

MUFFIN

*Magazine des Utilisateurs Francophones de Fedora
Imparfait mais Nécessaire*



fedora
project

Edito	3
La philosophie du Libre	5
Présentation de Fedora	8
Utilisation de la ligne de commande	10
Utilisation de YUM : Les premiers pas	15

Édito

Bienvenue à vous, nouveau lecteur de MUFFIN, le magazine de utilisateurs francophones de Fedora, Imparfait mais Nécessaire.

Ce premier numéro fut long à faire, mais c'est avec plaisir que nous vous le présentons aujourd'hui.

Imparfait ce numéro l'est sûrement. Je ne connais pas de magazine qui soit parfait et encore moins lors de son premier numéro. L'équipe de rédaction est encore jeune et il est certain que des erreurs se sont glissées tout le long de ce numéro.

Nécessaire ce magazine l'est obligatoirement. Il l'est, en mon sens, pour plusieurs raisons.

* La première est le développement et la promotion du logiciel libre. Si ce magazine est bien sûr orienté sur la distribution Fedora, une des premières missions du projet Fedora est la promotion et le développement du logiciel libre. Je suis surpris encore aujourd'hui quand je vois le nombre de personnes - y compris dans la communauté Fedora - qui n'appréhendent pas du tout les tenants et les aboutissants du logiciel Libre.

* La deuxième est le développement de la communauté Francophone de Fedora. Si le forum est très actif et le documentation bien fournie, j'ai, des fois, l'impression que la communauté ne réalise pas vraiment l'importance - et donc les responsabilités - qu'elle a. Promotion, traduction, tests sont autant de sous-projets du Fedora Project qui manquent de bonne volonté. Ceci est d'autant plus important que la communauté francophone est la la première communauté Fedora indépendante du projet.

* Troisième et dernière raison, la promotion du Projet Fedora. La promotion du projet doit se faire à deux niveaux. Au niveau interne, pour expliquer aux membres de la communauté comment le projet fonctionne et qu'elles sont les décisions qui ont été prises. Au niveau externe, pour promouvoir ce merveilleux projet. En effet la promotion de cette distribution reste souvent minime et souvent occultée par d'autres projets.

Ainsi, suite à la découverte de Magnum le magazine de Mandriva, je me suis lancé et pris la tête de ce fabuleux projet. Je suis fier du travail que nous avons réalisé et je tiens à remercier toute l'équipe de rédaction pour le travail qu'elle a fourni.

J'espère que vous prendrez autant de plaisir à lire ce magazine que nous en avons pris pour le rédiger.

Matthieu Gautier

L'équipe derrière MUFFIN

Rédacteur en chef

Matthieu Gautier

Auteurs

Matthieu Rondeau

Thomas Canniot

Valentin Guerlesquin

Relecteur

Fabien Archambault

Graphiste

Romain Dreux

Leading article

Welcome to you, dear new MUFFIN reader, the Fedora magazine for French speaking users. Why MUFFIN? In French, MUFFIN means "Magazine for the Users of Fedora who speaks French, Imperfect (the magazine... not the users), but Necessary (again, the magazine).

Even if it took a long time to make this first issue, it is a great pleasure to introduce MUFFIN to you today.

This issue is certainly imperfect. I don't know any perfect magazine, even less at their first release. The redaction team is pretty young and we surely missed some errors.

From my point of view, this magazine is more than necessary:

- * The first reason is the development and the promotion of Free and Open Source Software (FOSS). Even if MUFFIN is Fedora oriented, it sticks to the first goals of the Fedora Project: the promotion and development of FOSS. Even today, I'm surprised when I see the amount of people, even inside of the Fedora community, who doesn't get the ins and outs of FOSS.

- * The second reason is the development of the Fedora community. If the [French] forum is really active and the [French] documentation is pretty complete, I sometimes have the feeling that the community doesn't realize the importance - and the responsibility - it bears. Promotion, translation, quality assurance are so many sub-projects which lack so many people.

- * Third and last but not least reason, the Fedora Project promotion. It has to be achieved on two distinct levels. On the one hand, from an inside level, to explain the community members how the project works, what are the decisions that have been taken, and why. On the other hand, from an outside level, to promote this wonderful project as a whole. Indeed the promotion of Fedora as an operating system is most of the time minor and eclipsed by other projects.

So, after we discovered Mandriva magazine - i.e. Magnum, I involved myself into this project and I took its leadership. I am really proud of what we have achieved and I want to thank all the redaction team for the work they did.

I hope you will enjoy reading this magazine as much as we enjoyed creating it.

Matthieu Gautier

La philosophie du libre

Introduction

Je vais commencer par paraphraser Richard Stallman : « Imaginons que vous receviez des amis chez vous, vous leur présentez une nouvelle recette, peut être qu'ils l'apprécieront, et auront donc envie de la connaître, de la refaire, et peut être de la modifier, et de redistribuer cette nouvelle version ». Cet exemple culinaire est une métaphore parfaite de ce que peuvent être les logiciels Open Source.

Utilisateur ou non, nous avons eu l'occasion d'être confrontés aux logiciels Open Source, sans pour autant connaître, et comprendre leurs signification. Certaines personnes savent qu'il s'agit d'une vague nouvelle dans la branche de l'informatique, mais de là à connaître les lois qui la régissent.

Je vais donc essayer dans cet article de vous en faire une brève présentation, en détaillant au possible chacune des quatre lois fondamentales : Utilisation sans restriction, possibilité d'étude, de modification pour l'adapter à ses propres besoins et de redistribution, sous certaines conditions.

Utilisation sans restriction

Qu'est-ce que l'utilisation sans restriction ? Cela veut-il dire que l'on peut l'utiliser sans compter ? Pourtant de nombreux logiciels, ne serait-ce que les jeux vidéos sont utilisables sans compter, du moins par certains, mais

n'entrent pas dans la catégorie des logiciels Open Source.

L'utilisation sans restriction est différente de la définition que l'on donne habituellement au terme, ne serait-ce qu'en français.

La définition se base ici plus sur la manière que l'on a d'utiliser le produit à travers son système. Prenons un exemple, vous n'aurez en aucun cas le droit, et certainement la possibilité, d'utiliser un logiciel tel que Outlook sur un système autre que Windows, ce qui vous donne une restriction. Vous devez remplir certaines conditions pour pouvoir utiliser le logiciel, en plus de celle d'avoir un ordinateur.

Pour pouvoir soutenir la première loi de l'Open Source, le logiciel ne doit pas être le composant d'une distribution logicielle globale contenant plusieurs sources différentes.

Un autre point important, dans les restrictions, est que celle d'ordre financier ne doit pas être prise en compte, à savoir que l'exigence de quelque redevance ou honoraires implique directement que le logiciel n'est pas considéré comme Open Source.

La « gratuité » de ces logiciels est consciente, en renonçant dès le début aux bénéfices sur le long terme, on évite ainsi certaines pressions sur les épaules des coopérateurs.

Maintenant que nous avons pu voir la première loi, passons à la suivante.

Étudier

L'étude ne veut bien sûr pas dire la connaissance complète de toutes les possibilités du programme, même si par respect pour le développeur, cela est recommandé.

L'étude est ici utilisée dans le terme informatique de la chose, à savoir la possibilité de savoir comment est construit le programme et son fonctionnement, de manière à comprendre les tenants et les aboutissants, permettant alors une réponse à certaines questions.

Pour cela, le programme doit inclure ce que l'on appelle le code source, pour les lecteurs les plus novices, nous pouvons voir cela comme le plan de construction des meubles que vous achetez chez de fameux vendeurs de meubles suédois. Ce plan vous explique comment faire pour avoir une armoire aussi « réussie » que sur la photo du catalogue.

La loi exacte est donc que le



Richard Stallman :
Le fondateur du logiciel Libre.

programme doit inclure ce code source, mais aussi permettre sa distribution, et si jamais il n'est pas fourni avec, des moyens doivent permettre de mettre la main dessus, et ce avec un coût raisonnable, bien sûr le meilleur moyen restant le téléchargement par internet.

Bien entendu ce code source doit être clair, toutes ses parties pouvant être atteintes, et sans avoir nécessairement à passer par des intermédiaires, ou à se trouver face à un obscurcissement partiel, à savoir une manière plus ou moins compliquée de cacher le code source, ou une partie dans le cas d'un obscurcissement partiel.

L'accès au code source est donné parce que le développement n'est possible qu'avec la modification, or le but de l'Initiative Open Source est justement l'évolution, la modification facile, mais cela empiète sur la troisième loi que nous allons voir dès maintenant :

Modifier pour les adapter à ses propres besoins

Nous avons vu que l'accès au code source nous permet de comprendre le programme, mais aussi de pouvoir le modifier. Un exemple célèbre de modification, en tout cas de souhait de modification, est celui qui a poussé Richard Stallman à créer l'Initiative Open Source. Il avait l'habitude au MIT de modifier les programmes auxquels lui et ses collègues avaient accès, et un jour une des imprimantes ne marchait pas bien, Stallman appelle donc la compagnie qui vend l'imprimante et leur demande de lui fournir le code source pour qu'il puisse faire les modifications en conséquences. Face au refus de la société, Stallman se retrouva avec un appareil

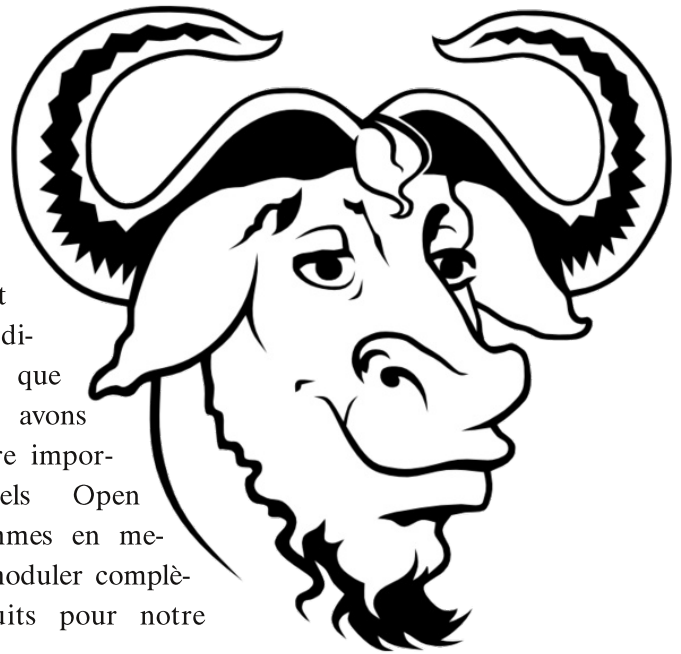
qui ne fonctionnait pas, et donc ne pouvait pas faire un travail productif.

La légende dit que l'Initiative Open Source est partie de ce fait divers, ce qui fait que maintenant nous avons accès à un nombre important de logiciels Open Source, nous sommes en mesure de pouvoir moduler complètement les produits pour notre propre utilisation.

Bien sûr cela demande un certain nombre de connaissances en matière de programmation, mais peut être intéressant d'un point de vue pédagogique, puisque le meilleur moyen pour apprendre reste l'expérimentation.

De plus cela implique un développement plus rapide des logiciels, car le monde de l'Open Source permet, grâce au développement d'internet, la mise en place de communautés qui se retrouvent autour de logiciels et autres programmes, le fait que ces programmes soient Open Source permet à chacun de les modifier pour qu'ils puissent convenir au plus grand nombre.

Nous pouvons prendre de nombreux exemples, mais les plus significatifs étant sûrement ceux que l'on peut voir au grand jour, à savoir le serveur Apache, la technologie PHP, etc.... Qui sont des outils que nous utilisons, en tout cas avons accès, tous les jours, sans même réellement le savoir, et dont le développement est poussé par la présence de communautés.



Redistribuer sous certaines conditions

Imaginons maintenant que vous ayez modifié en conséquence un logiciel et que vous aimeriez pouvoir le passer à vos amis ou aux personnes pour qui ces modifications seraient utiles.

Bien sûr vous devez respecter le travail de l'auteur initial, à savoir bien lire la licence, et voir si ce dernier le permet.

En imaginant le cas où l'auteur permet une redistribution, en n'oubliant pas de citer l'auteur original, vous pouvez vous permettre de faire partager vos modifications, quitte à recevoir des critiques, vous permettant alors d'avancer vers la perfection de votre programme.

Le terme utilisé est bien "votre" programme, car cela sous-entend que le programme est différent de celui utilisé précédemment. Par exemple, nous connaissons, en tout cas en majeure partie, Firefox, mais nous avons aussi un autre navigateur internet, IceWeasel, qui est écrit à partir des sources de Firefox, et

donc qui devient un programme indépendant.

Cette dernière loi peut paraître facile à comprendre, mais c'est aussi la loi la plus importante, car elle permet la distribution de différentes version d'un même logiciel, ce qui ne restreint plus l'utilisateur et empêche la formation d'un monopole.

Conclusion

Nous avons pu voir les quatre lois de l'Open Source, il ne faut pas oublier que le mouvement de l'Open Source est un mouvement assez jeune et surtout que c'est un mouvement qui va contre les lois et axiomes que l'on peut souvent trouver dans l'économie actuelle.

Pourtant l'Open Source n'empêche en aucun cas le développement économique, puisque de nombreuses entreprises basent leur stratégie sur la mise en place de logiciels Open Source, ne fournissant plus alors qu'un support pour l'utilisateur. Il est très important de se souvenir de cela, les gens ayant trop souvent tendance à ranger les logiciels Open Source dans la catégorie des logiciels gratuits, ayant un développement peu

Références

[1]

http://doc.fedora-fr.org/wiki/Liste_de_logiciels

Article sur Open Source Initiative de Wikipédia :

http://fr.wikipedia.org/wiki/Open_Source_Initiative

Définition de l'Open Source :

<http://www.opensource.org/docs/definition.php>

Forum francophone :

<http://www.fedora-fr.org>

professionnel, et donc un ensemble de programmes auxquels on peut difficilement faire confiance. Cela est bien sûr faux, Apache étant le meilleur exemple.

La prochaine étape du développement des logiciels Open Source semble maintenant être leur prise en main par le grand public, en espérant que ce dernier saura aller au delà des considérations de bas étage qui font des logiciels Open Source une sorte d'épouvantail de l'informatique pour se concentrer sur les possibilités qu'offrent ces derniers.

Je vous invite donc dès maintenant à vous renseigner un peu plus sur les logiciels auxquels vous avez accès[1], et à ne pas uniquement vous tourner vers la facili-

té, mais bien vers ce qui peut, sûrement, vous convenir le mieux.

Rédigé par : Matthieu Rondeau.

Relu et corrigé par : Fabien Archambault.



Présentation de Fedora

Naissance du projet

L'histoire de cette distribution Linux est généralement peu connue. Tout commence pourtant en 1995, lorsque la société RedHat est fondée. Son objectif est de fournir un système stable basé sur Linux ainsi qu'un service de support. RedHat sortira donc durant plusieurs années un système d'exploitation appelé RedHat Linux. Neuf versions majeures virent le jour jusqu'en 2003, année où RedHat décide de réorganiser sa gamme de produits. La fin de RedHat Linux est annoncée et cette dernière est alors remplacée par RedHat Entreprise Linux, distribution Linux pour serveur ou bureau à destination des entreprises, accompagné d'un support. Le Projet Fedora existait depuis quelques mois et sa vocation était de fournir des paquetages pour RedHat Linux. A la nouvelle de l'abandon de RedHat Linux, le Projet Fedora se vit demander de prendre en charge la continuité de RedHat Linux. Projet communautaire et ouvert aux contributeurs extérieurs, son objectif est de fournir une distribution Linux intégrant des logiciels très tôt dans leur cycle de développement, afin d'accélérer leur intégration à plus grande échelle. A cette époque et sans aucun recul, l'annonce de RedHat fut mal comprise par la communauté linuxienne. En effet, cela signifiait pour beaucoup l'abandon par RedHat de sa distribution phare « grand public ».

Il n'en fut rien. Derrière cet

abandon un accord avait été signé avec la communauté Fedora : l'objectif devenait de maintenir une distribution Linux basée sur la dernière version de RedHat Linux et de rendre le tout le plus communautaire possible. Pourtant, RedHat ne signait pas là la fin de son engagement dans la version grand public de sa distribution. Elle signait plutôt une sorte de partenariat avec Fedora pendant que l'entreprise concentrerait ses efforts sur le monde de l'entreprise et du serveur. Voilà, la messe était dite, Fedora serait communautaire. Avec le recul, nous pouvons dire que ce fut un pas de géant pour RedHat, qui avait pris l'habitude de minimiser les capacités de développement de sa communauté.

RedHat publiait bel et bien les sources de ses logiciels, mais peu de personnes en dehors des employés de l'entreprise étaient mises à contribution. Presque 5 ans après, la situation a encore évolué. RedHat, depuis presque un an, n'est plus le contributeur majoritaire dans Fedora. La communauté a pris le dessus. De plus, RedHat a synchronisé le cycle de développement de son système grand public avec celui de Fedora. RedHat s'appuie régulièrement sur une version de Fedora (une fois tous les dix huit mois) pour en faire son système RedHat Linux Entreprise. Plus que d'un signe de reconnaissance de la qualité de la distribution, c'est également le signe que la communauté prévaut.

Le nom

Le nom Fedora vient du nom du dépôt de logiciels tiers pour RedHat Linux. Un Fedora est un chapeau, et de surcroît celui que porte la mascotte de RedHat, ShadowMan. Depuis son lancement, le nom de la distribution a évolué et ce n'est qu'à partir de Fedora 7 que le nom perdit le mot « Core ». Avant Fedora 7, donc, Fedora s'appelait Fedora Core, signifiant par là qu'elle comprenait tous les logiciels inclus dans un dépôt logiciel du même nom, « Core ». Ce dépôt était exclusivement géré par RedHat. Le dépôt communautaire quant à lui s'appelait « Extras ». Grâce aux nombreuses demandes de la communauté, RedHat a accepté d'ouvrir le dépôt Core à la communauté. Finalement, c'est à une fusion des deux dépôts principaux que nous avons assisté, regroupant à la fois employés de RedHat et contributeur communautaire.



Le logo

Peu avant la sortie de Fedora Core 5, le Projet Fedora entreprit de renforcer son identité visuelle. Elle se lança donc dans la création d'un logo qui devrait refléter les principes de la distribution : la liberté sans limite des logiciels et le respect de la parole



de sa communauté. Dès lors, le logo que nous connaissons aujourd'hui a été élaboré et se compose ainsi.

Organisation interne

Un comité de pilotage fut mis en place pour gérer l'organisation du projet. Celui-ci est composé de 9 membres, dont 4 sont nommés par RedHat, 5 sont élus par la communauté des contributeurs du Projet Fedora. Un de ces membres est le Dirigeant du Projet Fedora et est à la tête du Comité de pilotage. Il représente le projet de part le monde et s'assure de son bon fonctionnement et du respect de ses règles. Toujours très respecté par la communauté, on peut facilement dire que ce dirigeant est une force d'argumentation non négligeable au sein du Projet Fedora. Le Projet Fedora est ensuite organisé en sous-projets, dont chacun s'occupe d'un aspect précis du Projet du développement de la distribution. Par exemple, le Projet Infrastructure s'occupe de gérer les sites web et les serveurs du Projet. Le Projet Documentation s'occupe de la documentation officielle de la distribution. D'autres encore, s'occupe du marketing, de la traduction, de l'identité visuelle de Fedora.

Les domaines sont si nombreux qu'il est très souvent facile de se laisser déborder par les possi-

bilités de s'impliquer dans la communauté de Fedora.

L'aspect communautaire

Le projet Fedora est avant tout communautaire, c'est à dire que le projet n'est pas seulement guidé selon la volonté d'une entreprise. RedHat est présent, bien sûr, pour lancer des impulsions fortes et financer le projet. Le reste n'est que bénévolat... mais quel bénévolat ! La communauté Fedora, initialement impulsée par RedHat se rassemble de part le monde autour du site web officiel de la distribution. C'est par lui, et malgré son austérité que les contributeurs de Fedora ont sû se sentir chez eux. De nombreux événements sont également organisés par les bénévoles eux-mêmes, par simple soif de parler du projet auquel ils appartiennent. RedHat a également mis à la disposition de la communauté de nombreuses listes de diffusion. C'est actuellement ces listes de diffusion que se discute et qu'est décidé l'avenir de Fedora aujourd'hui. La communauté s'est ensuite organisé d'elle même ou non en séparant les activités qui existent au sein d'un projet aussi vaste : le développement, le marketing, la traduction etc. Fedora est une méritocratie. C'est à dire que vous si vous prenez de plus en plus de responsabilités au sein du projet, c'est parce que vous méritez que l'on vous fasse confiance. Bien entendu, il faut avoir montré patte blanche auparavant.

Intransigente mais humaine

Oui, la communauté Fedora est intransigente. Non pas envers ses membres, mais envers les va-

leurs qu'elle défend et promet : la liberté du logiciel. Le Projet Fedora s'applique à travailler uniquement sur des projet et des logiciels libres. Vous ne trouverez dans Fedora aucun logiciel propriétaire ou allant à l'encontre des libertés fondamentales du logiciel libre. Fedora s'applique à respecter ses utilisateurs, en leur proposant un système robuste et fiable, basé sur des logiciels libres et gratuits. Régulièrement, des membres de la communauté ont tenté, en vain, de faire évoluer la position du Projet sur ce sujet. Derrière ce qui pourrait sembler un choix d'informaticien puriste se cache un idéal que nous ne sommes pas si loin d'atteindre : réussir à se passer de logiciel propriétaire dans tous les domaines de l'informatique. Et le seul moyen ayant été trouvé aujourd'hui pour achever cet objectif est de ne pas encourager le développement ni la distribution de logiciels propriétaires. Fedora n'a qu'un objectif, améliorer le logiciel libre. Tout ce qui va à l'encontre de cet objectif n'est pas le bien venu dans le Projet Fedora.

Rédigé par : Thomas Canniot.

Utilisation de la ligne de commande

Introduction

D'un point de vue historique, la ligne de commande est l'interface la plus ancienne développées sur ordinateur. Cette interface, permettant un dialogue entre l'ordinateur et l'utilisateur, a vu le jour à partir des années 60 avec l'apparition des téléscripteurs, qui sont des appareils permettant la génération et la réception de messages via des signaux électriques, le plus souvent via ligne filaire, et qui s'est vraiment développé à partir des années 70 avec l'invention des consoles.

D'un point de vue fonctionnel, la console interprète la commande que tape l'utilisateur, le procédé fonctionnant ainsi : la console analyse la commande rentrée pour ensuite l'exécuter si cette dernière est syntaxiquement correcte, puis passe à la commande suivante (si elle existe) selon l'opérateur de contrôle.

Convention d'écriture

Dans cet article (et le suivant) les conventions d'écriture suivantes seront utilisées:

ce que vous voulez correspond à du texte qui doit ou peut être adapté par le lecteur en fonction de son system et de ses besoins.

commande correspond a des commande que le lecteur doit entrer telquel.

A notre époque, où les interfaces graphiques font la loi, la majorité des utilisateurs ne consacre que peu de temps à l'apprentissage, et encore moins à l'utilisation de la ligne de commande.

Je vais tenter dans cet article de vous montrer la puissance de son utilisation, sans bien sûr aller trop loin, ce qui vous permettra ensuite de vous renseigner et d'apprendre par vous même ce que vous offre cette manière d'utiliser son ordinateur.

Mise en garde

Bien sûr, on ne peut commencer un article sur la ligne de commande sans mettre en garde les nouveaux utilisateurs. Tout d'abord n'oublions pas que l'interface que nous donne la ligne de commande peut être vraiment rebutante au départ. Nous avons pris l'habitude de "voir" nos fichiers, ici ce ne sera que des titres. De même, ici, bannie la souris, tout se fait au clavier (exception faite pour les copier/coller).

De plus cela vous demande un certain esprit d'abstraction, pouvoir voir au-delà des quelques lignes qui s'affichent, et un effort de mémoire, à savoir connaître les formules principales. Mais heureusement vous verrez que ces formules s'apprennent facilement au fur et à mesure de leur utilisation.



Un exemple de Téléscripteur

Par contre, les avantages sont nombreux dans l'utilisation de la ligne de commande, que j'espère mettre en avant au fil de cet article. Nous pourrons toute fois citer en tout premier lieu la légèreté de ce procédé, sa rapidité, et surtout le fait que c'est la seule chose qui fonctionne lorsque vous n'avez plus de serveur graphique.

Première approche

Maintenant que vous êtes mis en garde, passons dans le vif du sujet.

Le gros avantage de la ligne de commande est le faible besoin en ressource de son utilisation, ainsi vous pouvez facilement faire tourner plusieurs programmes, ou faire plusieurs tâches, sans voir le moindre ralentissement.

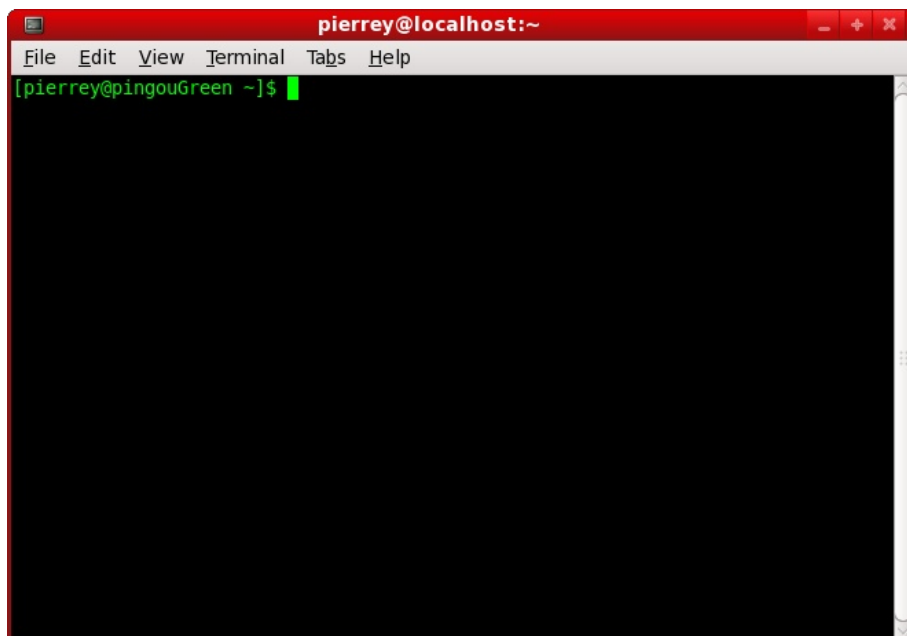
Vous connaissez sûrement les commandes principales :

\$ cd *le dossier où aller*

Vous permet de vous déplacer dans l'arborescence.

\$ pwd

Vous écrit le nom du répertoire où vous êtes.



Interface de ligne de commande

\$ ls
Liste les éléments du répertoire courant.

Consultez le lien [1] pour une liste complète.

L'interface de la ligne de commande marche de manière très simple, vous aurez à taper quelque chose de la forme :

\$ commande <option> arguments

Les options sont trouvable la plupart du temps en tapant :

\$ commande -h

ou simplement en faisant un **\$ commande**

sans mettre ni d'arguments ou d'option. Si vous voulez une description complète de la commande, vous aurez à taper :

\$ man commande

Le principe de la ligne de commande est le suivant, lorsque vous tapez une commande, l'ordinateur va chercher dans la variable PATH le chemin des dossiers contenant les commandes. Cette variable d'environnement est affichable en faisant :

\$ echo \$PATH

Le fait d'avoir plusieurs dossiers contenant les commandes

permet d'utiliser l'auto-complétion, à savoir lorsque vous commencez à taper une commande, vous pouvez utiliser la touche [Tab] qui vous terminera la commande, si il existe plusieurs possibilités, en appuyant deux fois sur cette touche, vous aurez les différentes commandes possibles qui s'afficheront.

Cela fonctionne aussi avec les noms de fichiers ou dossier en argument, mais pas pour les options.

Certaines commandes ne sont utilisables que par un utilisateur particulier, l'utilisateur "root". Pour pouvoir s'identifier en tant que cet utilisateur, vous aurez à taper :

\$ su -

votre mot de passe root

Le tiret est important, car il permet de charger le PATH du root.

Passons maintenant à quelques exemples.

Exemple n°1

Bon maintenant voyons un des premier avantage de la ligne de commande, imaginons que vous ayez des fichiers à classer,

avec différents type. En mode graphique, vous devriez classer selon le type, et sélectionner les différents fichiers, faire "couper" puis coller dans le répertoire voulu.

Voila comment le faire en ligne de commande :

Imaginons que le dossier /media/canon contienne des fichiers en .jpg à placer dans /home/Matt/Image/Photo et des .mpg à placer dans /home/Matt/Video, nous pourrions faire comme en mode graphique, à savoir prendre chaque photo et chaque vidéo, et les déplacer dans le dossier voulu. Bien que [Tab] peut être utile dans ce cas, le faire fichier par fichier peut être rapide, mais beaucoup plus lente qu'en mode graphique, pour peu que vous ayez classé vos fichiers selon type.

Nous allons donc utiliser une des fonctionnalité de bash, à savoir le caractère "*" (joker), ce caractère permet de représenter n'importe quel caractère, à savoir que si l'on tape "*.jpg", on parle ici de tous les fichiers terminant par .jpg, donc tous nos fichiers photos. Il suffit alors de taper :

```
$ cd /media/canon
```

```
$ mv *.jpg
```

```
/home/Matt/Image/Photo
```

```
$ mv *.mpg /home/Matt/Video
```

mv est la commande pour déplacer ou renommer les fichiers/dossiers

On peut ajouter les commandes les unes après les autres en utilisant l'opérateur de contrôle ";", qui permet de terminer la commande et de passer à la suivante, ce qui fait que l'on peut taper directement :

```
$ mv *.jpg  
/home/Matt/Image/Photo; mv *.mpg  
/home/Matt/Video
```

Exemple n°2

Imaginons que vous lanciez un programme tel que `amsn` par votre menu, et le programme plante, il vous faudra l'arrêter, mais comment faire cela simplement?

La première chose à connaître est ce que l'on appelle le pipe (un opérateur de contrôle), sans rentrer dans les détails des entrées/sorties en shell, il faut savoir que les commandes ont une sortie, qui va donc s'afficher (l'erreur est aussi une sortie, mais qui est différente), il serait alors utile de pouvoir rediriger cette sortie, c'est là que le pipe, que l'on écrit avec le caractère `"|"` en shell intervient. Le pipe prend la sortie de la commande de gauche comme entrée de la commande se trouvant à droite.

Voyons cela dans l'exemple suivant :

Si je tape

```
$ ps ax
```

`ps` liste les processus actifs, l'option `ax` donne tous les processus

et que je cherche `amsn`, je risque de perdre beaucoup de temps, mais si maintenant, je fais :

```
$ ps ax | grep amsn
```

`grep` vous donne les lignes contenant le mot voulu, ici `amsn`

Je ne perds pas de temps et je peux directement faire :

```
$ kill -9 pid_amsn
```

le `pid`, ou processus ID, est le numéro donné au processus

Exemple n°3

Un ami vous a donné un lien vers un ftp où il a mis les photos de vos dernières vacances,

le problème c'est qu'il y a d'autres fichiers tels que des vidéos, et que vous aimeriez bien télécharger directement toutes les photos en `.jpg`.

Voilà la procédure :

```
$ cd Où vous voulez mettre ces photos
```

```
$ wget -r -ll --no-parent -A.jpg  
trucmuche.free.fr
```

`wget` vous permet de télécharger les contenus d'un site

Je vous laisse un peu chercher pour savoir ce que veulent dire toutes ces options.

Passons maintenant à une utilisation plus avancée de la ligne de commande.

Maîtrisons la ligne de commande

Maintenant que nous avons vu quelques commandes utiles, nous allons essayer de voir quelques petites choses plus intéressantes. Comment faire pour personnaliser notre interface de ligne de commande, pouvoir utiliser au mieux cet outil pour automatiser des tâches souvent peu intéressantes.

Nous allons voir tout cela, bien sûr il va de soi que je ne peux en un article vous faire l'étalage de toutes ces choses, et le plus gros travail sera de chercher et découvrir de nouvelles manières d'utiliser la ligne de commande au mieux.

Les logiciels en ligne de commandes

Le sujet étant très vaste, je vais essayer de rester généraliste, en vous présentant les exemples les plus significatifs.

La première chose à maîtriser est l'éditeur de texte en

ligne de commande, d'abord parce qu'il est toujours utilisé de pouvoir ouvrir un fichier éditable en ligne de commande, et ensuite parce que vous pourrez avoir l'occasion de modifier certains fichiers uniquement en ligne de commande, comme par exemple le très célèbre `xorg.conf`.

Le plus célèbre est sans aucun doute `Vim`, qui est directement inspiré de `vi`, éditeur très répandu sur les systèmes UNIX, donc il est le clone le plus populaire. Son nom signifie `Vi Improved`, que l'on peut traduire par « `Vi AMélioré` », ou éventuellement « `VI Meilleur` ».[4]

Contrairement à `vi`, `Vim` est un logiciel libre, son code source a été publié pour la première fois en 1991 par `Bram Moolenaar`, son principal développeur. Depuis ce dernier a continué à l'améliorer, avec l'aide de nombreux contributeurs.

`Vim` possède son propre langage d'extension, ce qui lui permet la prise en charge de tâches évoluées, telles que la production de code source. Malgré de nombreuses fonctionnalités, il conserve un temps de démarrage court et reste ainsi adapté pour des modifications simples et ponctuelles (fichiers de configuration par exemple).

Le lancement de `Vim` est très simple, vous pouvez taper :

```
$ vim
```

lance directement `vim`

```
$ vim nom du fichier à éditer
```

lance `vim` en éditant le fichier voulu

Vous pouvez consulter un petit résumé des commandes en [3].

Pouvoir ajouter des logiciels

dans Linux est une chose très intéressante, car le nombre de logiciels auxquels vous avez accès ne cesse d'augmenter. Vous pouvez toujours utiliser des outils graphiques pour ceci, mais sachant que l'installation consomme déjà une partie importante des ressources, le faire en ligne de commande permet d'alléger le processus.

Fedora utilise Yum comme gestionnaire de paquets, YUM (Yellow dog Updater Modified) est un gestionnaire de paquet, créé par yellow dog et utilisé par Fedora pour la gestion des paquets.

Il permet de gérer l'installation et la mise à jour des logiciels sur une distribution GNU/Linux. C'est une surcouche à RPM gérant les téléchargements et les dépendances, de la même manière que APT de Debian ou Urpm* de Mandriva.

L'utilisation de Yum est très simple, vous aurez une syntaxe du type :

```
# yum <option> commande  
argument
```

Le plus souvent en root.

Nous ne parlerons pas plus de yum puisque l'article suivant lui

Fig 1 : Le contenu de `.bashrc` d'origine

```
# .bashrc  
# Source global definitions  
if [-f /etc/bashrc]; then  
    /etc/bashrc  
fi  
# User specific aliases and functions
```

Fig 2 : Le contenu de `.bashrc` modifié

```
# .bashrc  
# Source global definitions  
if [-f /etc/bashrc]; then  
    /etc/bashrc  
fi  
# User specific aliases and functions  
alias vlcfs = 'vlc --fullscreen'  
alias rpmi = 'rpm -ivh'  
alias rpmU = 'rpm -Uvh'
```

est consacré.

Les alias

Voyons, vous avez des commandes que vous aimeriez bien pouvoir lancer de manière la plus rapide possible, prenons par exemple :

```
$ vlc --fullscreen mavdeio.avi
```

lance vlc en plein écran

```
$ rpm -ivh monpaquet.rpm
```

installe un paquet rpm en mode verbose

```
$ rpm -Uvh monpaquet
```

met à jour un logiciel à partir d'un paquet rpm en mode verbose

Et ainsi de suite, vous pouvez lancer vos commandes comme ceci, mais par gain de rapidité, pourquoi ne pas les implanter vous même, et c'est

```
#!/bin/sh  
#ordonne l'interprétation par le bourne shell  
#  
# le '#' en début de ligne indique un commentaire  
#  
# Shell-script de mise en application de la récupération du résultat  
# d'une commande. Affiche l'heure en français.  
heures=$(date +%H) #nous donne l'heure prise à partir de la commande date  
minutes=`date +%M` #nous donne les minutes, en interprétant la commande, rôle des  
''  
echo "Il est $heures heures $minutes minutes et $(date +%S) secondes"
```

```
pierrey@localhost:~  
File Edit View Terminal Tabs Help  
Loaded plugins: presto, refresh-packagekit  
Setting up and reading Presto delta metadata  
fedora | 2.4 kB 00:00  
primary.sqlite.bz2 | 7.6 MB 00:18  
No Presto metadata available for fedora  
updates | 2.3 kB 00:00  
primary.sqlite.bz2 | 2.7 MB 00:00  
No Presto metadata available for updates  
livna | 2.1 kB 00:00  
primary.sqlite.bz2 | 187 kB 00:00  
No Presto metadata available for livna  
Setting up Update Process  
No Packages marked for Update  
[root@pingouGreen ~]# ls  
anaconda-ks.cfg install.log install.log.syslog robots.txt scsrn.log  
[root@pingouGreen ~]# ll  
total 96  
-rw----- 1 root root 1486 2008-03-31 21:28 anaconda-ks.cfg  
-rw-r--r-- 1 root root 55002 2008-03-31 21:27 install.log  
-rw-r--r-- 1 root root 5278 2008-03-31 21:22 install.log.syslog  
-rw-r--r-- 1 root root 115 2008-04-22 11:13 robots.txt  
-rw-rw-r-- 1 root root 16675 2008-05-17 20:58 scsrn.log  
[root@pingouGreen ~]# less install.log  
[root@pingouGreen ~]#
```

La ligne de commande devient plus compliquée

pour cela que les alias sont là.

Prenons par exemple le "vlc --fullscreen", que vous voudriez pouvoir lancer en faisant "vlfcs". Tout d'abord ouvrez votre fichier ~/.bashrc, qui devrait ressembler à contenance la figure 1.

Vous comprendrez bien sûr qu'il est recommandé de noter les alias en dessous de la ligne "# User specific aliases and functions", simplement pour une meilleure lisibilité, avec la syntaxe "alias <votre commande> = <commande existante>" votre .bashrc deviendra donc comme ce lui de la figure 2.

Il va de soi qu'il ne faut surtout pas utiliser de commandes déjà utilisées par le système.

Script

Les scripts permettent une automatisation du système, ils peuvent être écrits en différents langages. L'exemple suivant est un script en bash (Bourne Shell) pour nous donner l'heure en français.

Pour cela, créons donc le fichier "heure", ce que

nous pouvons faire en utilisant Vim :

Voilà ce qui devrait s'afficher à l'écran, après lancement :

```
$./heure
```

Il est 14 heures 08 minutes et 23 secondes

Voilà, cet exemple est très simple, mais essayons de voir plus loin, vous voulez automatiser une suppression de certains fichiers, un encodage vidéo, ou autre, tout cela est possible, de manière rapide.

Conclusion

Nous avons pu voir que la ligne de commande permet,

lorsqu'elle est maîtrisée, de vous rendre la vie beaucoup plus facile, mais il va de soi qu'il vous faut prendre le temps de se faire à ce genre d'outil, mais prendre le temps de pouvoir l'utiliser peut grandement vous aider.

Cet article est généraliste, et n'est pas là pour vous donner toutes les réponses à vos questions, mais bien pour vous permettre d'avoir un aperçu de cet outil.

Il est bon de noter que la ligne de commande est indispensable dans beaucoup de cas, par exemple tout simplement lorsque votre serveur graphique ne fonctionne pas, et que vous devez alors éditer votre xorg.conf à la main, mais aussi pour des tâches plus simples, comme l'exécution de scripts d'installation, la compilation de logiciel, puisque l'utilisation du script configure, et des commandes make, make install, ne se fait qu'à la ligne de commande.

Il tient maintenant à vous de prendre le temps de bien assimiler la ligne de commande, et de vous renseigner le plus possible pour pouvoir l'utiliser au mieux.

Rédigé par : Matthieu Rondeau.

Références

Commandes Linux :

<http://www.linux-france.org/article/debutant/debutant-linux.html> site

Cours de bash :

<http://eric-sanchis.developpez.com/linux/shell/bash/> site

Tutorial sur Vim :

http://randylinux.unblog.fr/2008/03/30/petites-choses-sur-vi-partie-i/Partie_1

http://randylinux.unblog.fr/2008/03/31/petites-choses-sur-vi-partie-ii/Partie_2

Article sur Vim :

<http://fr.wikipedia.org/wiki/Vim> Wikipédia

Utilisation de yum : Les premiers pas

Une distribution, c'est quoi ?

Votre système d'exploitation favori, vous le savez sans doutes, est diffusé par l'intermédiaire de ce que l'on nomme les Distributions Linux. Fedora en est une, mais nous devons entre autres retenir RedHat, Debian, Slackware, Mandriva... et de nombreuses autres encore. Une distribution est - globalement - le regroupement d'un ensemble de logiciels, gravitant autour du noyau linux, installable sur une machine, pour lui permettre de fonctionner. Cette définition est sans aucun doutes imparfaite, mais elle a le mérite d'être relativement simple.

Les logiciels utilisés sont des logiciels dont les sources sont disponible librement, et qu'en théorie, il faut compiler soit même. Grossièrement, les pièces détachées de la voiture sont données, les plans également, mais c'est à vous de réaliser le montage. Une distribution se propose de vous simplifier le travail, en rendant accessible ces logiciels déjà compilés, et utilisables directement. Comme plusieurs logiciels inter-agissent entre eux, et que le bon fonctionnement de l'un dépend fortement du bon fonctionnement d'un autre, une uniformité est utilisée dans les distributions, notamment dans l'emplacement des fichiers de configuration, des exécutable ...

On va mettre le paquet !

Les logiciels ainsi préparés sont distribués par les projets de distribution (dans notre cas, le projet Fedora) sous forme de paquet (ou paquetage, package en bon français de france...). Ils ont dans le monde Fedora un format et une extension nommée ".rpm". Ceci est historique : Fedora est issu du projet RedHat, distribution linux professionnelle : RPM signifie "Red Hat Package Manager".

Qu'est-ce qui différencie un RPM d'un bête - et c'est peu de le dire - d'un bête setup.exe ? Et bien, la notions de dépendance.

Pour les drogués...

La dépendance, c'est de ne pas pouvoir se passer de quelque chose. Les fichiers RPM incluent en leur sein des déclarations permettant de déterminer quels sont les programmes dont ils ont besoin pour fonctionner. Ils sont aussi capable de dire avec lesquels ils ne peuvent pas coexister ... Ainsi, si vous essayez d'installer deux driver différents pour votre carte vidéo, vous serez averti de cette incompatibilité de caractère et, à moins que vous ne soyez vraiment convaincant, vous serez forcé d'en supprimer un des deux.

Le comptable de la boîte.

Un paquet s'installe et se supprime d'un système avec la commande "rpm". Cependant, cet outils, s'il est capable de vérifier qu'un paquet peut s'installer



(dépendances, incompatibilité), il reste encore rudimentaire et n'automatise pas tout.

Le programme Yum vous permet de réaliser de nombreuses tâches relatives à la gestion des logiciels présent sur votre distribution linux. Les voici résumés ici :

- * Rechercher un logiciel ;
- * Installer un ou plusieurs logiciels ;
- * Gérer les dépendances, et les exclusions mutuelles ;
- * Supprimer un ou plusieurs logiciels ;
- * Mettre à jour vos logiciels.

Ce dernier point est remarquable, et voici une comparaison dans le but de l'éclaircir : lorsque vous mettez à jour votre système utilisant les logiciels Microsoft par exemple, vous mettez à jour le système lui même (windows), et, parfois, d'autres logiciels du même éditeur (la suite office par exemple). Pour le reste, débrouillez vous. Il vous appartient de vous tenir informé des nouvelles versions de l'ensemble de vos programmes, et de les mettre à jour manuellement, en allant les récupérer sur les sites des éditeurs respectifs.

Ce n'est pas le cas dans un système basé sur linux, et en

particulier avec Fedora. Lorsque vous mettez à jour votre système, l'ENSEMBLE des programmes présents sont mis à jour. D'un certain point de vue, la distribution (au sens "projet") est l'éditeur de l'ensemble des logiciels que vous utilisez.

Utilisation de Yum : quelques exemples

Yum est donc un logiciel qui vous permet d'installer un logiciel simplement. Il s'utilise en ligne de commande (c'est à dire, dans un terminal), et en étant super-utilisateur.

Nous allons tout d'abord installer un logiciel quelconque. Cet exemple vous permet d'installer `linux_logo`, un logiciel qui n'est probablement pas encore installé sur votre machine. C'est un tord, car il est presque absolument inutile. Pourquoi s'en priver ?

Dans un terminal, tapez donc :

```
$ su -
```

Mot de passe root

Vous êtes maintenant le super-utilisateur de votre machine. Faites attention à ce que vous faites !

Pour installer `linux_logo`, nous pouvons taper :

```
# yum install linux_logo
```

Comme vous pourrez le lire dans ce qui s'affichera, yum est parti chercher sur internet le ou les logiciels pouvant correspondre à `linux_logo`, en a trouvé un, vous récapitule cela dans un tableau, et vous demande si c'est bien ce que vous vouliez. Si vous répondez oui (avec "o"), alors yum installera le logiciel sur votre machine.

Ce n'était pas bien

compliqué...

Pour enlever un logiciel, vous pouvez utiliser la commande suivante :

```
# yum remove linux_logo
```

Et voir ce qu'il se passe. Cependant, je vous conseille de jouer avec `linux_logo` avant de l'enlever, ce serait dommage ...

Pour se rendre compte du mécanisme de dépendances, vous pouvez tester la commande suivante :

```
# yum install eclipse
```

Vous pourrez voir l'ensemble des logiciels dont `eclipse` a besoin, et qui seront donc installés automatiquement. C'est vachement pratique, tout de même, non ?

Si vous ne souhaitez pas installer `eclipse`, il vous suffira de répondre "n" (non) lors du résumé de la transaction.

Pour rechercher un logiciel, si vous connaissez pas son nom exact, vous pouvez utiliser les fonctions de recherche de yum. Deux options s'offrent à vous :

La version "simple", avec la commande `list`.

```
# yum list linux|*
```

vous permet de lister tout les paquets dont le nom commence par "linux". Vous devriez y retrouver le fameux paquet `linux_logo`. Si vous voulez chercher dans la description, le résumé, le nom de l'empaqueteur ..., vous pouvez utiliser la commande "search" :

```
# yum search linux
```

La liste de résultats est, je vous l'assure, conséquente.

Pour mettre à jour un logiciel, vous pouvez utiliser la fonction "update" de yum :

```
# yum update firefox
```

Pour mettre à jour l'ensemble de votre système :

```
# yum update
```

Dans ce cas précis, yum ira vérifier que les versions installées sur votre système sont bien les plus récentes disponible. Et si vous y ajouter l'option `-y` (pour "yes"), alors yum ne vous demandera même pas votre avis et installera les dernière versions disponibles.

```
# yum -y update
```

L'effet est garantie lorsque vous exécutez cette commande sur un système récemment installé



et disposant de nombreuses mises à jour. Par exemple, installez Fedora 10 sur une machine, à partir du DVD, puis lancez cette commande devant un ami : vous pourrez lui expliquer comment votre belle machine va trouver, toute seule, les dernières versions des logiciels.

Fonctionnement de Yum : quelques explications

Vous avez pu le remarquer, l'usage de yum est relativement simple. Cette simplicité apparente cache en fait un ensemble de fonctionnalités destinées à vous simplifier la vie.

Yum est capable de résoudre les dépendances. À partir des informations contenues dans le paquetage, Yum construit un arbre de dépendances et vérifie que celles-ci sont satisfaites. Si des éléments nécessaires au bon fonctionnement d'un logiciel sont manquants, alors ils sont proposés automatiquement pour l'installation.

Yum va chercher les informations sur les paquetages disponibles, et les paquetages eux-mêmes, sur les différents dépôts de Fedora. Un dépôt est un réseau de serveurs mis à disposition de la communauté, et synchronisés entre eux.

Les paquets présents sur les dépôts sont signés numériquement. C'est une mesure de sécurité, permettant de contrôler la provenance des paquets téléchargés via Yum. Ainsi, lors de la première utilisation de Yum, ils vous seront demandés de confirmer l'utilisation des clés nécessaires à la vérification des paquets. Cela

permet de s'assurer qu'un des serveurs faisant partie du dépôt ne cherche pas à diffuser des paquets corrompus (contenant des virus par exemple). Il est important de noter que cette vérification d'intégrité numérique est propre à la distribution Fedora. Certaines distributions populaires ne s'en préoccupent pas.

Conclusion

Yum est un outil puissant, et faisant partie intégrante de Fedora. Certes, il est en mode console, et pas en mode graphique, et peu donc en faire fuir quelques uns. Cependant, des interfaces graphiques basées sur Yum existent (comme PackageKit). S'il est parfois critiqué, notamment pour sa prétendue lenteur, Yum reste un outil extrêmement puissant, capable de gérer des problèmes complexes. De plus, de nombreux outils complémentaires, présents sous la forme de plugins, permettent d'étendre les fonctionnalités de yum de façon significative.

Nous aurons l'occasion de parler des différents plugins de yum, et de leur usage, dans un prochain article. Nous aborderons également l'utilisation d'autres dépôts que ceux utilisés par défaut sur votre système, et nous verrons à quoi ceux-ci peuvent servir.

Rédigé par : Valentin Guerlesquin.