la suite du projet.