



Full Circle

LE MAGAZINE INDÉPENDANT DE LA COMMUNAUTÉ UBUNTU LINUX

ÉDITION SPÉCIALE



Command & Conquer



Volume Deux

Numéros 26 à 50



Spécial Full Circle Magazine

Full Circle

LE MAGAZINE INDÉPENDANT DE LA COMMUNAUTÉ UBUNTU LINUX

Au sujet du Full Circle

Le Full Circle est un magazine gratuit, libre et indépendant, consacré à toutes les versions d'Ubuntu, qui fait partie des systèmes d'exploitation Linux. Chaque mois, nous publions des tutoriels, que nous espérons utiles, et des articles proposés par des lecteurs. Le Podcast, un complément du Full Circle, parle du magazine même, mais aussi de tout ce qui peut vous intéresser dans ce domaine.

BIENVENUE DANS UN AUTRE « NUMÉRO SPÉCIAL »

Une autre série, une autre compilation d'articles pour plus de commodité. Voici une réédition directe de la série Command & Conquer, des numéros 26 à 50.

Veillez garder à l'esprit la date de publication originale ; les versions actuelles du matériel et des logiciels peuvent être différentes de celles illustrées. Il convient donc de vérifier la version de votre matériel et de vos logiciels avant d'essayer d'émuler les tutoriels dans ces numéros spéciaux. Il se peut que les logiciels que vous avez installés soient plus récents ou qu'il y ait des versions plus récentes disponibles dans les dépôts de votre distribution.

Amusez-vous !

Clause de non-responsabilité :

Cette édition spéciale vous est fournie sans aucune garantie ; les auteurs et le magazine Full Circle déclinent toute responsabilité pour des pertes ou dommages éventuels si des lecteurs choisissent d'en appliquer le contenu à leurs ordinateur et matériel ou à ceux des autres.

Sommaire

Numéro 26, part. 12 : page 3
Numéro 27, part. 13 : page 5
Numéro 28, part. 14 : page 7
Numéro 29, part. 15 : page 9
Numéro 30, part. 16 : page 11

Numéro 31, part. 17 : page 13
Numéro 32, part. 18 : page 16
Numéro 33, part. 19 : page 19
Numéro 34, part. 20 : page 21
Numéro 35, part. 21 : page 23
Numéro 36, part. 22 : page 28
Numéro 37, part. 23 : page 30

Numéro 38, part. 24 : page 32
Numéro 39, part. 25 : page 34
Numéro 40, part. 26 : page 36
Numéro 41, part. 27 : page 39
Numéro 42, part. 28 : page 42
Numéro 43, part. 29 : page 44
Numéro 44, part. 30 : page 46

Numéro 45, part. 31 : page 48
Numéro 46, part. 32 : page 50
Numéro 47, part. 33 : page 52
Numéro 48, part. 34 : page 56
Numéro 49, part. 35 : page 58
Numéro 50, part. 36 : page 60

Comment contribuer : page 63



Les articles contenus dans ce magazine sont publiés sous la licence Creative Commons Attribution-Share Alike 3.0 Unported license. Cela signifie que vous pouvez adapter, copier, distribuer et transmettre les articles mais uniquement sous les conditions suivantes : vous devez citer le nom de l'auteur d'une certaine manière (au moins un nom, une adresse e-mail ou une URL) et le nom du magazine (« Full Circle Magazine ») ainsi que l'URL www.fullcirclemagazine.org (sans pour autant suggérer qu'ils approuvent votre utilisation de l'œuvre). Si vous modifiez, transformez ou adaptez cette création, vous devez distribuer la création qui en résulte sous la même licence ou une similaire.

Full Circle Magazine est entièrement indépendant de Canonical, le sponsor des projets Ubuntu. Vous ne devez en aucun cas présumer que les avis et les opinions exprimés ici ont reçu l'approbation de Canonical.



Avant d'en venir au sujet qui nous intéresse, je voudrais prendre un instant pour remercier le lecteur Harold De Bruijn pour avoir souligné l'existence de `pacpl` (<http://pacpl.sourceforge.net/>) et sa capacité à convertir les « tags ». En plus, il est dans les dépôts d'Ubuntu. Il peut être utilisé à la place de `ffmpeg` pour la fonction `m4a` que j'ai présentée dans le dernier numéro, car il sauvegarde les balises dans le format dans lequel vous convertissez.

Mais revenons à nos moutons. Lorsque vous parlez de ligne de commande, la plupart des gens pensent à des commandes compliquées ou à des terminaux noirs avec du texte blanc (ou souvent vert) qui défile en grande quantité. Ou alors ils pensent à `apt-get`, `aptitude`, `elinks`, etc., car ce sont des applications très répandues et couramment utilisées. Cependant, combien de personnes pensent à MOC (Music On Console) ou `irssi` (client IRC) ?

Ces deux programmes utilisent une interface de type console et sont très utiles (et léger, si c'est ce que vous recherchez).

Vous en avez assez que votre musique s'arrête quand votre serveur X plante (ou lorsque vous l'arrêtez) ? MOC continuera de lire la musique, car il fonctionne comme un serveur en ligne de commande et vous pouvez l'exécuter à partir d'une console tty sans serveur X. Ou alors, votre serveur X a planté et vous avez besoin d'aide pour résoudre le problème, mais vous n'avez aucun moyen d'en demander ? `Irssi` vous permettra d'aller sur IRC (ainsi vous pourrez vous rendre sur `#ubuntu` ou sur `#ubuntu-fr` sur `freenode`, par exemple). Ces applications sont non seulement utiles lorsque vous êtes sans interface graphique, mais elles sont aussi personnalisables à souhait (couleurs, fonctions supplémentaires, etc.). Moi, par exemple, j'utilise un script pour récupérer les informations de MOC dans `conky`, mais vous pouvez également écrire un script qui sera exécuté à la fin d'une

chanson pour afficher le titre de la nouvelle chanson et le placer dans un fichier texte (afin d'obtenir une mise à jour véritablement en temps réel).

Tout d'abord, voyons un peu MOC. Pour installer `moc`, exécutez la commande suivante :

```
sudo apt-get install moc
```

Une fois le programme installé, vous pouvez le démarrer en tapant :

```
mocp
```

La vue par défaut est un diptyque avec, sur le volet gauche, un navigateur de fichiers, et sur le côté droit, la liste de lecture des musiques (vide pour l'instant). Utilisez la touche `Tab` pour passer du navigateur à la liste de lecture. Dans le menu du navigateur de fichiers, vous pouvez explorer votre dossier contenant la musique. Si vous souhaitez ajouter à la liste tous vos fichiers d'un coup, appuyez sur `Maj+a`, afin qu'il ajoute le répertoire à la liste de lecture. En tapant simplement « a » vous n'ajoutez que le fichier actuellement sélectionné. Une fois

la plupart des gens pensent à des commandes compliquées ou à des terminaux noirs...

que toute la musique et toutes les informations des balises ont été chargées, vous devriez avoir une belle liste de vos fichiers musicaux. Mais attendez, si je veux lire un morceau en particulier, suis-je obligé de faire défiler toute cette liste ? Réponse simple : non. Si vous appuyez sur « g », puis tapez le nom d'un artiste, d'une chanson, etc. les chansons de la liste de lecture sont filtrées et vous pourrez alors sélectionner ce qui vous intéresse et appuyer sur `Entrée` pour le jouer. Vous pouvez utiliser le fichier `~/moc/config` pour spécifier une configuration de départ et activer - ou non - la lecture aléatoire, la fonction « répéter », etc. Je ne vais pas exposer ici la liste pratiquement infinie des options (parce que je ne saurais pas par où commencer, il y en a tellement !).

Mon fichier `~/moc/config` peut

être consulté à :

<http://fullcirclemagazine.org/moc-config/>

Raccourcis clavier de base :

g - rechercher

barre d'espace - pause/lire

Entrée - choisir/démarrer la lecture du fichier sélectionné

tab - passer du navigateur de fichiers à la liste de lecture et vice-versa

n - chanson suivante

b - chanson précédente

C - effacer la liste de lecture

A - ajouter de façon récursive un dossier à la liste

a - ajouter un fichier à la liste

s - basculement en lecture aléatoire ou pas

h - aide

IRSSI

Maintenant, passons à irssi. Pour installer ce programme, exécutez la commande :

```
sudo apt-get install irssi
```

(on commence à être habitué ?). Une fois qu'il est installé, vous pouvez le lancer avec :

```
irssi
```

Une fois le programme démarré, vous êtes accueilli par

un joli écran blanc. Pour commencer, vous devez taper

```
/connect irc.freenode.net
```

(vous pouvez remplacer ce serveur par celui auquel vous souhaitez vous connecter). Une fois que vous y êtes, si vous savez où vous voulez aller, tapez

```
/join #ubuntu
```

(Ou, encore une fois, le canal que vous voulez). Si vous vous connectez à de multiples canaux, vous verrez une liste juste au-dessus du champ de saisie. Si vous voulez passer de la fenêtre-2 à la fenêtre-1, par exemple, pressez la touche méta (en général la touche Windows, mais Echap fonctionne tout aussi bien) suivi de 1 pour obtenir le premier écran. Si vous voulez voir les deux à la fois, vous pouvez saisir

```
/window show 1
```

ce qui affichera le premier écran en plus de l'écran dans lequel vous étiez. Il y a tellement de commandes et d'options que je ne peux pas les détailler toutes ici, mais elles sont assez bien expliquées sur le site Web (voir la section « Pour en savoir plus » pour le lien). Il y a une dernière chose dont je vais parler. Il s'agit des thèmes, car ils sont souvent

très intéressants. Tout d'abord, trouvez (ou écrivez) un thème à votre goût, puis copiez (ou créez) le fichier dans le dossier ~/.irssi/ (quelque chose comme « arcs-en-ciel.theme »).

Une fois que vous aurez créé (ou copié) le thème, vous pourrez le choisir dans irssi en utilisant la commande :

```
/set theme <nom_du_theme>
```

où <nom_du_theme> est le nom du fichier (sans l'extension .theme), ce qui ferait

```
/set theme arcs-en-ciel
```

pour l'exemple donné ci-dessus.

Commandes de base :

```
/connect <URL du serveur> - se connecter à un serveur (Freenode, DALnet, etc.)
```

```
/join <canal> - se connecter à un canal (#ubuntu, #kubuntu, etc.)
```

```
/quit - quitter irssi
```

```
/disconnect - déconnexion du serveur
```

```
/part <message d'au revoir> - quitter le canal (le message d'au revoir est facultatif)
```

J'espère que cet article vous aura donné envie d'essayer quel-

ques programmes en ligne de commande. Ils sont extrêmement légers, flexibles et il est amusant de s'en servir ! Je vous conseille de les essayer avec un gestionnaire de fenêtres en mosaïque si vous voulez être aussi léger que possible. Des choses comme Xmonad, Awesome, etc., sont toujours des bonnes expériences à tenter - en particulier si vous êtes friands de programmation et d'applications en ligne de commande. À mon avis, c'est le meilleur environnement pour les deux. Même si vous ne prévoyez pas d'utiliser ces programmes, je vous recommanderais d'avoir au moins irssi à portée de main au cas où vous auriez besoin d'aide sans que votre interface graphique soit disponible.

Pour en savoir plus :

MOC - <http://moc.daper.net>

irssi - <http://irssi.org>



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système, et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à l'adresse : lswest34@gmail.com.



COMMAND & CONQUER

Écrit par Lucas Westermann

Dans le Command & Conquer de ce mois-ci, je vais parler de plusieurs choses qu'on évoque parfois en ligne avec quelques instructions, ou d'autres qui ne fourniraient pas matière à un article à elles seules, mais qui méritent tout de même d'être mentionnées. J'espère que ce que je vais dire sera utile aux lecteurs désireux d'aller plus loin avec leur shell, ou à ceux qui veulent le personnaliser et rencontrent pourtant de temps à autre des termes qu'ils ne connaissent pas encore.

Je commencerai avec la personnalisation de votre prompt. Admettons que vous avez passé beaucoup de temps à chercher un prompt qui vous plaît pour votre terminal, que vous avez enfin pu paramétrer tous les caractères d'échappement et que vous êtes prêt à faire un essai. Voici la variable d'environnement PS1 de mon .zshrc :

```
export
PS1="%{$fg[blue]%;}&#9484;&#9472; [%{$fg[green]%;}%n%{$fg[cyan]%;}%@%{$fg[green]%;}%m%{$fg[blue]%;}:%{$fg[magenta]%;}%~%{$fg
```

```
[blue]%;}]-
%{$fg[red]%;}[%{$fg[cyan]%;}]*
on
%D%{$fg[red]%;}]%{$reset_color
%}%{$reset_color}"$'\n'"%{$fg
g[blue]%;}&#9492;&#9472;>%{$re
set_color%} "
```

ce qui donne l'image ci-dessous.

Il y a deux façons de procéder :

- 1) vous pouvez ouvrir un nouveau terminal et visualiser le shell (ce qui n'est pas une très bonne option si vous êtes en train d'exécuter quelque chose dans une console tty ou de faire trop de choses à la fois ou si vous détestez changer de shell avant d'avoir fini ce que vous êtes en train de faire) ou
- 2) vous pouvez lui appliquer la commande « source ». Je préfère cette option parce qu'elle prend immédiatement effet dans le terminal original et est assez simple à mettre en œuvre. Pour ce faire, vous utilisez la commande « source » suivie du chemin (ou simplement du nom du fichier, s'il est présent dans votre répertoire de travail courant) vers le fichier de configuration (.bashrc dans ce cas).

source .bashrc

Cette commande fonctionne avec la plupart des fichiers de configuration modifiables, mais il y a certains cas où cela ne fera rien du tout (je ne pense pas en connaître tous les usages possibles et vous devrez donc faire des expériences par vous même).

Une autre chose utile à savoir est que vous pouvez utiliser des boucles du type « while » dans le shell en cours. Par exemple, si vous voulez lister tous les fichiers d'un répertoire et les afficher chacun sur une ligne différente, vous pouvez faire ceci :

```
ls|while read line; do `cat
${line}`; done
```

Bien sûr, vous obtiendrez un message d'erreur si vous rencontrez un répertoire, mais ce problème peut être facilement résolu par un simple test (utilisant une instruction « if »). Je vous laisse chercher...

Voici quelques autres commandes que je trouve aussi très utiles :

df

qui montre l'utilisation du système de fichiers (je l'utilise généralement avec l'argument « -h », afin que l'affichage se fasse en gigaoctets).

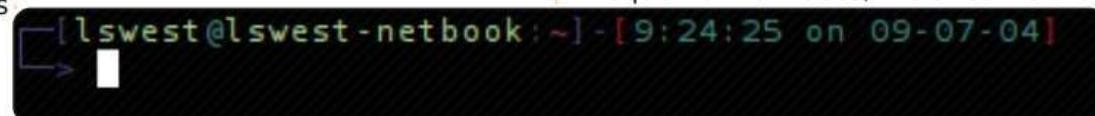
watch

laquelle exécute par défaut une commande toutes les 2 secondes (mais ce paramètre peut être modifié avec l'argument « -n <num> »).

scrot

qui est essentiellement un outil de capture d'écran en ligne de commande, mais avec un tas d'options et de possibilités (consultez sa page man pour obtenir plus d'infos, il y a bien trop d'options pour les détailler ici).

Et, bien sûr, les autres commandes dont j'ai parlé dans les numéros précédents sont aussi fréquemment utilisées, mais les trois ci-



dessus n'avaient pas encore été mentionnées, bien qu'elles soient très utiles. Une dernière chose dont je voudrais parler ici est de ce qu'est un gestionnaire de fenêtres en mosaïque « tiling window manager », parce qu'un certain nombre de développeurs que je connais les préfèrent aux gestionnaires classiques. Un gestionnaire de fenêtres en mosaïque arrange toutes les fenêtres en « carreaux » (les dimensionnant de telle sorte que les fenêtres remplissent tout l'espace disponible sur l'écran sans se recouvrir). Certains (pas tous) proposent un mode « flottant » dans lequel les fenêtres se comportent comme d'habitude (avec une taille prédéterminée, du recouvrement, etc.). La raison pour laquelle les développeurs (comme moi) préfèrent ce comportement est parce qu'il vous laisse visualiser tout votre code à la fois, avoir de nombreux scripts ouverts en même temps en vous permettant de passer de l'un à l'autre facilement. Vous pouvez ainsi avoir un terminal ouvert pour tester les commandes et un autre pour écrire le script. Mais il n'y a pas que ça, vous pouvez contrôler le gestionnaire de fenêtres en utilisant uniquement le clavier, ce qui vous permet de travailler plus efficacement parce que vous n'êtes jamais obligé d'enlever

les doigts du clavier. Quelques exemples de ces gestionnaires de fenêtres sont AwesomeWM, DWM, Xmonad, Ratpoison et Ion. J'utilise Awesome parce qu'il propose un mode flottant dans lequel les fenêtres ne sont pas toujours dessus ou toujours dessous mais les deux à la fois (NDT : une fenêtre entre deux fenêtres ?) (Xmonad semble ne permettre à l'origine que l'un ou l'autre et je n'ai pas réussi à le feinter). Cependant, au cas où un lecteur voudrait se lancer, il y a beaucoup d'options et la plupart sont bien documentées.

Pour en savoir plus :

Awesome -

<http://awesome.naquadah.org/>

Xmonad - <http://www.xmonad.org/>

Ratpoison -

<http://www.nongnu.org/ratpoison/>

Ion - <http://modeemi.fi/~tuomov/ion/>

DWM - <http://dwm.suckless.org/>



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système, et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à : lswest34@gmail.com.



Live-Office est une suite libre et professionnelle, en ligne, d'applications collaboratives qui facilite l'organisation de toutes vos données personnelles. Il est inutile d'installer des

applications supplémentaires sur votre ordinateur, car tous les composants sont installés sur le serveur Web. On a simplement besoin d'une connexion internet ainsi que d'un navigateur avec JavaScript activé. Vous pouvez accéder à Live-Office sur notre serveur, ou l'installer facilement sur votre propre serveur. Toutes vos données personnelles seront stockées de manière centralisée et sécurisée dans une base de données en ligne, soit la nôtre, soit la vôtre, ce qui vous offrira des modules et des widgets divers qui vous permettront de sauvegarder toutes vos données personnelles en un seul endroit. Événements, choses à faire, contacts, favoris, documents et notes, ne sont que quelques exemples parmi d'autres. Vous pouvez accéder à ces modules et widgets au moyen d'une interface utilisateur haut de gamme et intuitive. Cette fonctionnalité unique est la clé qui vous aide à rassembler et à visualiser toutes vos informations en même temps. Par exemple, vous pouvez ouvrir votre agenda et votre répertoire côte à côte, ajouter un contact dans la fenêtre de votre carnet d'adresses et ajouter un rendez-vous dans la fenêtre de votre agenda.

Avec Live-Office, vous pouvez personnaliser votre environnement personnel. Vous n'aimez pas la couleur de votre bureau ? Changez-la pour un papier peint. Vous préférez utiliser les applications dans votre langue maternelle ? Live-Office supporte de nombreuses langues.

De nouveaux modules et widgets seront bientôt ajoutés, tels que budget et factures et un trousseau de clefs, ainsi que de nouvelles langues. En plus, nous travaillons sur une « Live-Office Desktop Edition » - une version hors-ligne écrite en Java. Cette version vous permettra de gérer toutes vos données personnelles hors-ligne et de les synchroniser au besoin.

Contactez-nous ou visitez notre site internet pour en savoir plus sur la manière dont vous pouvez contribuer à Live-Office :

info@live-office.net ou :

<http://www.live-office.net>





Récemment, un lecteur m'a demandé de faire un article sur la structure de base du système d'aide (commande « help » ou « man », pour manuel). Afin d'illustrer cet article, je vais prendre en exemple la commande « ping » en m'intéressant aux commandes « help » et « man ». La commande :

`ping -h`

affichera l'aide de la commande pour ping (le commutateur « -h », ainsi que l'argument « --help », sont de facto la valeur par défaut). Les informations données par l'aide devraient ressembler à ce qui se trouve dans l'encart en haut à droite

Les premiers crochets contenant « -LRUbdfnqrvVaA » vous donnent une liste des commutateurs éventuels qui ne nécessitent pas d'arguments (principalement parce qu'ils forment les données de sortie) ; pour avoir une explication de l'action de chacun de ces commutateurs, il faut consulter la page du manuel. La série de crochets suivante qui contient une combinaison commutateur et

valeur, par exemple « -c count », vous demande de remplacer « count » par une vraie valeur. Ces mots ont pour but de vous donner une idée de la fonction du commutateur.

Idéalement, la page d'aide est conçue comme une référence rapide, au cas où vous ne savez pas quel commutateur correspond aux données que vous souhaitez utiliser. Toutefois, si vous découvrez la commande et que vous ne connaissez pas l'action de la moitié des commutateurs. Il est préférable de lire la page du manuel, car elle offre, notamment, une explication pour chaque commutateur, ainsi que ses différentes utilisations possibles et un site Web d'aide. Quelques-unes des pages d'aide décrivent les arguments et

```
Usage: ping [-LRUbdfnqrvVaA] [-c count] [-i interval] [-w deadline]
          [-p pattern] [-s packetsize] [-t ttl] [-I interface or address]
          [-M mtu discovery hint] [-S sndbuf]
          [-T timestamp option] [-Q tos] [hop1 ...] destination
```

count - nombre de tentative de « ping »

interval - durée entre les « ping »

deadline - durée maximale d'attente d'une réponse

pattern - permet de compléter le « ping » à l'aide d'un motif de mots (bytes)

packetsize - définit la taille du paquet à envoyer

ttl - définit la durée de vie du protocole internet

interface or address - définit l'adresse IP ou le périphérique source

mtu discovery hint - « do », « want » ou « dont » sont possibles

sndbuf - définit la taille du tampon d'envoi

timestamp option - permet de définir des options d'horodates IP spéciales

tos - définit l'option de la qualité du service (valeur décimale ou hexadécimale acceptée)

hop1... - peut être une liste de destinations

commutateurs, et fournissent d'autres d'informations complémentaires, mais ce n'est pas toujours le cas.

Pour afficher la page du manuel, utilisez cette commande :

`man ping`

La section Synopsis affiche quelque chose de similaire à la commande d'aide, mais elle est suivie par une

section de Description qui explique ce que la commande devrait faire, ou comment vous devriez l'utiliser. La section Options répertorie tous les commutateurs avec une courte description de la fonction de chacun. Pour naviguer dans le reste de la page du manuel, utilisez les flèches haut et bas, ou page précédente et page suivante. Le texte tout à la fin se présente comme ceci : « Manual page ping(8) line 13/221 15% ». Il vous

COMMAND & CONQUER

indique que vous consultez la page du manuel concernant la commande ping, et que vous êtes à la ligne 13 sur 221, soit à 15% du document. Une fois que vous avez fini de visualiser le reste de la section Options, vous arrivez à une section appelée « ICMP Packet Details », qui définit ceux-ci. Il y a quelques définitions supplémentaires avant que vous n'arriviez à la section Bogues, mais chacune est assez complète et je ne vais donc pas tout expliquer ici. On arrive ensuite à la liste des bogues actuellement ouverts (les bogues connus qui n'ont pas été corrigés dans cette version-ci), puis à la section « See Also : » (voir aussi), qui présente d'autres commandes à consulter éventuellement, parce qu'elles pourraient être utiles combinées avec la commande actuelle. La section « History » (historique) n'a pas besoin d'explication, tandis que les sections Security (sécurité) et Availability (disponibilité) vous donnent un peu plus de renseignements sur la commande.

Toutes les pages « man » ne sont pas disposées de la même manière, mais elles suivent les mêmes conventions, qui sont :

Name (Nom)

Synopsis (Synopsis)

Description (Description)

Options (Options)
Informations pertinentes à la commande (définitions, explications, etc.)
Bugs (Bogues)
See Also (Voir aussi)
History (Historique)
Security (Sécurité), si applicable
Availability (Disponibilité)

Ceci est utile au cas où vous voudriez chercher une section spécifique de la page d'aide, puisque vous saurez exactement où aller. En outre, si jamais vous écrivez votre propre outil, ou si vous souhaitez ajouter une page de manuel à un script que vous avez écrit, vous saurez ainsi comment le formater. Espérons que cet article vous aura aidé à mieux comprendre la façon dont les pages du manuel fonctionnent, et comment donner un sens aux pages d'aide qui sont souvent confuses.



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système, et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à : lswest34@gmail.com.

ATLANTA LINUX FEST

SOFTWARE FREEDOM DAY!

DATE:
**Saturday
September 19, 2009**

WEBSITE:
atlantalinuxfest.org

COST:
Free!

ADDRESS:
**IBM
4111 Northside Pkwy
Atlanta, GA 30327**

All lovers of Linux and Open Source Software are invited to Atlanta Linux Fest: the place to learn, make new friends, and have fun! 🐧

Atlanta Linux Fest
FOR MORE INFO LOG ON TO
atlantalinuxfest.org



Est-ce que vous avez déjà tapé sur une touche spéciale du clavier de votre ordinateur portable pour vous rendre compte qu'elle n'est tout simplement reliée à rien et, après avoir fait des recherches, que le programme qui gère les raccourcis ne reconnaît pas la touche ? Eh bien, je me suis retrouvé face à un problème similaire en configurant Openbox sur mon portable, alors je me suis dit que j'allais vous montrer comment assigner des touches à des fonctionnalités, même si elles ne semblent pas reconnues.

Première étape : reconnaissance de la touche

Tout d'abord, il nous faut vérifier si la touche est bien reconnue par le noyau. Ouvrez un terminal et exécutez la commande suivante :

```
xeV | grep -A2 --line-buffered '^KeyRelease' | sed -n 's/^.*keycode \([0-9]*\) .* (.*) \(.*)$/\1 \2/p'
```

Cela aura pour effet d'afficher le

code de la touche suivi de son nom (XF86AudioMute, XF86MonBrightnessDown, a, b, etc.). Si « NoSymbol » s'affiche après le code de la touche, c'est qu'il n'y a aucune fonction liée à cette touche pour l'instant ; dans ce cas vous pouvez passer directement à l'étape deux. Si la touche n'affiche rien, il est temps d'essayer showkey.

Affichez tty0 (avec la combinaison ctr + alt + F1) et connectez-vous avec votre nom d'utilisateur. Une fois connecté, entrez la commande :

```
showkey
```

Ce programme va renvoyer le code des touches enfoncées et s'arrêter automatiquement 10 secondes après la dernière touche activée. Une fois que le programme est lancé, enfoncez les touches que vous voulez tester et notez chaque code retourné. Si aucune de ces méthodes ne retourne le code de la touche, nous allons regarder si la touche a un scancode.

Pour ce faire, enfoncez la touche à tester et vérifiez ensuite le contenu de dmesg avec la commande :

```
dmesg|tail -5
```

Si quelque chose dans ce genre apparaît :

```
atkbd.c: Unknown key pressed (translated set 2, code 0xf1 on isa0060/serio0).
```

```
atkbd.c: Use 'setkeycodes e071 <keycode>' to make it known.
```

vous pouvez relier le scancode au code de la touche en utilisant HAL ou setkeycodes (outil de noyau), comme vous l'indique dmesg. La section « Lectures complémentaires » à la fin de cet article vous propose un lien vers la page des bizarreries de mapping de touches d'HAL. Je ne donnerai pas plus de détails dans cet article étant donné que, d'après mon expérience, il est relativement rare d'avoir besoin de le faire.

Étape deux : assigner les codes des touches

Je vais me concentrer sur comment assigner les codes aux touches dans Xorg, étant donné que

la plupart des touches multimédia ne sont pas nécessaires dans la console tty0. Pour commencer, vous devez créer le fichier .Xmodmap. Vous pouvez le faire en utilisant la commande touch ou simplement en éditant le fichier puis en le sauvegardant avec gedit. Les entrées dans le fichier doivent respecter ce format :

```
keycode <Xkeycode> = keysymbol
```

Quelques exemples :

```
keycode 153 = XF86MonBrightnessDown
```

```
keycode 154 = XF8MonBrightnessUp
```

Étape trois : tester les codes des touches

Commencez par lancer cette commande :

```
xmodmap ~/.Xmodmap
```

Vous devriez maintenant être en mesure d'ajouter les touches pour n'importe quelle fonction dont vous



avez besoin. Si ce n'est pas le cas, vérifiez les noms des codes et des symboles des touches au cas où vous auriez fait une faute de frappe. Une liste complète des symboles peut être trouvée ici :

`/usr/include/X11/keysymdef.h`
Et pour les fonctionnalités additionnelles :

`/usr/include/X11/XKeySymDB`

Une fois que vous êtes sûr que la touche fonctionne, passez à l'étape quatre.

Étape quatre : rendre cela définitif

Pour rendre les modifications définitives, vous devez lancer la commande `xmodmap` à chaque fois

que vous vous connectez. Je vous recommande de l'ajouter à votre `.xprofile`.

Un outil alternatif pour `xmodmap` est `xbindkeys` et il est assez simple. Une interface graphique est disponible : `xbindkeys_config`, mais je ne suis pas sûr qu'elle soit présente dans les dépôts d'Ubuntu.

Lectures complémentaires :

Bizareries de mapping de touches d'HAL : <http://people.freedesktop.org/~hughsient/quirk/quirk-keymap-index.html>



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système, et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez envoyer un courriel à l'adresse : lswest34@gmail.com.

CRITIQUE RAPIDE - PREY

Dans le monde du jeu vidéo FPS il y a très peu d'innovations. La plupart du temps un nouveau jeu FPS comportera des éléments qui ont fait le succès d'autres jeux plus anciens de ce genre.

Ce n'est pas le cas de *Prey*. *Prey* est unique. *Prey* est différent de tout ce à quoi vous avez joué avant. Porté récemment sous Linux, ce jeu, un de mes préférés, trouve actuellement un regain de vitalité.



Vous jouez un guerrier Cherokee nommé Domasi Tawodi (alias Tommy), un homme qui veut oublier son héritage de Cherokee, quitter la Réserve et entrer dans le monde civilisé, mais il y a un problème : sa petite amie Jen veut rester, parce que c'est là son pays. Tout à coup, alors qu'ils en discutent dans un bar, ils sont attirés vers le haut par un vaisseau alien qui les emporte à bord, lui, Jen et son grand-père afin que ses occupants puissent se nourrir d'eux.

Bien sûr, Tommy ne va pas abandonner sa vie si facilement, c'est pourquoi il essaye de sauver son grand-père et sa petite amie.

Une chose qui rend ce jeu remarquable (mis à part le fait qu'il utilise un moteur Doom 3 fortement modifié) c'est son utilisation de la pesanteur (si vous jetez un œil à quelques captures d'écran vous comprendrez pourquoi) et des portails (se tirer dessus soi-même si on ne sait pas ce que l'on fait est rendu tout à fait possible). Ceux-ci sont utilisés dans un certain nombre d'énigmes courtes ou longues, mais rien de bloquant pour le joueur moyen occasionnel. Une autre facette du jeu que j'ai adorée était le fait que, à partir d'un certain point du jeu, vous ne pouvez pas mourir. Oui vous avez bien lu, « vous ne pouvez pas mourir », ainsi vous ne « mourrez » plus, alors appuyez rapidement sur votre touche de chargement rapide et réessayez avec une minuscule quantité de vie. Je ne veux pas en dire trop sur ce jeu à part le fait qu'il est vraiment génial.

Plus de dix ans de développement et voilà ce que nous avons : un jeu vraiment habile. C'est un des rares exemples de jeu vidéo moderne qui, je pense, va s'imposer dans le livre des records comme un classique incontournable.

Christopher Hart

Note : 9/10



Revenu récemment de mes vacances d'été, après une mise à jour d'environ 300 paquets, j'ai remarqué dans Conky que la partition principale était quasiment pleine. Du coup, je me suis dit qu'il pourrait être utile d'écrire un article reprenant les quelques trucs que j'ai découverts au cours des années et que j'utilise quand un disque est saturé.

Pour commencer, nous devons probablement contrôler l'espace disponible sur le disque. Pour cela, j'utilise la commande « df » dans un terminal. Gparted permet aussi ce type de vérification, mais je me concentrerai sur la ligne de commande. Alors, allons-y. Pour voir une liste de tous les systèmes de fichiers montés et leur taux d'utilisation, saisissez :

```
df -h
```

Une liste des partitions montées s'affiche, avec l'espace utilisé, la quantité disponible, le pourcentage d'utilisation et le point de montage. Je trouve

acceptable d'utiliser un disque jusqu'à 90 % de sa capacité (votre dossier personnel peut sans problème atteindre les 95 %). La partition racine nécessite un peu d'espace libre pour les fichiers logs, les dossiers de la racine, etc. et vous serez prévenu quand elle approchera de la saturation (tout l'espace est utilisé à l'exception de celui réservé aux logs). En général, vous ne pouvez alors plus rien installer ni même déplacer un fichier, ce qui n'est guère souhaitable. Si vous ne pouvez libérer suffisamment d'espace, il sera peut-être nécessaire de redimensionner la partition elle-même pour l'agrandir. Généralement, 10 Go suffisent pour ma partition principale (ma partition home fait généralement 25 Go ou plus).

Si c'est la partition root qui commence à être pleine, votre première étape devrait être de vider le cache des paquets (aptitude, apt-get et synaptic enregistrent les paquets téléchargés dans un cache, de sorte qu'ils n'aient pas besoin de

les télécharger à nouveau pour pouvoir les réinstaller). Ouvrez une nouvelle console (en laissant visible le résultat de « df -h ») et saisissez

```
sudo aptitude autoclean
```

ou

```
sudo apt-get autoclean
```

pour effacer tous les paquets obsolètes qui se trouvent dans le cache. Ensuite, j'exécute à nouveau « df -h » pour connaître la quantité d'espace ainsi libérée. Si suffisamment d'espace a été libéré, je laisse les choses en l'état et je fais autre chose. Si trop peu d'espace a été libéré et que vous utilisez Ubuntu sans le dépôt « backports », sans « bêta », ni beaucoup de paquets à la pointe du progrès, vous pourriez envisager de vider l'ensemble du cache. Je faisais cela avant de passer à Arch, car Ubuntu teste les paquets à fond et laisse les versions stables dans les dépôts normaux. Néanmoins, si vous utilisez les dépôts getdeb ou similaires, je vous recommande

de ne pas le faire au cas où un problème quelconque vous obligerait à revenir à une version antérieure. Cependant, ne le faites pas de façon régulière, car vous pourriez un jour tout simplement avoir besoin de ces paquets en cache, pour une raison quelconque. Vous pouvez effacer tous les paquets du cache en saisissant :

```
sudo aptitude clean
```

ou

```
sudo apt-get clean
```

Notez accessoirement que si je liste les deux commandes aptitude et apt-get, c'est que je préfère utiliser aptitude pour le nettoyage, et n'ayant pas utilisé Ubuntu depuis quelques temps, j'ignore si apt-get le fait désormais de manière similaire ou non.

Après avoir effacé les paquets, vous devriez disposer d'un peu plus d'espace. Si vous en voulez encore davantage, ou que la partition principale n'était pas celle qui avait besoin de plus

d'espace, vous pouvez lister vos fichiers et dossiers par taille et supprimer manuellement les gros fichiers dont vous n'avez plus besoin (de vieux fichiers .iso, des archives, des thèmes d'icônes, etc.). J'ai récupéré cette astuce dans « Linux Journal » (je ne sais plus quel numéro) - cela fonctionne bien. D'abord, si vous avez deux partitions distinctes pour root et home et que vous voulez récupérer de l'espace sur la partition root, saisissez :

```
cd /  
du -ckx|sort -n
```

Cette commande change d'abord de dossier (cd) pour ouvrir la partition racine, puis affiche l'utilisation du disque (« du ») par blocs de taille de 1 ko (« -k ») et affiche le total général (« -c ») pour la partition (sans se déplacer dans la partition home, etc.). Tout ceci sera trié du plus petit au plus grand (le dernier fichier est le plus grand) grâce à la commande « sort -n » à laquelle est envoyée la sortie de « du ». Je vous recommande de ne rien effacer de votre partition principale sans l'avoir sérieusement étudiée (à moins qu'il s'agisse du cache d'un logiciel

que vous n'utilisez plus et que vous pouvez retirer en sécurité - mais vérifiez d'abord). Sachez que vous pouvez provoquer de sérieuses erreurs système en les supprimant. Ensuite, pour afficher les informations relatives à votre dossier/partition personnel, saisissez :

```
cd ~  
du -ck|sort -n
```

Cette commande retourne la taille de tous les fichiers et dossiers dans votre dossier personnel, triés du plus petit au plus grand. Après avoir découvert les plus grands dossiers, vous pouvez voir la taille de ces dossiers (en Ko/Mo/Go) en saisissant :

```
du -h ~/<foldername>/
```

L'option « -h » signifie lisible par un humain. Le tilde (~) remplace /home/<utilisateur>/ (quelques frappes économisées), et bien sûr, vous remplacerez « <foldername> » par le nom du dossier dont vous souhaitez obtenir les renseignements. Une fois que vous aurez découvert les fichiers et dossiers qui monopolisent votre espace, vous pouvez choisir de supprimer ceux dont

vous n'avez plus besoin (ou de sauvegarder sur un autre système de sauvegarde ceux que vous voulez garder mais n'utilisez plus). Après tout ceci, vous devriez avoir libéré un peu de place et être en mesure de reprendre l'installation de paquets !

Je vous recommande vivement de parcourir de temps en temps la liste des paquets installés et de décider si vous voulez vous débarrasser de ceux que vous n'utilisez pas. Ceci est assez simple à faire, avec aptitude (qui affiche un « i » si les paquets sont installés). Saisissez :

```
sudo aptitude search '~i'
```

ou

```
sudo apt-cache search '~i'
```

Une courte note sur la commande apt-cache : je ne suis pas certain qu'elle fonctionne de la même manière qu'aptitude. Aussi, avant de désinstaller un paquet que vous ne vous rappelez pas avoir installé, vérifiez ! Ce pourrait bien être un paquet nécessaire à Ubuntu.

Attention ! Soyez extrêmement prudent quand vous supprimez ou désinstallez des choses, plus

particulièrement si vous avez la moindre incertitude les concernant. Mieux vaut prévenir que guérir.

Je suis ouvert aux demandes d'articles, alors, si vous êtes confronté à un problème de ligne de commande qui puisse concerner d'autres personnes, envoyez-moi un courriel (en anglais) en le décrivant et je ferai peut-être un article dessus. Le mérite d'avoir suggéré le thème vous sera attribué, bien entendu, sauf contordre.

Lectures complémentaires :

Anti-sèche apt-get :

<http://www.cyberciti.biz/tips/linux-debian-package-management-cheat-sheet.html>

Informations sur aptitude search '~i' :

<http://www.linuxquestions.org/questions/debian-26/aptitude-how-to-get-a-list-of-all-installed-packages-458119/#post2310207>



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système, et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à l'adresse :

lswest34@gmail.com



Avant que je n'entame le sujet de cet article, j'aimerais prendre un moment pour remercier David Rowell qui m'a signalé que le répertoire des vignettes peut être également très gourmand en espace disque sur certains systèmes (ceci ne concerne que les systèmes où des vignettes sont générées). Dans Ubuntu le répertoire par défaut est `~/thumbnails`. Toutefois, je pense que Thunar (sur Xubuntu) les stocke dans un répertoire différent - de même que Konqueror sur Kubuntu. Les miniatures ne sont pas supprimées quand l'image/la vidéo concernée est supprimée, du moins c'était le cas dans Gnome 2.24/2.26. Donc, si vous stockez/avez stocké beaucoup de médias sur votre disque dur, il y a de fortes chances que votre dossier de miniatures soit plutôt important. Pour remédier à ce problème, il suffit de supprimer ce répertoire comme ceci :

```
rm -r ~/.thumbnails
```

et la prochaine fois que vous ouvrirez un dossier contenant des médias, les miniatures seront régénérées, ce qui peut prendre quelques minutes (en fonction du nombre de fichiers et du proces-

seur de votre système). Si vous n'êtes pas sûr de la taille de votre dossier de miniatures, vous pouvez le vérifier en utilisant l'astuce de mon article du mois dernier :

```
du -h ~/.thumbnails
```

Passons maintenant au sujet du mois. Je sais que de nos jours beaucoup de personnes utilisent Ubuntu, ou une autre distribution Linux, sur des ordinateurs portables, aussi j'ai pensé qu'il serait intéressant d'expliquer comment désactiver la gestion de l'énergie des disques durs, qui peut provoquer beaucoup d'usure sur les disques de portables. Il y a des discussions concernant ce sujet sur la plupart des forums dédiés aux distributions, et, autant que je sache, cela n'a toujours pas été résolu. L'inconvénient de cette manipulation, c'est que le disque dur ne s'arrête jamais de tourner. Cela peut causer des pertes de données si le portable tombe (en particulier si le disque dur est en train d'écrire), et cela augmente de quelques degrés la température du portable, le disque dur générant de la chaleur. Le côté positif, c'est que le disque dur fonctionnera plus longtemps que si la gestion de l'énergie est

activée et les performances du disque dur augmenteront légèrement. On va voir comment déterminer si votre portable est affecté par ce bogue et comment désactiver la gestion de l'énergie. Je vous montrerai également comment utiliser smartmontools pour contrôler l'état de votre disque dur.

Avant de contrôler des valeurs, vous devez installer l'outil que l'on va utiliser. Smartmontools est dans le dépôt de la plupart des distributions, y compris, bien entendu, Ubuntu. Pour l'installer, exécutez la commande suivante :

```
sudo apt-get install smartmontools
```

Une fois l'installation terminée, vous contrôlez les valeurs S.M.A.R.T. (Self-Monitoring, Analysis, and Reporting Technology = technologie d'auto-surveillance, d'analyse et de signalement) de votre disque dur en exécutant la commande suivante :

```
sudo smartctl -H /dev/sda
```

Vous devez remplacer `/dev/sda` par le disque dur que vous souhaitez contrôler. Cela retournera des

informations dans le format suivant :

```
smartctl version 5.38
[x86_64-unknown-linux-gnu]
Copyright (C) 2002-8 Bruce
Allen
Home page is
http://smartmontools.sourceforge.net/
```

```
=== START OF READ SMART
DATA SECTION ===
SMART overall-health self-
assessment test result:
PASSED
```

Comme vous pouvez le voir, les données SMART de mon portable n'ont pas déclenché d'alarmes (ce qui signifie que l'état de mon disque dur est bon). Si cela dit que le disque dur a échoué aux tests, vous devriez penser à le changer très prochainement. Si cela dit que votre disque dur ne supporte pas SMART, alors vous pouvez arrêter de vous en faire, car vous ne pourrez changer aucun de ses paramètres.

Avant de passer à la prochaine section, je vais prendre le temps de vous signaler que vous devriez lire le lien vers la discussion de l'UbuntuForums ci-dessous avant d'appliquer ces correctifs, car cela

ne devrait effectivement être fait que si vous comprenez bien ce qui se passe. Vous devez également prendre en compte l'âge du disque dur, etc. Si vous décidez de suivre les recommandations de cet article, vous le faites à vos risques et périls. J'écris cet article parce que j'ai pu constater que la plupart des portables que j'ai utilisés avaient besoin de ce correctif. Mon expérience comprend l'explication des différentes étapes de ce correctif aux propriétaires des portables. Attention : certains portables récents peuvent ne pas avoir besoin de ce correctif et, dans ce cas, si vous installez le correctif malgré tout, vous pourriez réduire la durée de vie du disque dur. Il est possible de trouver sur le Web des listes de portables qui souffrent du problème de gestion de l'énergie. Elles vous diront si d'autres possesseurs du même modèle que votre portable ont signalé ce problème sur leur machine.

Pour vérifier le début du `Load_Cycle_Count`, saisissez la commande suivante :

```
sudo smartctl -a  
/dev/sda|grep  
Load_Cycle_Count @
```

Cela donnera une ou deux lignes de code comme celle-ci :

```
225 Load_Cycle_Count 0x0032  
099 099 000 Old_age Always -  
14091
```

Le premier nombre est le numéro d'identificateur, le nom est celui de l'attribut :

« `ATTRIBUTE_NAME` », la valeur hexadécimale est le drapeau « `FLAG` », le premier nombre (099 ici) est la valeur « `VALUE` », le `WORST` est le 099 suivant, le 000 est le `THRESH`, `Old_age` est le `TYPE`, `Always` se rapporte à `UPDATED`, le « - » est la colonne `WHEN_FAILED` (qui serait une date, si le disque dur avait échoué), et le 14091 est la `RAW_VALUE`. Je vais maintenant expliquer ce que représentent certains de ces termes. `VALUE` est la valeur sous la forme d'un pourcentage de `SMARTCTL`. `WORST` est la valeur la plus faible stockée pendant la vie du disque dur, et `THRESH` est la valeur seuil qui indique à `SMART` que le disque dur a échoué (donc quand la valeur atteint 000, le disque dur est défaillant). `TYPE` se rapporte au type de `THRESH` (il y a le choix entre `Pre-fail`, qui vous avertit avant que le disque dur ne soit défaillant, et `Old_age`, qui indiquera simplement que le disque dur a dépassé sa durée de vie). `UPDATED` correspond à combien de fois / sous quelles conditions l'attribut est mis à

jour, `WHEN_FAILED` montre la date à laquelle l'attribut a dépassé `THRESH`, et `RAW_VALUE`, combien de fois c'est arrivé.

Peu importe, enregistrez votre `RAW_VALUE` quelque part pour la mettre en sécurité, et contrôler à nouveau cette valeur plus tard. La meilleure façon de la contrôler serait d'écrire un script à exécuter en `root` dans `CRON` une fois par jour à la même heure, ce qui vous donnerait une idée de la fréquence de son augmentation. Toutefois, vous pouvez également contrôler manuellement comment elle évolue pendant 5 minutes, etc. Si elle augmente de plus de 5 en 5 minutes alors que vous utilisez votre portable, cela signifie probablement que la valeur retournée n'est pas bonne, et vous pourriez alors diviser la `RAW_VALUE` que vous avez par l'augmentation (donc si ça augmente de 10 chaque minute, divisez par 10). Une fois que vous aurez une idée de l'augmentation moyenne journalière, vous devez alors calculer cette valeur dans 3 ans (durée de vie approximative d'un disque dur), en prenant en compte, bien sûr, l'âge du portable ! Si la valeur obtenue est au-dessous de la valeur de `Load_cycle_count` que le disque dur devrait pouvoir atteindre (cette valeur est généralement de 600 000 mais vous



devriez vérifier sur Google ce qu'il en est de la limite du `Load_cycle` de votre disque dur juste pour être sûr), alors vous ne devriez pas avoir besoin du correctif. Si, toutefois, le résultat dépasse allègrement la valeur limite, vous devriez appliquer le correctif pour que votre disque dur fonctionne aussi longtemps que possible. Par exemple, mon Samsung n110 (sous ArchLinux) augmente d'environ 1 par minute, donc par jour cela augmente de 1 440, $1\,440 * 365 = 525\,600$, $525\,600 * 3 = 1\,576\,800$. Je n'ai cependant pas tenu compte du

fait que mon portable a déjà 4 mois. Comme la valeur est très élevée, je n'ai pas pris la peine de chercher une valeur plus juste, vu que cela ne fera pas vraiment de différence. Cette valeur est bien au-delà de toute limite raisonnable pour un disque dur, aussi j'ai désactivé la gestion de l'énergie. Pour ce faire, exécutez cette commande :

```
sudo hdparm -B 255 /dev/sda
```

Ou, si vous voulez juste utiliser la valeur la plus basse possible pour ce paramètre (attendre le plus longtemps possible avant d'entrer en mode d'économie d'énergie), exécutez :

```
sudo hdparm -B 254 /dev/sda
```

Si un jour vous souhaitez défaire ce que vous venez de paramétrer, la valeur par défaut de la gestion de l'énergie pour la plupart des disques durs est 128, donc en exécutant :

```
sudo hdparm -B 128 /dev/sda
```

vous remettrez le paramétrage de la gestion de l'énergie à sa valeur par défaut.

L'essentiel de ce que je souhaitais aborder est maintenant fait. Si ce correctif fonctionne pour vous et réduit le Load_Count, alors

vous aurez peut-être besoin d'ajouter un script pour qu'il soit exécuté à chaque démarrage, mais c'est expliqué dans la discussion que je donne ci-dessous. Je conseille vivement à tout lecteur qui n'est pas sûr à 100 % que ce soit requis pour son disque dur de lire au moins une partie de la discussion, pour qu'il puisse ainsi avoir une meilleure compréhension du processus.

Pour ceux d'entre vous qui se demandent pourquoi j'ai fourni cette information dans un article après avoir averti à plusieurs reprises le lecteur qu'elle ne devrait pas être utilisée à la légère, la réponse est simple : la commande smartctl est très utile. Elle peut vous donner beaucoup de renseignements sur votre disque dur, et elle peut vous informer sur la durée de vie présumée de votre disque dur. J'ai ajouté les renseignements sur la gestion de l'énergie (APM) tout simplement parce qu'elle utilise pas mal des commandes dont je me sers pour surveiller les informations de durée de vie d'un disque dur, et aussi parce que ce sont des choses utiles à connaître. Je ne dis pas que n'importe qui devrait suivre ces instructions ; j'avertis le lecteur des éventuels problèmes, et j'offre une possibilité de le vérifier/corriger, dans le cas où il

le jugerait nécessaire. Si acheter un nouveau portable / disque dur de portable une fois par an vous convient, alors il y a de fortes chances pour que ce correctif ne vous intéresse pas. Cela étant dit, peu de gens font de tels achats. J'espère que l'introduction à smartmontools était utile à tous et que l'explication du problème « Load_Cycle » était utile pour certains (avec un peu de chance, un peu moins qu'il y a un an ou deux, mais qui sait ?).

Pour conclure cet article, je suis, comme toujours, ouvert aux suggestions, questions, commentaires, opinions, et presque n'importe quoi en rapport avec la ligne de commande (CLI). Si vous désirez vous exprimer à ce sujet, n'hésitez pas à m'envoyer un courriel (NdT : en anglais) à : lswest34@gmail.com. N'oubliez pas d'inclure le mot « FCM » dans le titre et de faire référence à Command & Conquer dans le sujet du message (juste pour être sûr que je le lise). J'aimerais également prendre un moment pour signaler que c'est mon dixième article Command & Conquer. Merci à tous ceux qui ont suivi cette série d'articles depuis que j'ai commencé à l'écrire, à la suite de Robert Clipsham.

Pour aller plus loin :

Official Ubuntu thread on load_cycle_count :
<http://ubuntuforums.org/showthread.php?p=5031046>

Pour la page de « man » de hdparm :

```
man hdparm
```

Pour la page de « man » de smartctl :

```
man smartctl
```



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système, et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à l'adresse :
lswest34@gmail.com.



COMMAND & CONQUER

Écrit par Lucas Westermann

Avant de commencer le sujet de ce mois, je dois vous avouer une erreur ! Un lecteur, Stefan Eike, a remarqué que j'avais oublié le mois dernier un 't' dans la commande :

```
sudo smartctl -H /dev/sda
```

Merci donc à Stefan de l'avoir remarqué, et mes excuses à tous ceux qui ont pu avoir des tracas avec cette commande.

J'ai reçu un e-mail du correcteur Brian Jenkins le 15 novembre, suggérant qu'un article sur le GNU Screen serait sympa à lire, car il avait commencé à l'utiliser et le trouvait des plus utile. Donc, Brian, voici ton article ! Je dois le remercier à nouveau pour m'avoir rappelé Screen. C'est comme si j'avais toujours fait en sorte de l'oublier au moment de choisir un sujet d'article. Après avoir reçu sa suggestion et convenu que c'était une bonne idée d'en faire un article ou deux (je ferai vraisemblablement une suite le mois prochain avec un peu plus d'informations sur Screen), je me mis à utiliser Screen

le plus possible pendant les semaines qui suivirent, et à le configurer du mieux possible - après tout, on ne peut pas écrire à propos d'un programme que l'on n'utilise pas !

Dans cet article, je me concentrerai sur l'installation, l'utilisation (raccourcis clavier, etc.), la configuration d'un fichier .screenrc, et les avantages et inconvénients de Screen. L'article du mois prochain parlera des utilisations plus avancées de Screen (sessions multi-utilisateurs, encapsulation de Screen dans SSH, etc.). Ainsi, tout le monde devrait avoir les connaissances nécessaires pour comprendre le prochain article et je pourrai me consacrer aux pourquoi et comment plus qu'à l'usage de Screen. Bon, pour démarrer, qu'est-ce que GNU Screen ? C'est un multiplexeur de terminal. Si cela ne vous dit strictement rien, un multiplexeur de terminal sert essentiellement à créer une série de terminaux « virtuels » dans un émulateur de terminal/écran tty, et ces terminaux virtuels peuvent être attachés/rattachés dans un nou-

veau terminal, avec un compte différent, etc. Vous vous posez peut-être la question : « pourquoi ne pas simplement ouvrir deux terminaux ou plus ? » Ceci marche, et, je dois l'admettre, j'ai l'habitude d'utiliser de multiples terminaux, mais Screen vous offre la possibilité de plusieurs terminaux dans une seule session, ce qui revient un peu à utiliser des onglets (oui, je sais il existe aussi des terminaux à onglets). Cela dit, Screen vous permet aussi de détacher et rattacher la session complète (onglets inclus) dans un nouveau terminal, avec un compte utilisateur différent, ou dans une fenêtre tty.

Évidemment, la meilleure façon de savoir ce qu'est Screen est de l'installer et de l'utiliser. Pour installer Screen sur votre système, vous pouvez taper :

```
sudo apt-get install screen
```

Une fois ceci fait, vous pouvez commencer à goûter à Screen en tapant simplement :

screen

Que voyez-vous s'ouvrir... un terminal vide ? Screen ressemble exactement à un terminal (s'il est lancé sans argument ni configuration), mais vous pouvez voir que c'est réellement Screen en tapant C-a d (« Ctrl-a » puis « d »). Et vous verrez maintenant le terminal que vous aviez ouvert auparavant montrant une ligne :

[detached]

Ce qui signifie simplement que la session qui a été démarrée a été détachée et non pas tuée.

Pour une liste complète des raccourcis clavier, suivre le lien du paragraphe « Pour Approfondir ». En voici quelques-uns que j'utilise beaucoup personnellement :

Ctrl + a, d – detache un écran

Ctrl + a, 0-9 – bascule vers ce terminal virtuel à l'intérieur d'une session écran

COMMAND & CONQUER

Ctrl + a, Ctrl + a –
retourne à la fenêtre
précédente

Ctrl + a, Ctrl + c – crée
une nouvelle fenêtre avec un
shell et bascule vers celle-
ci.

Ctrl + a, k – ferme la
fenêtre actuelle

Une fois que vous aurez détaché
votre écran, vous vous demanderez
sûrement comment le récupérer. Si
vous tapez la commande suivante
dans le terminal, une liste d'écrans
s'affiche :

```
screen -ls
```

Ma liste ressemble à ceci :

```
There is a screen on:
```

```
17153.pts-0.lswest-netbook  
(Detached)
```

```
1 Socket in /tmp/screens/S-  
lswest.
```

Ou, si je tape la commande
dans la session :

```
There is a screen on:
```

```
17153.pts-0.lswest-netbook  
(Attached)
```

```
1 Socket in /tmp/screens/S-
```

lswest.

Après avoir vu cette liste, vous
êtes peut-être un peu perplexe. En
fait, c'est la liste des fichiers créés
par chaque session dans
/tmp/screens/S-<username>/. Elle
affiche aussi l'état de cet écran
(attaché, détaché, etc.). Pour se
reconnecter, ou « attacher » une
session d'écran, vous devez entrer
la commande :

```
screen -r <nom de l'écran>
```

Donc, avec l'exemple ci-
dessus, la commande serait :

```
screen -r 17153.pts-0.lswest-  
netbook
```

Bien sûr, nous sommes pares-
seux, et nous préférons donc
utiliser juste l'identificateur numé-
rique (17153, dans ce cas). L'identi-
fication devrait être suffisante pour
accéder à une session locale ;
cependant, je pense que le reste
sera obligatoire si vous êtes
connecté à une session distante
d'une manière ou d'une autre.

Encore une suggestion un peu
plus complexe, que l'on peut
trouver utile, c'est d'avoir le
numéro d'écran dans le prompt
Bash ou Zsh (car je suis un
inconditionnel du Z-shell). Vous
pouvez faire cela en ajoutant la
variable « \$WINDOW » à l'invite de

prompt, pour que soit affichée la
valeur de la fenêtre correspondante
(ex : si vous avez 3 fenêtres
ouvertes de screen et que vous
êtes dans un shell sur l'écran 1 (la
numérotation commence à 0, donc
le 1 sera le deuxième ouvert), la
valeur affichée sera 1). Mon invite
est configurée en utilisant le script
ci-dessous.

C'est une invite de commande
pour Z-shell, donc cela ne
marchera pas pour Bash, mais vous
donne une idée de comment faire.
En fait, le script vérifie si
\$WINDOWS retourne une valeur, et
si c'est le cas, l'affiche dans l'invite
de commande, sinon non. Les

```
if [ x$WINDOW != x ]; then  
    ##9484;##9472;[5:lswest@lswest-netbook:~]-[15:21:07]  
    ##9492;##9472;>  
    export  
    PS1="%{$fg[white]}##9484;##9472;[%{$fg[cyan]}]$WINDOW%{$fg[white]}:%{$fg[green]}%n%{$fg[cyan]}%@%{$fg[green]}%m%{$fg[white]}:%{$fg[yellow]}%~%{$fg[white]}] %{$fg[yellow]}%  
}-  
%{$fg[red]}[%{$fg[cyan]}%*%{$fg[red]}] %{$reset_color}%{$reset_color}"$'\n'"%{$fg[white]}##9492;##9472;>%{$reset_color}" "  
else  
    ##9484;##9472;[lswest@lswest-netbook:~]-[15:21:07]  
    ##9492;##9472;>  
    export  
    PS1="%{$fg[white]}##9484;##9472;[%{$fg[green]}%n%{$fg[cyan]}%@%{$fg[green]}%m%{$fg[white]}:%{$fg[yellow]}%~%{$fg[white]}] %{$fg[yellow]}%  
%{$fg[red]}[%{$fg[cyan]}%*%{$fg[red]}] %{$reset_color}%{$reset_color}"$'\n'"%{$fg[white]}##9492;##9472;>%{$reset_color}" "  
fi
```

sections en commentaire montrent l'apparence de mon invite de commande pour chaque option. Je trouve que ceci est un petit truc utile à faire quand on utilise Screen.

La dernière chose à décrire ce mois-ci est la création d'un fichier `.screenrc`, pour modifier le paramétrage par défaut de Screen. Mon fichier `.screenrc` ressemble à celui de l'encart en haut à droite (fortement inspiré du `.screenrc` de rson posté sur les forums ArchLinux).

Le commentaire au-dessus de la ligne « `hardstatus alwayslastline` » est un exemple de ce à quoi ressemble le résultat final. Toutes les autres commandes sont plutôt bien commentées. Le résultat montre l'invite de commande ci-dessous.

C'est un `.screenrc` simple, il me faudrait un article ou deux pour décrire à peine la moitié de ce que vous pouvez faire avec ces configurations. Je laisserai donc le `.screenrc` comme il est avec les commentaires. Vous devrez suivre les liens du paragraphe « Pour Approfondir » vers un site qui tente d'expliquer tous les paramétrages

possibles pour les fichiers `.screenrc`.

Enfin la vraiment toute dernière chose que je dois décrire dans cet article est comment quitter Screen. On peut le faire de deux façons :

1. Ctrl + a, \ quitte l'écran et tue toutes les fenêtres.

2. Fermez toutes les fenêtres sauf un shell, et taper juste

```
exit
```

Si vous avez d'autres questions, ou aimeriez demander un article sur un aspect de Screen, vous pouvez m'envoyer un e-mail à lswest34@gmail.com. Il en est de même pour quiconque a des idées d'articles de toutes sortes, ou des questions sur le l'interface en ligne de commande. Je vous souhaite à tous de bonnes vacances et une bonne année.

```
# Screenrc - Screen config file
# Author: Lswest
# Created: 24-11-2009 16:08:50
#
#
# General Settings

startup_message off          # Disable startup message
vbell on                    # Give visual alert instead of sound
defutf8 on                  # Always use utf8

# Hardstatus

backtick 10 1 300 "/usr/bin/updateCheck" # List number
of available updates

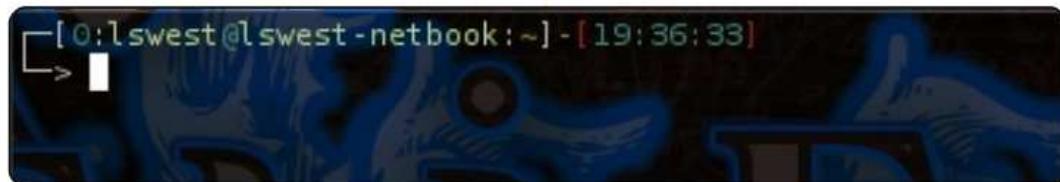
# 0 Zsh  1 IRC              --INSERT-- No Packages to
Update
hardstatus alwayslastline "%{= dd}%-w%{+u}%n %t%{-}%+w
%=%{= dW}%h%{-}%20`%10`"

# autostart screen sessions
screen -t Zsh 0 /bin/zsh
screen -t IRC 1 /usr/bin/irssi
#
```

Pour approfondir :

http://www.gnu.org/software/screen/manual/html_node/Default-Key-Bindings.html#Default-Key-Bindings - La page de manuel pour les raccourcis clavier sur la page d'accueil GNU.

http://www.math.utah.edu/docs/info/screen_9.html - Liste courte et concise de trucs pour `.screenrc`, et Screen en général.



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Quand il en trouve le temps, il publie également un blog à l'adresse: lswest34@gmail.com.



Le mois dernier, je vous ai présenté GNU Screen. Maintenant, nous sommes prêts à passer à des configurations légèrement plus utiles. Je ne parlerai que de quelques aspects de Screen que j'utilise et que je trouve utiles, mais ils sont loin d'être aussi avancés que d'autres, et ils ne sont pas les seuls à valoir le coup d'être utilisés dans Screen. Je serais heureux que des lecteurs m'envoient leur configuration et leurs réglages préférés pour Screen. Je peux les faire paraître en tête du Command & Conquer chaque mois, afin que l'on puisse apprendre d'autres choses grâce à eux. Voici les notions que je vais aborder dans cet article : démarrer automatiquement des fenêtres avec des commandes, utiliser Screen avec SSH pour démoniser des commandes, partager des sessions Screen, le découpage des fenêtres et les avantages de l'utilisation Screen au lieu des écrans de terminaux ou une fenêtre de commande. Alors commençons tout de suite, et faisons chauffer les sessions Screen et les fichiers .screenrc !

Si vous avez parcouru mes fi-

chiers de configuration le mois dernier, vous avez sans doute remarqué les deux lignes suivantes à la fin de mon fichier .screenrc :

```
# autostart screen sessions
screen -t Zsh 0 /bin/zsh
screen -t IRC 1
/usr/bin/irssi
```

Ces lignes ajoutent une nouvelle fenêtre à Screen (dont le numéro est situé après le titre) avec un titre (spécifié après le drapeau « -t ») et la commande qui suit. Ainsi, le fichier .screenrc lance Screen avec une fenêtre 0 exécutant mon z-shell et portant le titre « Zsh », et une fenêtre 1 où tourne irssi et qui porte le titre « IRC ». Screen reste dans la dernière fenêtre créée et ainsi mes sessions Screen démarrent toujours dans IRC, car c'est ce que je voudrais probablement utiliser. Je ne sais pas si le nombre de fenêtres Screen est limité, mais je recommande de ne pas dépasser le nombre de raccourcis clavier permettant de passer rapidement de l'une à l'autre (donc un maximum de 9 fenêtres). Ceci est particulièrement utile si vous êtes de ces personnes qui utilisent régulièrement des programmes en ligne de

commande (mutt, irssi, midnight commander, etc.) et qui veulent les avoir sous la main dans une fenêtre facilement accessible, sans avoir à ouvrir tout un tas de terminaux.

Si vous utilisez SSH, vous avez probablement rencontré le problème suivant : vous accédez à une machine, lancez une commande, et, enfin, perdez la connexion ; vous êtes alors frustré de devoir relancer la commande, car vous avez perdu sa sortie écran. Voici pourquoi moi, je pense que Screen (ou un autre multiplexeur de terminal) est indispensable à tout utilisateur de SSH qui restera connecté plus de quelques minutes et qui aura sans doute besoin de lancer plus d'une commande. Même si vous faites partie de ces utilisateurs qui ne lancent qu'une commande et ne restent connectés que quelques minutes, Screen pourra quand même vous rendre service. Si vous vous connectez en SSH et utilisez Screen, vous pouvez régler les commandes à exécuter, détacher la session Screen et vous déconnecter du serveur SSH. Si vous voulez vous reconnecter et récupérer l'information, vous avez

juste à rattacher la session Screen. Je vous recommande de détacher la session avant de vous déconnecter de SSH, mais Screen devrait automatiquement détacher la session et continuer à fonctionner lorsque la connexion se ferme. C'est très utile pour les administrateurs système qui risquent de devoir lancer un script pour mettre à jour notamment des autorisations, et qui n'auront pas à exécuter le processus en tâche de fond ou à garder la connexion SSH ouverte durant tout le processus. Si je me souviens bien, vous pouvez même configurer SSH pour qu'elle lance toujours Screen pour n'importe quelle connexion SSH, ce qui veut dire que vous serez toujours dans une fenêtre Screen.

Une autre fonctionnalité vraiment utile de Screen est la possibilité de partager des sessions. C'est très pratique si vous éditez un script et avez besoin qu'un autre utilisateur saisisse quelque chose. Vous pouvez autoriser quelqu'un d'autre à se connecter en SSH à la session, et partager l'écran. Pour cela, l'hôte (le premier utilisateur) doit faire la chose suivante :

Ctrl + a
:multiuser on

Ctrl +a est en fait la combinaison de touches à taper, puis vous devez saisir « :multiuser on ». Vous devez ensuite autoriser l'utilisateur distant à se connecter à la session Screen ainsi :

Ctrl +a
:acladd <ruser>

Remplacez « <ruser> » par le nom de l'utilisateur distant. Une fois que c'est fait, l'utilisateur distant peut se connecter à la session Screen en tapant :

screen -x \$USER/<screen ID/name>

Vous devez remplacer \$USER par le nom de l'utilisateur original (l'hôte), et « Screen ID/name » par le nom ou l'identifiant de la session Screen que l'utilisateur a initiée. Si vous voulez savoir comment régler un nom Screen, j'ajouterai quelques indications rapides à la fin de l'article sur la sécurisation de Screen et j'expliquerai comment ajouter des noms aux sessions Screen pour rendre leur gestion plus simple.

La toute dernière fonctionnalité

dont je vais vous parler pour le moment est la possibilité de découper la fenêtre de Screen. Cela signifie que vous pouvez avoir deux fenêtres tournant en parallèle et qui se positionnent comme dans un gestionnaire de fenêtre en cascade. Screen ne propose par défaut que le découpage horizontal ; il existe un patch pour pouvoir découper verticalement, mais vous devrez recompiler Screen à la main. Pour découper l'écran horizontalement dans Screen, utilisez la combinaison de touches suivante :

C-a S

Pour ceux qui n'ont pas lu mon article précédent, cela signifie Ctrl + a, puis S (Shift + s) . Cela prendra la fenêtre en cours d'utilisation et la coupera en deux au milieu. Une fois que vous aurez deux panneaux, vous pouvez passer de l'un à l'autre avec :

C-a <Tab>

où <Tab> est la touche tabulation du clavier. Si quelqu'un veut mettre en place le découpage vertical, j'enverrai un lien vers un tutoriel dans la section « Pour aller plus loin ».

La dernière chose que je veux expliquer dans cet article est la raison

pour laquelle on devrait utiliser Screen au lieu d'un écran de terminal ou d'une simple fenêtre de commandes. La réponse courte : par préférence personnelle. La réponse longue est que cela permet de lancer des sessions SSH en multi-tâche sans aucun risque de perdre les processus en se déconnectant et cela permet aussi de minimiser la quantité de RAM utilisée. En outre, cela vous permet de mieux surveiller tout ce qui tourne ; si vous donnez des noms aux fenêtres Screen, et que vous les lancez dans un seul émulateur de terminal, vous aurez une barre d'état listant les programmes en cours d'exécution. Je ne suis pas en train de dire qu'il vous faut l'utiliser absolument, mais pour ceux d'entre vous qui se sentent attirés par les fonctionnalités de Screen, je vous recommande vivement de l'utiliser et d'essayer de prendre l'habitude de l'utiliser régulièrement (ce que, je dois avouer, je n'ai pas encore réussi à faire).

Comme toujours, n'hésitez pas à m'envoyer toutes vos questions par courriel à lswest34@gmail.com. Vous pouvez aussi m'envoyer des suggestions pour de futurs articles à la même adresse.

Dernière commande pour

Screen :

screen -S <name of screen session>

Cette commande crée une session Screen en utilisant le nom fourni après le drapeau -S, pour un accès rapide et aisé (c'est super quand vous lancez beaucoup de sessions Screen).

Pour aller plus loin :

Découpage vertical : <http://scie.nti.s.t/2008/8/22/gnu-screen-with-vertical-split-support>.

Le manuel des combinaisons de touches sur la page d'accueil GNU : http://www.gnu.org/software/screen/manual/html_node/Default-Key-Bindings.html#Default-Key-Bindings.

Une courte liste de trucs utiles pour les fichiers .screenrc, et pour Screen en général : http://www.math.utah.edu/docs/info/screen_9.html.



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Quand il en trouve le temps, il publie également un blog à l'adresse : <http://lswest-ubuntu.blogspot.com>.



Après la sortie du numéro 33 le mois dernier, Chris Johnston, membre d'Ubuntu, m'a envoyé un courriel sympathique pour m'indiquer l'existence de Byobu (<https://edge.launchpad.net/byobu>). Je dois avouer que je n'ai pas eu autant de temps que j'aurais souhaité pour jouer beaucoup avec, mais j'ai pu cependant avoir une idée assez générale de ce qu'il propose. J'ai également beaucoup entendu parler de tmux sur les forums Arch Linux avant d'attaquer cette série d'articles et je vais donc en parler comme une alternative possible. Je n'entrerai pas trop dans les détails des fonctionnalités, car Byobu est similaire à Screen pour ce qui est des raccourcis clavier et des paramètres, tmux est extrêmement bien documenté, et mon fichier de configuration est commenté et devrait être suffisamment clair. Vous pouvez m'envoyer par courriel des demandes pour un article plus approfondi sur l'un ou l'autre, car je ne suis pas certain qu'il y ait une forte demande pour une description très détaillée de leurs fonctionnalités.

Byobu

« *Byobu est un terme japonais désignant des cloisons mobiles pliantes à plusieurs panneaux décoratifs. En tant que projet open source, Byobu est une amélioration élégante du très fonctionnel, simple et pratique GNU Screen. Byobu inclut un profil amélioré, et des utilitaires de configuration pour le gestionnaire de fenêtre GNU Screen, comme par exemple des notifications système avec des interrupteurs à bascule.* » (D'après la page Launchpad en anglais.)

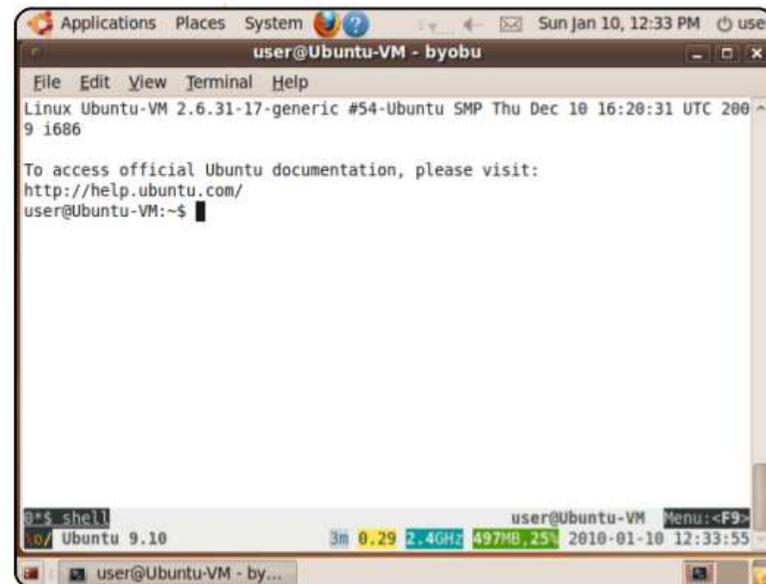
Cela signifie que Byobu n'est pas une réécriture de Screen, mais plutôt un programme qui accompagne Screen, lui ajoutant de nouvelles fonctionnalités. Les fonctionnalités principales que j'ai remarquées et qui sont différentes de Screen sont que, par défaut, Byobu fournit une barre d'état qui offre plus d'informations sur le système que ne le fait Screen par défaut, comme on peut le voir en haut à droite.

De plus, Byobu offre un menu basé sur ncurses pour créer des profils d'écrans. On peut y accéder avec la touche F9 (comme

on le voit sur la barre de tâches, « Menu : <F9> »). Je ne mets pas de copie d'écran du menu, car il est plutôt simple et explicite.

tmux

« *tmux est un multiplexeur de terminaux : il permet de créer, d'accéder, et de contrôler plusieurs terminaux (ou fenêtres), chacun exécutant un programme distinct, depuis un écran unique. tmux peut être détaché d'un écran et continuer à tourner en tâche de fond, puis rattaché plus tard. tmux utilise le modèle client-serveur. Le serveur gère des sessions multiples et chaque fenêtre est une entité indépendante qui peut être manipulée librement, notamment reliée à plusieurs sessions, déplacée entre les sessions, et j'en passe. Chaque session peut être attachée (affichage et saisie clavier) à des clients multiples. tmux est destiné à être une alternative moderne et*



sous licence BSD à des programmes tels que GNU Screen. » (D'après la page d'accueil : <http://tmux.sourceforge.net/>).

Ce que tmux propose, et qui est différent de GNU Screen, ce sont des commandes faciles à comprendre, un découpage horizontal et vertical, et des valeurs par défaut sensées (barre d'état, etc.) ; de plus, on peut le modifier de façon dynamique depuis la ligne de commandes, et il nécessite moins de mémoire que Screen. Il faut environ 2,4 Mo de mémoire pour la première session, mais chaque nouvelle fenêtre ne demande que 1 Mo de

mémoire. Cela n'est peut-être pas grand-chose, mais sur une vieille machine ça peut vraiment faire une différence. De plus, les raccourcis clavier par défaut commencent tous par Ctrl + b, au lieu de Ctrl + a, ce qu'on peut changer et que j'ai d'ailleurs changé dans mon fichier de configuration, que je posterai sous forme de lien à la fin de l'article. De par sa licence, tmux est inclus par défaut dans les systèmes BSD, pour ceux que ça intéresse. Enfin, quelques raccourcis clavier sont différents, mais la page de manuel de tmux est vraiment très claire, et propose une liste complète des commandes que l'on peut obtenir en tapant :

tmux list-commands

En haut à droite, vous verrez une copie d'écran de tmux (comme vous pourrez le deviner, j'utilise en fait tmux sur mon netbook, contrairement à Byobu qui tourne sur une machine virtuelle).

Fichier de configuration :

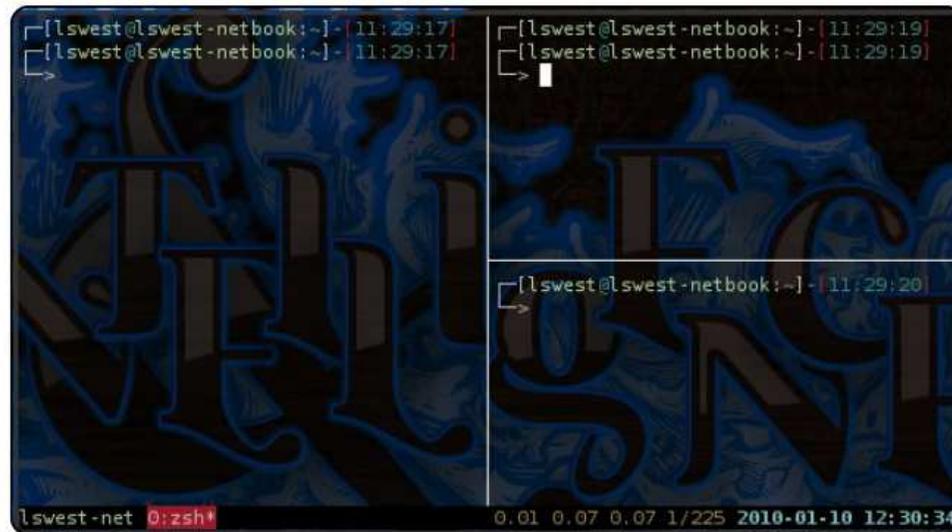
(~/tmux.conf)

<http://lswest.pastebin.com/fa64f955>

Liste de commandes pour tmux :

<http://lswest.pastebin.com/f7d0cad21>

Je me suis dit que je devrais inclure quelques choix supplémen-



taires dans cette série, puisque les principes de Linux sont le choix et la liberté d'utiliser ce dont vous avez envie. Personnellement, je trouve tmux un peu plus facile à utiliser et à comprendre, mais GNU Screen est le programme le plus connu, ce qui explique qu'on en a parlé plus en détail que tmux. Cependant, la page de manuel couvre toutes les informations habituelles, ainsi que les raccourcis clavier, les options de configuration, etc. Je recommande vivement de regarder la page de manuel avant de demander comment faire telle ou telle chose, car presque tout ce dont vous aurez besoin de savoir s'y trouve (et est plutôt facile à trouver). Ces deux programmes sont dans le dépôt Universe dans Ubuntu 9.10. Comme toujours, toute suggestion d'article ou

question est la bienvenue à l'adresse [lswest34@gmail](mailto:lswest34@gmail.com) et je ferai de mon mieux pour répondre aux questions et aux demandes.

Pour aller plus loin :

Byobu :

<https://edge.launchpad.net/byobu>

et :

<http://blog.dustinkirkland.com/search/label/Byobu>

tmux :

<http://www.openbsd.org/cgi-bin/man.cgi?query=tmux&sektion=1> (page de manuel en ligne)

et :

<http://tmux.sourceforge.net/> (page d'accueil).



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez écrire à Lucas à : lswest34@gmail.com.





Après avoir terminé les articles sur Screen, je me suis dit que mes lecteurs voulaient peut-être en savoir davantage sur ce que des shells comme bash ou zsh peuvent faire. Par conséquent, je vais maintenant traiter les divers shells qui existent sous Linux (avec une courte description) et faire une section plus approfondie sur la configuration et la personnalisation de Z-shell (Zsh) et de Bourne-Again Shell (Bash), puisque ce sont les deux shells les plus utilisés et ceux avec lesquels je me sens le plus à l'aise. Cela vous donnera peut-être envie, chers lecteurs, d'essayer quelques nouveaux shells par vous-mêmes pour découvrir ce qu'ils permettent de faire.

Les shells suivants sont disponibles :

Bourne Shell (sh) – le shell original sous Unix. Il n'offrait pas de fonctionnalités particulières autre que celles qu'on attend d'un shell.

Almquist Shell (ash) – une réécriture de Bourne shell, sous

licence BSD, et proposant les mêmes fonctionnalités.

Bourne-Again Shell (bash) – le shell standard utilisé dans les distributions Linux. Il offre un sur-ensemble des fonctionnalités de Bourne shell. Il fait partie du projet GNU.

Debian Almquist Shell (dash) – un remplaçant moderne de l'Almquist shell pour les distributions Linux basées sur Debian.

Korn Shell (ksh) – un shell écrit par David Korn.

Z Shell (zsh) – considéré comme le shell le plus complet disponible (c'est celui qui offre le plus de fonctionnalités). On peut le décrire comme un sur-ensemble de sh, ash, bash, csh, ksh, et tcsh (TENEX C-shell).

C Shell (csh) – un shell écrit par Bill Joy, spécial dans le sens où sa syntaxe ressemble à celle du langage de programmation C.

Cette liste est loin d'être exhaustive, mais ces shells sont

ceux, je pense, qui sont encore activement développés et utilisés par les membres de la communauté. Vous vous demandez peut-être pourquoi quelqu'un prendrait la peine de modifier son shell par défaut. Je préfère le Z-shell à Bash tout simplement parce qu'il offre certaines fonctionnalités que je préfère (une meilleure complétion automatique que Bash avec la touche TAB, une meilleure coloration syntaxique pour les invites de commandes, une invite de commande à gauche, mais aussi à droite, etc.). Comme pour tant d'autres choses sous Linux, ça se résume au final à une question de choix. Peut-être êtes-vous un programmeur talentueux en C, qui préférez avoir un shell avec une syntaxe similaire et avez donc opté pour le C-shell. Je ne dirai pas que l'un est meilleur qu'un autre, simplement parce qu'il apporte une ou deux fonctionnalités en plus, et vice-versa. Cependant, je ne parlerai que de la configuration de Z-shell et de Bourne-Again shell dans cet article, car j'ai une certaine expérience avec ces deux-là et que ce sont, paraît-il, les plus largement utilisés.

La première chose qui doit être traitée est comment installer et tester un nouveau shell et comment opter pour un autre shell par défaut. Pour installer le shell que vous voulez essayer, vous avez juste à utiliser apt-get. Une fois qu'il est installé, la page de manuel (man) vous indiquera l'emplacement du fichier de configuration. En outre, puisque vous voudrez sans doute en voir l'invite par défaut, vous pourrez basculer entre les shells tout simplement en lançant le nom binaire d'un autre shell (sh, ash, bash, zsh, csh, ksh, et ainsi de suite). Vous vous trouverez ainsi dans le nouveau shell sans modifier le shell par défaut. Je recommande toujours de regarder le fichier de configuration par défaut, et d'en faire une version locale pour la personnalisation, au cas où quelque chose irait de travers. Je recommande également de tester une nouvelle invite en ligne de commande, avant de la placer dans le fichier de configuration. C'est aussi simple que d'exporter la variable PS1 au moyen de la ligne de commande. Continuez à figoler jusqu'à obtenir satisfaction, puis copiez la version finale

dans le fichier de configuration. Une fois que la configuration vous convient et que vous êtes sûr que le fichier de configuration ne contient pas de bogues majeurs, vous êtes prêts à changer votre shell par défaut (si c'est ce que vous souhaitez). Pour ce faire, lancez la commande suivante :

```
sudo chsh -s  
/chemin/du/binaire $USER
```

Vous devez remplacer « /chemin/du/binaire » par le chemin de la commande du shell (par exemple : /bin/bash), et \$USER par votre nom d'utilisateur ou celui du compte pour lequel vous voulez changer le shell. Si vous ne savez pas quels shells sont disponibles (et reconnus par votre système), vous pouvez les afficher avec la commande :

```
chsh -l
```

Cela n'affichera peut-être pas tous les shells, car la commande n'affiche que ceux classés dans le fichier /etc/shells, mais la plupart des paquets devraient mettre ce fichier à jour en conséquence.

Vous vous demandez peut-être ce qui, exactement, peut être configuré dans un shell, et à quoi ça peut

vous servir. Je vais expliquer les quelques points suivants : comment exporter des variables d'environnement pour les utiliser dans les gestionnaires de fenêtres (openbox au lieu de Gnome par exemple), comment faire des alias de commandes pour une utilisation plus simple, comment personnaliser l'invite elle-même et comment ajouter des fonctions au shell.

Configurer Z Shell

Suivez ce lien pour voir un fichier .zshrc complet : <http://lswest.pastebin.com/WBm22Wig>. Une note au sujet des « bindkeys » : zsh ne reconnaît pas les touches début/fin/page précédente/page suivante et n'affiche que des séquences d'échappement quand on les utilise, sauf si vous définissez des « bindkeys » comme je l'ai fait. Vous aurez peut-être à chercher la bonne séquence d'échappement pour la touche. Si vous activez l'émulation vi (c'est ce que j'utilise, vous l'obtenez en exécutant « bindkeys -v »), vous pouvez voir la séquence d'échappement en appuyant sur Ctrl + v, puis sur la touche pour laquelle vous voulez afficher la séquence d'échappement. Normalement celles que j'utilise

devraient fonctionner chez la plupart d'entre vous, alors commencez par les essayer.

Je me référerai à ce fichier pour les exemples (en utilisant le numéro attribué aux lignes sur pastebin).

La première chose dont j'aimerais parler est l'exportation de variables, car c'est quelque chose de très utile, et plutôt facile à faire. Pour exporter une variable, utilisez simplement la syntaxe suivante :

```
export $VARIABLE="valeur"
```

Comme vous le voyez dans mon fichier de configuration aux lignes 11, 15, 17, 82, 116, 117, 131 et 132. Vous devez bien sûr remplacer \$VARIABLE par le vrai nom de la variable (par exemple DE ou OOO_FORCE_DESKTOP), et « valeur » par la vraie valeur. Vous pouvez mettre des apostrophes doubles autour de la valeur, mais c'est inutile s'il n'y a qu'un mot (comme vous pouvez le voir dans le fichier). Les deux derniers « export » dans mon fichier sont extrêmement utiles lorsque vous utilisez openbox, car ce dernier règle l'environnement de bureau sur Gnome pour xdg-open (le programme qui sélectionne automatiquement l'application par

défaut pour chaque type de fichier). Autrement dit, « xdg-open » suivi d'un chemin vers un fichier ouvrira nautilus quand il est réglé sur Gnome, thunar quand il est sur xfce, et konqueror quand il est sur KDE. L'exportation de OOO_FORCE_DESKTOP règle OpenOffice pour qu'il utilise le thème gtk au lieu du thème Qt, qui est l'option par défaut à moins que DE ne soit réglé sur Gnome.

Un autre truc utile est d'ajouter des alias à votre fichier de configuration, ce qui permet d'utiliser facilement des arguments étendus d'une commande. On peut voir cela très bien à la ligne 84, où j'utilise l'alias « trayer » (ignorant ainsi le véritable fichier binaire portant ce nom) pour lancer trayer avec un ensemble d'arguments spécifiques. Toutefois, si vous vous apercevez que vous souhaitez parfois utiliser le fichier binaire trayer sans l'alias, vous pouvez passer outre l'alias de façon temporaire en utilisant :

```
\<nom de l'alias>
```

Ainsi, dans ce cas, on écrirait \trayer. Cela ressemble à la façon dont on échappe certains caractères pour que le shell les voie en tant que texte. Un alias très utile que

j'utilise sur tous mes ordinateurs sous Linux est l'alias pour ls (ligne 64), car il me donne une liste bien plus détaillée des fichiers d'un répertoire.

Maintenant nous arrivons à la personnalisation la plus utilisée des shells : l'invite de commande elle-même. L'invite que je préfère utiliser dans le Z-shell est la suivante :

```
[lswest@laptop:~] - [14:24:29]  
➤
```

C'est une invite sur deux lignes, qui me laisse plus de place pour taper des commandes et me montre l'utilisateur actuellement connecté, le nom de la machine, et le nom du répertoire courant (après les deux points). Pour ceux qui se demandent comment j'arrive à faire une invite sur deux lignes, la magie se passe ici : "\$\n" où je coupe la section principale de l'invite, puis j'ajoute une séquence d'échappement pour passer à la ligne, pour ensuite continuer l'invite. Ceci ne fonctionnait pas (la dernière fois que j'ai essayé) en mettant juste la séquence d'échappement entre des apostrophes doubles. On peut également ajouter une invite du côté droit en utilisant la variable RPPROMPT (je l'ai commentée dans

mon fichier de configuration, mais elle y est).

Je mets à jour le fichier de configuration régulièrement et la copie qui se trouve sur pastebin en ce moment est une version ou deux en retard, mais le changement principal est que mon invite actuelle me montre également l'heure à laquelle j'ai lancé une commande. Si vous regardez le fichier de configuration, vous pourrez voir qu'il y a en fait deux invites dans une instruction « if ». Cela vérifie simplement si j'utilise Screen, et, si c'est le cas, cela affiche le numéro de la fenêtre Screen avant mon nom d'utilisateur, me permettant de savoir facilement où je me trouve. Une liste complète des séquences d'échappement pour Zsh se trouve sur la page de manuel de zshmisc, mais voici une liste de celles que j'utilise souvent (prise sur <http://www.acm.uiuc.edu/workshops/zsh/prompt/escapes.html>) :

Caractères littéraux :

%% - A %
) - A)

Répertoires :

%d - le répertoire courant (\$PWD)

%~ - \$PWD, mais avec deux types de substitutions. Si le répertoire nommé X est un préfixe du répertoire courant, alors ~X sera affiché. Si le répertoire courant est votre répertoire racine, \$HOME, seul ~ est affiché.

%c - partie finale de \$PWD. Si vous voulez n morceaux, ajoutez l'entier n après le %.

%C - pareil que %c and %, sauf que les ~ ne sont jamais affichés à la place des noms de répertoires.

Informations sur la machine :

%M - le nom complet de la machine.

%m - Le nom de machine jusqu'au premier point. On peut préciser combien de morceaux du nom on souhaite en faisant suivre le % avec un entier.

Informations sur l'heure actuelle :

%t - heure actuelle au format 12 heures (am/pm).

%T - heure actuelle au format 24 heures.

%* - heure actuelle au format 24 heures, avec les secondes.

Informations sur la date actuelle :

%w - la date au format jour (jj).

%W - la date au format mm/jj/aa.

%D - la date au format aa-mm-jj.

%D{chaîne} - chaîne est formatée en utilisant la fonction strftime. Regardez strftime(3) pour plus de détails. Trois codes supplémentaires sont disponibles : %f affiche le jour du mois, comme %e mais sans espace avant si le jour est sur un seul chiffre, et %K/%L correspondent à %k/%l pour l'heure du jour (horloge 24/12 heures) de la même façon.

Informations diverses :

%h - numéro de l'événement actuel dans l'historique.

%n - équivalent à \$USERNAME.

%l - le terminal (tty) sur lequel l'utilisateur est connecté.

%# - un « # » si le shell tourne avec

des privilèges, un « % » sinon. La définition des privilèges, dans ce cas, est que soit l'identifiant utilisateur est effectivement égal à 0, soit, si les capacités POSIX.1e sont supportées, qu'au moins une des capacités soit levée dans l'un des vecteurs de capacité « Effective » ou « Inheritable ».

Zsh propose quelques couleurs par défaut, accessibles par leur nom anglais, comme rouge (red), cyan, etc. Mais il accepte aussi le style de formatage usuel `\e[0;31m` (on en parlera dans la section sur Bash).

Pour finir, mais c'est peut-être la plus utile, nous avons la possibilité d'ajouter des fonctions au shell. Le paramétrage est exactement le même que pour les scripts Bash. La méthode consiste à définir une fonction avec « fonction nom() { #code } ». J'ai quelques fonctions dans mon fichier Zshrc, comme m4a, flvmp3, google, etc. Comme vous pouvez le voir, on peut aussi définir une fonction sans utiliser le descripteur « fonction », mais il rend les choses plus lisibles. Mon fichier de configuration n'est pas vraiment un bon exemple de fichier bien organisé. Idéalement, j'aurais dû regrouper tous les « export » ensemble, toutes les fonctions

ensemble, tous les alias et ainsi de suite. Au lieu de cela, j'ajoute les trucs au fichier au fur et à mesure que les idées me viennent, ce qui mène à ce bazar. J'arriverai sans doute à nettoyer tout ça un jour (cela m'arrive environ une fois par an).

Configurer le Bourne-Again Shell

Exporter et créer des alias se fait de la même façon en Bash qu'en Zsh, reportez-vous donc aux deux premières explications dans la section sur Zsh pour voir comment faire. Les seules parties dans le fichier .zshrc dont je parle qui ne concernent pas les invites en Bash sont les sections sur les « bind-keys » et sur PROMPT.

Pour personnaliser l'invite en Bash, cela ressemble à ce qu'on a vu en Zsh, sauf pour la liste des séquences d'échappement utilisables, ainsi que pour la façon dont la variable réagit quand il est question d'invite sur deux lignes. Voici une liste des séquences d'échappement pour bash (prise sur <http://www.cyberciti.biz/tips/howto-linux-unix-bash-shell-setup-prompt.html>) :

`\a` : le caractère ASCII bip (07)

`\d` : la date au format "JourDeSemaine Mois Jour" (par exemple "Mar Mai 26")

`\D{format}` : le format est envoyé à `strftime(3)` et le résultat est inséré dans la chaîne d'invite ; un format vide donnera une représentation du temps locale. Les accolades sont obligatoires.

`\e` : un caractère d'échappement ASCII (033)

`\h` : le nom de machine jusqu'au premier point

`\H` : le nom de machine

`\j` : le nombre de tâches actuellement gérées par le shell

`\l` : le nom de base du terminal correspondant au shell

`\n` : nouvelle ligne

`\r` : retour chariot

`\s` : le nom du shell, le nom de base de \$0 (la partie qui suit le slash final)

`\t` : l'heure actuelle au format HH:MM:SS sur 24 heures

`\T` : l'heure actuelle au format HH:MM:SS sur 12 heures

`\@` : l'heure actuelle au format am/pm sur 12 heures

`\A` : l'heure actuelle au format HH:MM sur 24 heures

`\u` : le nom d'utilisateur de l'utilisateur actuel

`\v` : la version de bash (par exemple 2.00)

`\V` : le numéro de sortie de bash, version + niveau de patch (par exemple 2.00.0)

`\w` : le répertoire actuel, avec \$HOME abrégé en tilde

`\W` : le nom de base du répertoire actuel, avec \$HOME abrégé en tilde

`\!` : le numéro de cette commande dans l'historique

`\#` : le numéro de cette commande

`\$` : si l'uid vaut 0, un #, sinon un \$

`\nnn` : le caractère dont le code octal

est nnn

/: un backslash

[: commence une suite de caractères non imprimables, qui peuvent être utilisés pour intégrer une séquence de contrôle du terminal dans l'invite

] : termine une suite de caractères non imprimables

Pour faire une invite sur plusieurs lignes en Bash, vous devez simplement placer la séquence d'échappement « \n » à l'endroit où vous voudrez passer à la ligne. Vous pouvez aussi personnaliser PS2 et ainsi de suite, qui apparaissent quand vous utilisez une commande sur plusieurs lignes (par exemple une boucle « for »). Quant aux couleurs, les séquences d'échappement sont disponibles (par exemple ici : http://wiki.archlinux.org/index.php/Color_Bash_Prompt#List_of_colors_for_prompt_and_Bash).

Bien sûr, vous pouvez placer les couleurs à l'intérieur des variables et utiliser le résultat dans le fichier de configuration. La version bash de mon invite Zsh (sans la date et l'heure) ressemblerait à ça :

export

```
PS1="\[\e[0;37m\] ──[\[\e[0;32m\]\u[\e[0;36m\]@\[\e[0;32m\]\h[\e[0;37m\]:\[\e[0;33m\]\w[\e[0;37m\]]\[\e[0;36m\]\n[\e[0;37m\] L->[\e[0m\] "
```

Je suis désolé de ne pas avoir un exemple de fichier bash à vous montrer, mais les syntaxes de configuration pour Zsh et pour Bash sont similaires et l'exemple Zsh devrait donc suffire pour les deux. S'il y a des volontaires, je serais heureux de montrer leur fichier .bashrc personnalisé, avec une représentation textuelle de l'invite, au début de chaque rubrique Command & Conquer. Si vous êtes intéressé, envoyez-moi un courriel à lswest34@gmail.com avec votre fichier .bashrc et une représentation textuelle de l'invite ou une copie d'écran. Faites référence à Command & Conquer dans l'objet du courriel pour que je le fasse monter dans ma liste de priorités. Pour ceux d'entre vous qui utilisent urxvt ou définissent des couleurs personnalisées dans le .Xdefaults, merci de bien vouloir me faire parvenir la section pertinente aussi (si vous envoyez une copie d'écran).

Vous pouvez m'envoyer toutes vos questions et suggestions ou vos problèmes à lswest34@gmail.com, et des idées d'articles sont toujours

les bienvenues dans ma boîte de réception ! Je vous souhaite à tous de bien vous amuser en configurant vos invites de commandes, et je serais curieux d'en voir les résultats ! J'espère que mes explications sont compréhensibles et je continuerai avec joie à parler d'autres personnalisations du terminal, s'il y a assez de personnes intéressées. Comme toujours, il y a encore plein de renseignements sur ce sujet dans la section « Pour aller plus loin ».

Pour aller plus loin

http://en.wikipedia.org/wiki/Alias_%28command%29 : informations sur la commande alias.

<http://www.cyberciti.biz/tips/howto-linux-unix-bash-shell-setup-prompt.html> : comment personnaliser l'invite de commandes sous Bash.
<http://markelikalderon.com/2007/11/24/full-paths-and-the-multiline-shell-prompt/> : les invites sur plusieurs lignes.

http://wiki.archlinux.org/index.php/Color_Bash_Prompt#List_of_colors_for_prompt_and_Bash : coloriser les invites Bash.

<http://docs.cs.byu.edu/linux/advanced/zsh.html> : comment configurer les invites Zsh.



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez contacter Lucas par courriel à : lswest34@gmail.com.



À la suite de l'article du mois dernier au sujet de la personnalisation des shells et de leurs invites, j'ai pensé qu'il serait sympa d'expliquer comment personnaliser les couleurs de votre terminal. C'est aussi une bonne introduction à votre fichier `.Xdefaults`, qui offre pas mal de contrôle sur les réglages utilisateur. Il permet de choisir le curseur de la souris, faire des réglages spécifiques à `urxvt`, configurer le terminal, les DPI, l'anti-aliasing et autres préférences de polices X, et choisir un thème pour « `xscreensaver` » (l'écran de veille), entre autres. Il y a plein d'exemples de palettes de couleurs pour terminaux sur le site web d'Aaron Griffin (le développeur en chef d'Archlinux) : <http://phraktur.net/terminal-colors/>. Aujourd'hui nous verrons les étapes de base permettant de définir une palette de couleurs personnalisée pour votre terminal :

- Comprendre la syntaxe utilisée par le fichier `.Xdefaults`

pour les couleurs.

- Trouver les valeurs hexa des couleurs et les couleurs complémentaires.
- Un moyen de tester l'affichage de la palette couleur obtenue.

Je me concentrerai sur les méthodes que je connais bien, mais ce sont loin d'être les seuls moyens de créer des palettes de couleurs. La première chose à faire est d'afficher la palette courante pour voir ce qu'il faut modifier, le cas échéant. Pour ce faire, je recommande chaudement le script `colour-scheme.sh` de Daniel Crisman (cliquer sur le premier lien de la section « Pour aller plus loin » et prendre le tout dernier script de la page). Pour l'utiliser, copiez-le dans un fichier et faites un `chmod +x` dessus. Par exemple :

vim couleurs

(voir note [1] page suivante pour plus d'information)

<Coller le script, sortir de vim et ensuite>

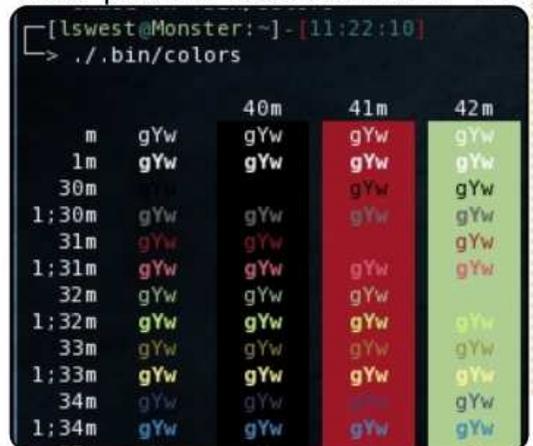
```
sudo chmod +x couleurs
```

Ensuite, pour le lancer, faites simplement

```
./couleurs
```

Cela affichera quelque chose comme la figure ci-dessous.

L'étape suivante consiste pour moi à choisir une couleur de base qui aura un impact sur les autres couleurs, car nous voudrions y ajouter des couleurs complémentaires. Vous avez toujours la possibilité de sélectionner une couleur primaire, le vert, par exemple. Vous pouvez ensuite ouvrir une



palette en utilisant soit `gcolor2` si vous désirez un sélectionneur de couleurs autonome, soit GIMP pour créer de nouvelles couleurs. L'important est de noter la valeur hexa de la couleur choisie. Une fois la couleur de base choisie, il est temps de trouver des couleurs complémentaires. Si, comme moi, vous ne savez pas retrouver de tête les couleurs complémentaires, vous pouvez utiliser la fonction de recherche disponible sur : <http://www.colourlovers.com/>. En lui donnant la valeur hexa, elle trouvera des palettes de couleurs correspondantes. Une fois les 16 couleurs de votre palette (et celles de premier/arrière-plan pour un total de 18 valeurs hexa) choisies, il est temps de les mettre dans votre fichier `.Xdefaults`. Le format pour les définir pour tous les terminaux est

```
!---- Couleurs des terminaux
*background: #000000
*foreground: #ffffff
*color0: #000000
*color1: #9e1828
*color2: #aece92
*color3: #968a38
```

```
*color4:      #414171
*color5:      #963c59
*color6:      #418179
*color7:      #bebebe
*color8:      #666666
*color9:      #cf6171
*color10:     #c5f779
*color11:     #fff796
*color12:     #4186be
*color13:     #cf9ebe
*color14:     #71bebe
*color15:     #ffffff
```

La première ligne donne le format des commentaires du fichier `.Xdefaults`. Vous pouvez aussi définir une palette pour un terminal spécifique en mettant son nom d'exécutable avant l'astérisque. Par exemple (même palette pour `urxvt` seulement) :

```
urxvt*background: [70]#000000
urxvt*foreground:  #ffffff
urxvt*color0:      #000000
urxvt*color1:      #9e1828
urxvt*color2:      #aece92
urxvt*color3:      #968a38
urxvt*color4:      #414171
urxvt*color5:      #963c59
urxvt*color6:      #418179
urxvt*color7:      #bebebe
urxvt*color8:      #666666
urxvt*color9:      #cf6171
urxvt*color10:     #c5f779
urxvt*color11:     #fff796
urxvt*color12:     #4186be
urxvt*color13:     #cf9ebe
urxvt*color14:     #71bebe
urxvt*color15:     #ffffff
```

Dans cette version, la valeur

entre crochets devant la valeur hexa de « `background` » correspond au réglage d'opacité (70% opaque, soit transparent à 30%). Avoir une vraie transparence n'est possible que sur les terminaux qui la prennent en charge et sur les systèmes faisant tourner un gestionnaire de composition (`xcompmgr`, `cairo-compmgr`, `compiz`, `mutter`, etc.).

Après avoir ajouté vos couleurs préférées à `.Xdefaults`, vous voudrez sans doute voir le résultat sans avoir à vous déconnecter et reconnecter. Heureusement, c'est possible avec un peu de magie de ligne de commande. La commande

```
xrdb -merge ~/.Xdefaults
```

forcera `xrdb` (X Resource Database Manager) à recharger les réglages de `.Xdefaults`, effaçant ainsi les précédents. En relançant le script « couleurs » vous aurez également un aperçu de votre nouveau jeu de couleurs.

C'est à peu près tout ce qu'il y a à faire. Il vous faudra sans doute procéder par tâtonne-

ments pour obtenir un résultat qui vous plaise vraiment, mais ça fait partie du jeu. Les deux liens que je propose pour plus d'information sur `.Xdefaults` sont pour Arch, mais je n'ai rien trouvé de correspondant pour Ubuntu. De plus, les instructions seront quasiment identiques pour les deux systèmes. Les exemples utilisés ci-dessus correspondent à mes palettes et je crois qu'ils étaient basés sur un `.Xdefaults` que j'ai trouvé en ligne il y a des années. Il ne reste sans doute plus grand chose de l'original, mais je voulais signaler que tout n'était pas de moi. Comme toujours, vous pouvez envoyer vos questions, suggestions et commentaires à : lswest34@gmail.com. Pour être sûr que je les lise, mettez dans l'en-tête de vos courriels « FCM - C&C ». J'aimerais aussi voir les résultats de vos `.Xdefaults`. Je serais heureux d'en montrer quelques-uns dans le prochain C&C, si vous m'envoyez une capture d'écran avec les réglages `.Xdefaults` correspondants.

Pour aller plus loin :

Le `colourscheme.sh` de Daniel Crisman sur : <http://tldp.org/>

HOWTO/Bash-Prompt-HOWTO/x329.html

La page Wiki d'Arch sur `.Xdefaults`, avec quelques liens et exemples : <http://wiki.archlinux.org/index.php/Xdefaults>

Un fil de discussion sur les forums Arch avec des palettes de couleurs pour terminaux : <http://bbs.archlinux.org/viewtopic.php?id=51818&p=1>

ColourLovers (pour les palettes de couleurs) : <http://www.colourlovers.com/>

[1] Pour coller dans Vim sans ajouter des espaces dus à l'auto-indentation, faire un « `:set paste` », entrer en mode coller en tapant « `i` », puis coller votre script dans le fichier. Pour sortir du mode coller, entrer « `:set nopaste` ». Ces deux commandes se lancent à la manière Vim habituelle, sans les guillemets.



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à : lswest34@gmail.com



Si vous avez un portable vous avez probablement contrôlé vos courriels, lu des actualités ou fait quelque chose en ligne lors d'un voyage. Si vous êtes comme moi, vous avez frissonné intérieurement à chaque fois que vous l'avez fait, en sachant bien qu'il y avait peut-être quelqu'un connecté sur l'accès gratuit/public en train de faire tourner un renifleur de paquets, en espérant trouver quelques mots de passe ou données bancaires. J'ai établi une règle stricte « pas de transaction bancaire sur la route » pour moi et ma famille, mais je m'inquiète aussi des autres mots de passe et des données privées. C'est là que la redirection de port SSH peut être très utile. Elle utilise SSL pour encrypter toutes les données envoyées ; elle n'utilise le wifi public que comme lien vers votre serveur SSH, quel qu'il soit (serveur maison, PC domestique, serveur du bureau, serveur virtuel privé, etc.) ; enfin, elle protège le trafic, qu'il soit montant ou descendant, de la plupart des renifleurs de paquets et autres attaques d'interception.

La première chose à faire est d'installer un serveur SSH sur votre PC ou de vous inscrire pour avoir un compte shell sur un serveur privé virtuel quelconque, si vous ne l'avez déjà fait. Si vous avez déjà un compte SSH, passez directement à l'étape 7.

Étape 1 :

Installez le serveur OpenSSH sur votre système Ubuntu :

```
sudo apt-get install openssh-server openssh-client
```

Étape 2 :

Créez une sauvegarde de `/etc/ssh/sshd_config`

```
cp /etc/ssh/sshd_config -
```

Étape 3 :

Modifiez le fichier `sshd_config`. Vous pourrez voir toutes les options possibles sur la page `man` :

```
man sshd_config
```

Le fichier de configuration de base devrait simplement interdire

une connexion en root et spécifier les utilisateurs qui peuvent se connecter par SSH. Pour ce faire, ouvrez `/etc/ssh/sshd_config` :

```
sudo vim /etc/ssh/sshd_config
```

Une fois ouvert, changez la ligne « `PermitRootLogin yes` » en « `PermitRootLogin no` » et ajoutez la ligne « `AllowUsers utilisateur1, utilisateur2, utilisateur3` » quelque part dans le fichier. Il faut bien sûr remplacer « `utilisateur1` » par le vrai nom d'utilisateur, tandis qu'`utilisateur2` et `utilisateur3` doivent être remplacés par tout autre compte autorisé à se connecter au serveur. Par exemple, comme je n'ai qu'un compte et un utilisateur autorisé, ma ligne serait :

```
AllowUsers lswest
```

Étape 4 :

Redémarrez le serveur SSH une fois vos changements effectués sur le fichier de configuration :

```
sudo /etc/init.d/ssh restart
```

Étape 5 :

Générez les clés SSH (si vous le désirez). Cette étape étant facultative, je ne décrirai pas les commandes exactes. Si vous désirez générer des clés, suivez les instructions du wiki (voir le lien dans la section « Pour aller plus loin »).

Étape 6 :

Configurez votre serveur/PC pour autoriser les accès internet et configurer `dyndns`. N'ayant jamais configuré des DNS dynamiques pour aucun ordinateur, veuillez consulter les instructions de l'article wiki (deuxième lien de la section « Pour aller plus loin »). Une solution temporaire serait de créer une tâche cron en effectuant la commande suivante :

```
curl http://www.whatis-myip.org
```

et d'en rediriger la sortie dans votre Dropbox ou dossier Ubuntu One, afin de pouvoir la consulter depuis d'autres ordinateurs. C'est-à-dire :

```
curl http://www.whatis-myip.org > ~/Dropbox/IP.txt
```

Si vous avez besoin d'approfondir les tâches cron, je les ai expliquées dans le numéro 24. Sinon, les étapes suivantes vont créer une tâche cron qui effectuera la commande ci-dessus tous les jours à 12h00 (midi) :

```
crontab -e $USER
```

Ajoutez la ligne suivante dans le fichier :

```
00 12 * * * curl
http://www.whatismyip.org >
~/Dropbox/IP.txt
```

Une fois cette étape achevée, vous êtes prêt à commencer.

Étape 7 :

Vous aurez besoin des informations suivantes :

L'adresse IP de votre serveur, le nom d'utilisateur et le mot de passe ou le nom d'utilisateur et un fichier clé (voir étape 5).

Pour vous connecter et transférer tout le trafic à travers le port 8080 sur votre connexion SSH, saisissez la commande suivante :

```
ssh -D 8080 lswest@localhost
```

Il vous sera ensuite demandé d'accepter l'identifiant rsa de votre serveur, répondez « oui ». Puis votre mot de passe vous sera demandé (sauf si vous avez généré un fichier clé). Une fois le mot de passe entré, vous serez accueilli par l'invite normale de SSH. Il vous faudra laisser la connexion/fenêtre ouverte (à moins que vous n'utilisiez screen ou tmux - dans ce cas il vous suffira de détacher la session).

Étape 8 :

Configurez le proxy SOCKS dans Firefox. Cela se fait simplement en allant à Édition / Préférences / onglet Avancé / sous-onglet Réseau / Paramètres de connexion. Une fois la fenêtre ouverte, utilisez les réglages montrés dans l'image en haut à droite.

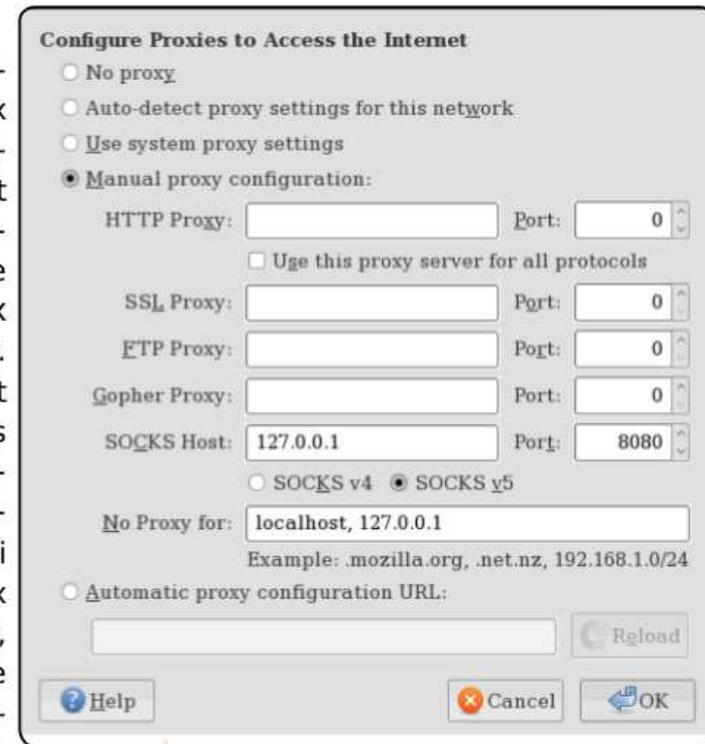
Étape 9 :

Déconnexion du proxy SOCKS. Changez simplement la configuration à « Utiliser les paramètres proxy du système » ou à « Pas de proxy », et déconnectez-vous du serveur SSH.

J'espère que cet article sera utile aux personnes très attentives à la sécurité et qu'il permettra d'éveiller aux problèmes de sécurité quotidiens ceux qui n'y pensent pas. On peut également utiliser le proxy dans d'autres domaines notamment avec Evolution/Thunderbird, ainsi que dans de nombreux autres programmes, mais il m'a semblé que celui-ci serait le scénario le plus universel/utile. Comme toujours, je serai ravi de répondre à des questions précises ou de répondre à des demandes d'articles par courriel. Envoyez toutes réactions, questions et autres à lswest34@gmail.com avec les mots "Command & Conquer" (ou simplement C&C) dans la ligne de sujet.

Pour aller plus loin:

<https://help.ubuntu.com/9.10/serverguide/C/openssh-server.html> - Article wiki sur l'installation d'OpenSSH.



<https://help.ubuntu.com/community/DynamicDNS> - Article wiki sur l'installation et la configuration des dns dynamiques.



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à : lswest34@gmail.com.



Avant de commencer l'article même, je voudrais vous faire part d'un courriel que j'ai reçu d'un lecteur. Alexandre a été assez aimable pour me faire remarquer qu'il y a un programme doté d'une interface graphique appelé « gstm » qui réalise la redirection de port pour le SSH. Pour les lecteurs qui préfèrent autant que possible utiliser une interface graphique, en voilà une. Ce programme est disponible dans les dépôts universe. D'autre part, un lecteur (qui, hélas, ne m'a pas communiqué son nom - mais qui se reconnaîtra) m'a signalé que, dans l'étape 7 de mon C&C du FCM n° 37, j'ai oublié de faire remarquer que vous devez remplacer « localhost » par l'IP de votre serveur. (La commande était « `ssh -D 8080 lswest@localhost` », où `lswest@localhost` doit être remplacé par `votre_nom@ADRESSE_IP_DU_SERVEUR`). Désolé pour la confusion que cela a pu engendrer.

Ce mois-ci, j'ai passé un certain temps à réécrire quelques correctifs pour DWM (Dynamic Window Manager) de sorte qu'ils puissent

fonctionner avec le correctif pango qui ajoute le support des polices xft à la barre d'état de DWM. Ce faisant, j'ai appris pas mal de choses à propos de diff et j'ai décidé de partager ce que j'ai appris avec vous, chers lecteurs. Si vous vous demandez : « Pourquoi devrais-je apprendre à utiliser diff, puisque je n'utilise pas DWM ni ne génère de correctif ? », la réponse est simplement : parce que diff peut être utilisé dans de nombreuses situations. Imaginez que vous êtes en train d'écrire un script et que vous voulez faire un ajout, mais avez besoin de l'ancienne version pour un autre ordinateur - plutôt que de créer et sauvegarder deux scripts séparés, vous pouvez écrire le script, en créer une copie, modifier la copie, générer un fichier .diff, et sauvegarder le script original et le fichier .diff, ce qui vous épargnera du travail supplémentaire à l'avenir. Autre exemple, vous voulez aider un ami et vous ne pouvez simplement lui envoyer le fichier que vous avez besoin de corriger, vous pouvez lui envoyer un .diff pour opérer les changements nécessaires. Il y a très certainement bien

d'autres usages (ajuster des fichiers de configuration, etc.) auxquels je n'ai pas encore pensé.

Diff est installé par défaut dans la plupart des distributions. Si vous ne l'avez pas sous Ubuntu, installez-le simplement avec :

```
sudo apt-get install diff
```

Une fois que c'est installé, vous êtes prêt à commencer. Pour générer un fichier .diff, vous avez besoin de deux fichiers que vous voulez analyser. Le premier sera « l'original » (je l'appellerai ainsi à partir de maintenant), et l'autre sera le fichier « mis à jour ».

Pour plus de simplicité, disons que j'ai un fichier qui contient ceci :

Tutoriel VirtualBox (mise en place, installation et configuration de la machine virtuelle).

Série sur la virtualisation : écrire chaque mois un article pour une distribution avec des captures d'écran, etc.

C&C : *parle de différentes choses qu'on peut faire avec curl, wget, etc ? et diff ?*

Et je veux le changer en :

Tutoriel VirtualBox (mise en place, installation et configuration de la machine virtuelle).

Série sur la virtualisation : écrire chaque mois un article pour une distribution avec des captures d'écran, etc.

C&C : *Diff (wget et curl le mois prochain)*

Je ferais les modifications désirées dans le fichier « mis à jour ». J'ajoute généralement le suffixe « -corrigé » à la fin du nom du fichier. Une fois les changements faits et que je veux générer un diff, je tape ce qui suit dans un terminal :

```
diff -up « original » « mis à jour » > articles\liste\correctif.diff
```

Remplacez « original » et « mis à jour » par le vrai nom des

fichiers et leur chemin. Si vous ne voulez pas que le fichier diff se crée dans le répertoire courant, ajoutez un chemin au nom du fichier de l'autre côté du chevron (>). Le chevron dit au shell de rediriger les sorties vers ce que vous passez après lui - en l'occurrence, le fichier .diff. Si vous voulez appliquer les changements à une autre copie du fichier original (par exemple sur un autre ordinateur), vous devrez exécuter une des deux commandes dans le dossier contenant le fichier que vous voulez corriger (elles font la même chose) :

```
patch -p1 < /chemin/vers/le/.diff
patch -Np1 -i /chemin/vers/le/.diff
```

où, bien évidemment, vous changez le « /chemin/vers/le/.diff » par le chemin réel.

Je me rends compte que mon exemple ne parle pas vraiment d'un cas où vous décideriez d'utiliser un fichier patch/.diff pour opérer les changements, mais je l'ai choisi pour sa simplicité. Un autre scénario où diff est utile : si vous avez deux fichiers (dans mon cas, c'est souvent des fichiers de configuration) et que vous ne savez pas s'ils ont été modifiés et, si

tel est le cas, quels sont ces changements. Pour le savoir, vous pouvez simplement exécuter la commande :

```
diff /chemin/vers/le/premier/fichier
/chemin/vers/le/second/fichier
```

N'oubliez pas de remplacer les chemins. La sortie devrait ressembler à ceci :

```
5c5
< - parle de différentes choses que
l'on peut faire avec curl, wget, etc. ?
et diff ?
---
> - Diff (wget et curl le mois prochain)
```

Je vais vous expliquer ces lignes. Le « 5c5 » est (je pense) une comparaison du nombre de lignes entre le premier et le second fichier. Je n'en suis pas certain, mais ça semble être le cas. La ligne suivante montre une ligne qui a été enlevée (le chevron ouvrant signifie la suppression) et la ligne en dessous montre la ligne qui a été ajoutée (qui remplace par conséquent la ligne d'origine), ce qui se traduit par le chevron fermant.

Comme vous pouvez le constater, c'est un outil très utile pour se rendre compte des différences

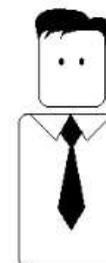
entre deux fichiers, particulièrement s'ils sont plutôt longs. Vous pouvez diriger la sortie avec un pipe vers les commandes « more » et « less », pour une lecture plus simple, ou la rediriger vers un fichier texte. Le format sera le même si vous n'ajoutez pas d'argument à la commande diff.

Avec un peu de chance, cette introduction à diff vous aura aidé à visualiser un scénario dans lequel vous pourriez en avoir l'usage et il rendra sans doute la vie plus facile à tous ceux qui décideront de s'en servir. Comme toujours, si vous avez la moindre question ou le moindre commentaire, vous pouvez m'envoyer un courriel à lswest34@gmail.com. Assurez-vous que vous avez ajouté « C&C » ou « FCM » dans l'objet du message, de sorte que je réponde rapidement (et puisse organiser mes courriels plus facilement !).



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à : lswest34@gmail.com.

Nous lançons une campagne contre les réseaux sociaux.



Soyez le premier de vos amis à nous rejoindre !





COMMAND & CONQUER

Écrit par Lucas Westermann

Avant de commencer l'article de ce mois, j'ai quelques corrections à apporter à mon dernier article. Le lecteur *Grofaty* a remarqué que le patch n'est pas installé par défaut dans Ubuntu, et il m'a aussi fait connaître *vimdiff* (interface de style vim pour diff).

Ce mois-ci, j'ai pensé qu'il serait amusant de parler de deux outils en ligne de commande pour télécharger des sites ou pages web, à savoir **cURL** et **Wget**. Vous pensez sans doute, « mais j'ai Firefox, quel est donc l'intérêt d'avoir cURL ou Wget ? ». Je les utilise principalement actuellement pour faire des copies de sites web (pas seulement une page web), pour les avoir même en étant déconnecté, ou pour télécharger une page web ou un fichier en étant derrière un pare-feu qui bloque ce site-là. Ces outils ont bien d'autres fonctions, comme archiver votre propre site web, faire de l'analyse grammaticale d'un site web à

l'intérieur d'un script, télécharger rapidement quelque chose sans avoir à ouvrir Firefox, ou télécharger tous les fichiers d'un certain type (utile aux étudiants ayant un portail web avec beaucoup de PDF de recherche). Pour ceux qui se demandent quelle est la différence entre cURL et Wget, elle est subtile, mais importante. cURL rapatrie le code HTML et l'envoie à STDOUT (c'est-à-dire qu'elle retourne le résultat de la commande), alors que Wget télécharge les fichiers HTML. Cela signifie que cURL est parfait pour faire l'analyse grammaticale de certains flux (si vous écrivez un script de recherche Google, par exemple), alors que Wget est utile pour faire une archive complète d'un site web.

Voici quelques exemples pour cURL :

```
curl -L www.w3schools.com/css
```

Cette commande dit à cURL de suivre les redirections sur la

page CSS de w3schools.com (précisément, Location: pointers). Sur ce site, il devrait suivre les liens "Next Chapter" automatiquement.

```
curl -u name:password https://mail.google.com/gmail/feed/atom
```

Cette commande donne à cURL un nom d'utilisateur et un mot de passe pour pouvoir s'authentifier sur le site web (dans ce cas, un flux atom de Gmail), et ainsi avoir accès au site

sans que vous ayez à ouvrir Firefox.

Ces exemples peuvent être utilisés dans un script qui accède à Google, cherche quelque chose, et retourne les résultats et/ou le HTML du premier site trouvé. Il peut aussi vous faire entrer dans votre compte mail de Google.

Et des exemples Wget :

```
wget -r -l3 http://w3schools.com/css/
```

```
↳ curl -L http://192.168.2.103/test
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <title>Test Page</title>
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
<div id="Content">
<div id="Header">
Test Page
</div>
<div>
<ul id="list-nav">
<li><a href="#">Home</a></li>
<li><a href="#">About Us</a></li>
<li><a href="#">Services</a></li>
```

COMMAND & CONQUER

Cette commande dirige Wget vers w3schools.com, et suit les liens récursivement sur 3 niveaux (c.à.d. 3 Homepage -> CSS Intro -> CSS Syntax). À noter que faire un parcours récursif dans Wget peut induire une lourde charge sur le serveur web, il doit donc toujours être utilisé avec l'argument sur les niveaux pour minimiser le trafic sur le site.

```
wget -c -U Mozilla www.website.com
```

Cet exemple ne comporte pas de vrai lien, car je n'ai pas eu d'idée de site adéquat assez

rapidement. Qu'importe, cette commande Wget se fait passer pour un navigateur Mozilla (en modifiant l'agent utilisateur), pour contourner les restrictions mises sur les gestionnaires de téléchargement. L'option -c demande à Wget de stocker les fichiers partiellement téléchargés, pour pouvoir reprendre le téléchargement plus tard.

```
wget -r -ll -A.pdf --no-parent http://url-to-website-with-pdfs/
```

Cette commande demande à Wget de suivre récursivement les liens d'un site web sur

un niveau et de télécharger tous les fichiers PDF trouvés. L'option -no-parent permet de ne jamais suivre un lien vers le répertoire parent (par exemple, www.test.com quand on est sur www.test.com/QuelqueChose), ce qui est aussi très utile pour éviter la surcharge du serveur. L'option -A accepte une liste séparée par des virgules d'extensions de fichier, ou des jokers/expressions régulières. Pour rejeter les fichiers d'un certain type, utiliser -R à la place de -A.

J'espère que cet article (court, je l'avoue) vous aura fait comprendre la puissance de Wget et cURL ; comme toujours, plein d'informations peuvent être trouvées sur leurs pages de manuel respectives. Si des gens veulent demander des articles sur des outils en ligne de commande, ils peuvent m'envoyer un mail à lswest34@gmail.com avec pour objet « FCM C&C » ou « Command & Conquer ». Si je ne connais pas encore l'outil, je me renseignerai avant d'écrire l'article. Pour ceux qui ont trouvé des utilisations astucieuses de cURL et Wget, n'hésitez pas à me les envoyer par mail aussi.

Pour aller plus loin :

http://curl.haxx.se/docs/http_scripting.html : manuel cURL intéressant (j'y ai emprunté quelques exemples).

<http://linuxtuts.blogspot.com/2008/03/tutorials-on-wget.html> : manuel Wget intéressant (j'y ai emprunté quelques exemples).

```
[lswest@Monster:~]-[15:07:18]
└─> wget -r -ll http://localhost/current
--2010-07-17 15:07:47-- http://localhost/current
Resolving localhost... 127.0.0.1
Connecting to localhost|127.0.0.1|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://localhost/current/ [following]
--2010-07-17 15:07:47-- http://localhost/current/
Reusing existing connection to localhost:80.
HTTP request sent, awaiting response... 200 OK
Length: 1061 (1.0K) [text/html]
Saving to: "localhost/current/index.html"

100%[=====]
2010-07-17 15:07:47 (187 MB/s) - "localhost/current/index.html"
```



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à : lswest34@gmail.com.



Ah là là !

Le lecteur Inkimar m'a fait remarquer que, lors de la discussion sur l'utilité de wget comme récupérateur de PDF, il n'était pas clair si l'article faisait référence aux deux outils cURL et wget. CURL peut être capable de remplir cette fonction, mais il enverrait le code binaire du PDF sur STDOUT, résultant en un vrai charabia. Je pousse tous ceux qui ne savent pas quel programme utiliser à suivre ce simple conseil : cURL est pour les pages Web et les fichiers en ligne dont vous voulez faire l'analyse et wget sert à télécharger les fichiers Web que vous voulez stocker.

Ce mois-ci, je ne vais pas vous révéler de concepts extrêmement nouveaux ; au contraire, mon objectif est de consolider quelques-uns des programmes et idées que je vous ai expliqués dans les articles précédents. Plus particulièrement de vous montrer comment la plupart de mes articles peuvent vous aider face à un écran tty ou quand, sur l'ordinateur de quelqu'un d'autre, vous ne pouvez vous fier qu'aux lignes de

commande. Mon intention est double. D'abord, Je veux que mes lecteurs se sentent à l'aise avec un système en ligne de commande, pour que, si le serveur X meurt, ils puissent continuer à travailler pour le réparer au lieu de tout réinstaller à partir de zéro ou de suivre des instructions aveuglément. Ensuite, je vais traiter de quelques distributions (et systèmes Unix) qui n'installent pas grand chose de plus qu'une interface en ligne de commande, laissant le reste aux bons soins de l'utilisateur. Ainsi, je peux me concentrer sur le processus d'installation. Aujourd'hui, je traiterai de la recherche de solutions aux problèmes et le mois prochain je parlerai des programmes utiles à l'installation (fdisk, mkfs, etc.), afin de coïncider avec la première machine virtuelle Unix que nous allons créer.

Je présume que beaucoup d'entre vous sont habitués à résoudre des problèmes technologiques. En général, on suit ces étapes : deviner ce qui ne va pas (ou trouver les journaux d'erreur), rechercher le message d'erreur ou le problème sur Google, vérifier si l'un des résultats résout votre problème et, si vous n'avez pas réussi à le résoudre, poser la question dans un forum.

Cependant, que feriez-vous si vous étiez bloqué sur un écran en ligne de commande, sans accès immédiat à Firefox, Nautilus, ou d'autres programmes de ce type ? Je me rends compte qu'il est fréquent d'avoir plusieurs ordinateurs dans un foyer de nos jours, mais j'ai toujours trouvé inefficace d'utiliser un deuxième ordinateur pour dépanner le premier. Tant que l'ordinateur est connecté à Internet, vous pouvez y faire votre diagnostic. Tout ce qu'il vous faut c'est ifconfig/iwconfig (et peut-être wpa_supplicant), dhclient, cd/lis (ou quelque chose comme midnight commander [Ndt : un gestionnaire de fichiers en mode texte]), vim/nano, et elinks. Bien sûr, midnight commander et elinks ne sont généralement pas installés dès le départ (ni wpa_supplicant, je crois), mais ce sont des outils qu'il vaut mieux avoir à portée de main, au cas où. D'ailleurs ils ne prennent pas beaucoup de place.

Pour vous assurer que vous êtes connecté à Internet, vous pouvez faire un ping rapide :

```
ping -c 3 google.com
```

Si vous avez des réponses, vous êtes connecté, sinon, vous aurez pro-

bablement un peu de travail. Je pars du principe que le réseau n'est pas connecté.

D'abord, il faut savoir comment vous vous connectez à Internet : à partir d'un câble Ethernet ou d'une connexion sans fil chiffrée avec WEP, chiffrée avec WPA ou bien sans mot de passe ?

Si c'est une connexion Ethernet, vous avez juste besoin de ifconfig et dhclient. Tapez les commandes suivantes pour savoir si l'interface Ethernet est « up » (activée) :

```
sudo ifconfig
```

S'il y a une interface « eth0 » dans la liste, c'est bon et vous avez juste à lancer :

```
sudo dhclient eth0
```

La commande va demander une adresse IP à votre routeur et, après cela, Internet devrait marcher sans problème. Si votre interface n'est pas listée, c'est parce qu'elle est « down » (désactivée). Pour l'activer, tapez :

```
sudo ifconfig eth0 up
```

Et lancez la même commande dhclient que ci-dessus. Si vous avez plusieurs cartes Ethernet, vous pouvez avoir une liste de toutes les interfaces possibles avec :

```
sudo ifconfig -a
```

Cela ne doit pas être nécessaire pour la plupart des PC.

Si vous avez une connexion sans fil chiffrée avec WEP (ou sans mot de passe), vous aurez besoin de iwconfig et dhclient. D'abord, assurez-vous d'avoir le mot de passe/clé de chiffage, et le ESSID [Ndt : Service Set Identifier, le nom de votre réseau ou, en wifi, l'identifiant de 32 caractères propre à votre réseau] prêt. Puis saisissez les commandes suivantes :

```
sudo iwconfig $interface  
ssid $ESSID key $KEY
```

En remplaçant « \$interface » par le nom de l'interface (d'habitude eth1 ou wlan0, vous pouvez le vérifier en faisant juste un iwconfig sans argument ou un ifconfig), \$ESSID avec le nom du réseau sans fil (on peut le trouver avec « iwlist scan »), et \$KEY avec le mot de passe (mot de passe ASCII utilisé pour se connecter) ou la clé hexadécimale de chiffage (actual hex string). Si la clé est un mot de passe, vous devrez y ajouter

un « s: ». Dans cet exemple l'interface est wlan0, le ESSID est home et le mot de passe est passkey :

```
sudo iwconfig wlan0 essid  
home key s:passkey
```

Une fois que vous avez entré cette commande, vous pouvez demander une adresse IP en tapant :

```
sudo dhclient $interface
```

Assurez-vous de bien remplacer « \$interface » par le nom de l'interface. Si cela échoue, vous avez peut-être besoin d'options supplémentaires pour la commande iwconfig (canal, etc.) ce qui est expliqué clairement dans les pages man. Ou bien vous êtes peut-être en train d'essayer de vous connecter à un réseau sécurisé par WPA.

Pour ceci, vous avez besoin de wpa_supplicant. Il faut d'abord créer l'information qui sera traitée par wpa_supplicant. Pour faire ceci, lancez cette commande :

```
wpa_passphrase $ESSID  
$passphrase > ~/passphrase.txt
```

En remplaçant « \$ESSID » par le vrai ESSID, et « \$passphrase » par la vraie phrase de passe. Le chemin de fichier après le « > » est comme vous le voulez. Un fichier dans votre répertoire /home

appelé passphrase.txt sera alors créé. Il ressemblera à ceci :

```
network={  
    ssid="test"  
  
    #psk="testing123"  
  
    psk=a9ff0c9d1f2367bccf9959e95  
bc08695bf411f82b146c55b9486dd  
b17495f39d  
}
```

Vous pouvez alors vous connecter à votre réseau par la commande (mieux vaut la lancer dans un deuxième terminal car elle continuera à tourner, mais pas en arrière-plan) :

```
sudo wpa_supplicant -i$inter-  
face -c$file -D$driver
```

En général, il n'y a pas d'espaces entre les arguments et les commutateurs d'options, mais c'est comme vous voulez. Remplacez « \$interface » par votre nom d'interface (habituellement wlan0), « \$file » avec le chemin du fichier de configuration que nous avons créé à l'étape précédente, et « \$driver » avec le pilote de votre carte (la plupart du temps, wext vous le donnera, mais d'autres pilotes sont listés dans les pages man). Une fois que la commande aura commencé à tourner, revenez dans un terminal libre et

lancez :

```
sudo dhclient $interface
```

Remplacez « \$interface » par le vrai nom de l'interface.

Une fois connecté, lancez une nouvelle fois la commande ping. Si ça fonctionne, c'est terminé. Vous pouvez alors aller dans /var/log/ ou là où vous devez aller, et vérifiez les journaux avec cat. Une fois que vous aurez décidé de chercher un terme sur Google, ouvrez elinks avec la commande :

```
elinks
```

Par défaut, elinks vous demandera directement une URL, qui sera en général google.com. Une fois la page chargée, utilisez les flèches pour aller dans la case de recherche (affichée avec des soulignés), si elle n'est pas déjà choisie. Puis appuyez sur Entrée pour valider et tapez votre mot à chercher, puis appuyez sur Entrée de nouveau pour commencer la recherche. Utilisez les flèches pour choisir les liens et Entrée pour les suivre. Si vous tombez sur un fichier à télécharger, surlignez simplement le lien vers ce fichier, tapez Échap, allez à Liens, et choisissez « download link » (télécharger lien) (ou tapez « d »). Vous pouvez vérifier les téléchargements

soit en tapant Échap, puis en allant dans Outils et choisissant Téléchargements, ou bien en tapant « D » (shift+d). Une fois que vous aurez terminé vos recherches, vous pouvez fermer elinks avec « q » ou Échap (pour afficher le menu), puis Fichier et Quitter.

J'espère que ce guide rapide (et il était vraiment rapide, il y a tant d'autres choses à dire) aidera ceux qui suivent ma série sur la virtualisation et ceux qui se trouvent confrontés à un terminal. Le mois prochain, je parlerai de ce qui est nécessaire pour installer un système d'exploitation en utilisant la ligne de commande et comment gérer les partitions en ligne de commande. Si quelqu'un a des questions ou veut des explications plus détaillées concernant tout ce que j'ai abordé dans cet article, il peut me contacter à l'adresse : lswest34@gmail.com. Assurez-vous de mettre « C&C » ou « FCM » en tant qu'objet, pour que je voie votre message.



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à lswest34@gmail.com.

U^3 (U-Cubed) - 28 août 2010

Le 28 août, MadLab est l'hôte de l'U^3 (U-Cubed [Ndt : U au cube]), une journée de non-atelier Ubuntu et en amont en collaboration avec HacMan, ManLUG et Manchester Free Software. Cette journée est inspirée par l'Ubuntu Global Jam, événement qui se déroule le même week-end.

Il s'agit d'une occasion de montrer comment aider les utilisateurs à s'impliquer dans un travail de distribution plus large, à la fois dans les projets qu'ils utilisent actuellement (peut-être Ubuntu, ou peut-être juste une application libre et Open Source particulière), mais aussi dans les projets en amont, comme Debian, Gnome et autres. Il est inspiré par les événements du Ubuntu Global Jam qui se tiennent le même jour.

Nous espérons trouver le jour même une aide de personnes expérimentées avec Ubuntu, mais aussi des gens qui sont impliqués dans plus de programmes que Ubuntu, aussi essayons-nous d'établir un contact avec des personnes de la région nord-ouest du Royaume Uni pour voir si les gens sont prêts à aider (même si Ubuntu n'est pas la distribution Linux que vous utilisez). Aussi, si vous êtes intéressé(es) et disponible(s) entre 11h et 21h et pouvez vous rendre à Manchester, ou si vous pouvez seulement être présent une partie de la journée, prenez contact avec moi, ou, mieux encore, allez à <http://u-cubed.eventbrite.com> pour réserver une place.

Comme notre espace est limité, nous ne pouvons allouer que 60 billets et, pour que ce soit équitable, nous y donnons accès à 13 h le jeudi 12 août (donnant ainsi aux gens l'occasion de découvrir l'événement), mais si vous pouvez venir pour aider avec des informations techniques ou des conseils, faites-le moi savoir et je ferai en sorte que vous soyez invité quand les portes s'ouvriront au flot de visiteurs !

J'espère vraiment que vous pourrez venir ce jour-là et aider à faire de ce jour une grande réussite !

Les Pounder





COMMAND & CONQUER

Écrit par Lucas Westermann

Ce mois-ci, j'ai décidé de parler à la fois des outils nécessaires à des installateurs en ligne de commande (fdisk, mkfs, etc.), et aussi des outils utiles pour rechercher des fichiers sur le disque dur, au cas où vous décideriez de chercher les fichiers de configuration d'un système nouvellement installé, ou sur votre vieux système au cas où vous voudriez les transférer sur le nouveau. Il y a environ deux ans, j'ai installé pour la première fois Arch Linux et, ce faisant, j'ai appris beaucoup de choses sur l'interface de ligne de commande. J'ai trouvé la plupart d'entre elles très utiles. Pour rendre ma série sur la virtualisation un peu plus accessible à mes lecteurs, j'ai cherché également à étendre certaines des idées de cette série, de façon à essayer d'en faire un tout cohérent.

Fdisk :

J'utilise fdisk principalement pour lister les partitions d'un système installé ; pour cela, saisissez simplement :

```
sudo fdisk -l
```

Cela devrait donner un résultat semblable à celui qui est ci-dessus à droite.

```
Disk /dev/sda: 320.1 GB, 320072933376 bytes
255 heads, 63 sectors/track, 38913 cylinders, total 625142448 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x76692ca8
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1		2048	30716279	15357116	1c	Hidden W95 FAT32 (LBA)
/dev/sda2	*	30716280	186996599	78140160	7	HPFS/NTFS
/dev/sda3		186996600	625137344	219070372+	f	W95 Ext'd (LBA)
/dev/sda5		186996663	543109454	178056396	7	HPFS/NTFS
/dev/sda6		543109518	570452084	13671283+	83	Linux
/dev/sda7		570452148	625137344	27342598+	83	Linux

Comme vous le constatez, la première moitié donne des informations sur le disque dur lui-même et la seconde moitié donne des informations sur les partitions.

Si vous voulez éditer la table des partitions, entrez la ligne suivante :

```
sudo fdisk /dev/sda
```

Assurez-vous de remplacer « /dev/sda » par le disque à éditer. Cette commande vous amènera à une invite qui ressemble à ceci : « Commande (m pour l'aide): ». Si vous saisissez « m », vous obtiendrez la liste des commandes possibles.

Commande d'action

```
a basculer le fanion d'amorce
b éditer l'étiquette BSD du
disque
c basculer le fanion de compa-
tibilité DOS
d supprimer une partition
l lister les types de parti-
tions connues
m afficher ce menu
n ajouter une nouvelle parti-
tion
o créer une nouvelle table
vide de partitions DOS
p afficher la table de parti-
tions
q quitter sans enregistrer
les changements
s créer une nouvelle étiquet-
te vide pour disque de type Sun
t modifier l'« id » (l'identi-
fication) de système de fichiers
d'une partition
```

```
u modifier les unités d'affi-
chage/saisie
v vérifier la table de parti-
tions
w écrire la table sur le disque
et quitter
x fonctions avancées (pour ex-
perts seulement)
```

Comme vous le voyez, les instructions sont assez claires. Si vous entrez une commande, vous aurez à entrer des informations complémentaires ensuite. Notez qu'en créant une partition (avec la commande « n »), on vous demandera à partir de quel cylindre elle doit commencer, la réponse proposée par défaut est généralement la bonne (sauf si vous souhaitez laisser de l'espace libre entre les parti-

tions). On vous demandera également où doit se terminer la partition ; vous pouvez saisir un numéro de cylindre ou « + 1024 M » pour 1 Go plus loin et ainsi de suite. Les octets et kilo-octets sont également acceptés comme unités, même si la plupart des gens utiliseront les méga-octets. Une fois les partitions créées, utilisez « a » pour rendre la première partition amorçable, et « t » pour modifier le format de la partition (« l'id de système »). Pour choisir le type de partition, il faudra choisir un nombre hexadécimal et non pas un nom ; on peut afficher une liste des codes avec « L ». Vous aurez par exemple :

Hex code (type L to list codes): 82

Comme vous pouvez le voir, ceci est l'invite de commande où vous pouvez saisir « L », et 82 est le code hexa pour le type « Linux swap » (ou disque d'échange). Fdisk peut aussi vous donner des informations sur des problèmes possibles avec les réglages que vous avez choisis, elles sont en général claires et au pire sont suffisantes pour trouver une solution sur Google. Une fois que vous êtes à l'aise avec fdisk, vous pourrez également faire tout ce que vous souhaitez en utilisant des arguments de ligne de commande, au lieu de faire les choses une par une.

Mkfs :

Vous vous demandez peut-être pourquoi vous auriez besoin de fabriquer un système de fichiers (mkfs signifie « make file system ») alors que vous pouvez simplement utiliser fdisk pour tout cela. C'est plutôt simple : la plupart des gens ne veulent pas utiliser fdisk pour tout faire, car c'est parfois plus compliqué que nécessaire. Si la partition est créée et que vous voulez juste la reformater, mkfs est probablement un meilleur outil pour le faire. Il existe plusieurs façons d'utiliser mkfs, listées ci-dessous.

```
mkfs /dev/sdXY
```

```
mkfs.ext2 /dev/sdXY
```

```
mkfs -t ext2 /dev/sdXY
```

Chacune de ces commandes formatera « sdXY » (remplacez X par la lettre du disque et Y par le numéro de partition) en ext2 (le format par défaut de mkfs). La première commande ne fonctionne qu'avec des partitions de type ext2, car elle choisit par défaut ext2 si on ne précise pas de type.

Cette commande vous permet aussi de préciser la taille des blocs, le nom des volumes, le pourcentage de blocs réservés pour le super-utilisateur (« root »), un UUID, etc. Pour avoir une liste complète,

examinez la page de manuel de mkfs et de mkfs.<type> (remplacez type par le format réel). Je n'approfondirai pas plus pour cette fois, car je n'ai pas sous la main ma machine de test pour la formater. Si quelqu'un veut des explications plus détaillées sur mkfs, envoyez-moi un courriel.

Find :

Combien d'entre vous ont déjà cherché un fichier et découvert que Nautilus ne le trouve pas ? Je sais qu'il y a des alternatives pour faire une recherche de fichier, mais je sais aussi que « find » est un utilitaire intégré à la plupart des distributions et qu'il est donc important de le connaître.

```
find /home/ -name "*~"
```

Cette commande cherchera dans le répertoire /home/ (et ses sous-répertoires, comme les comptes utilisateurs) tous les « fichiers tilde » (des sauvegardes automatiques de fichiers en cours d'édition). Je trouve cela utile pour écrire des scripts de nettoyage de certains fichiers. Comme vous vous en doutez sans doute, « -name » permet de n'afficher que les fichiers dont le nom contient ce qui est entre les guillemets. Dans ce cas, je lui demande d'afficher tout ce qui se termine par un ~ (l'astérisque est un caractère joker qui signifie « n'importe quelle suite de caractères »). Vous pouvez préciser

[...] vous avez déjà cherché un fichier, et découvert que Nautilus ne le trouve pas ?

n'importe quel chemin à la place de /home/. Notez que si vous essayez de chercher dans un répertoire pour lequel vous n'avez pas les droits de lecture, ou voulez simplement rechercher sur le disque entier, vous devrez exécuter la commande find en tant que root avec sudo. Sinon, vous obtiendrez des erreurs « permission non accordée ». Si vous voulez utiliser cette commande dans un script, et ne voulez pas chercher dans les répertoires pour lesquels vous n'avez pas les droits, vous devrez utiliser un « grep -v » (grep inverse). Je trouve que c'est une excellente ressource si vous pouvez attendre un peu plus longtemps les résultats de la recherche (ça peut prendre beaucoup de temps lorsqu'on recherche dans des gros répertoires), mais que vous souhaitez obtenir des résultats à jour et précis. Sinon, pour une recherche rapide du style « où est ce fichier ? », j'utilise « locate ».

Locate :

C'est un programme qui utilise une base de données indexée de fichiers, pour

trouver rapidement une réponse. Cette base est mise à jour régulièrement, mais vous devriez prendre l'habitude de forcer une mise à jour avant de lancer la recherche d'un fichier récent. Vous pouvez faire cela avec la commande suivante :

```
sudo updatedb
```

Cela peut prendre quelques minutes, mais n'est pas nécessaire à chaque fois ; seulement si vous cherchez quelque chose que vous venez de télécharger ou d'installer. Une fois la base à jour, vous pouvez lancer une recherche avec la commande suivante :

```
locate "*~"
```

Comme vous l'avez peut-être remarqué, puisque locate cherche dans une base de données, vous obtiendrez les résultats provenant de tous les répertoires du disque dur, même ceux pour lesquels vous n'avez pas les permissions d'utiliser find, car updatedb est exécuté en tant que root. Si vous obtenez trop de résultats, vous pouvez utiliser « grep », « head » ou « tail » pour réduire les résultats affichés. Il y a encore bien d'autres commandes utiles pour rechercher des fichiers, mais je ne parlerai que de « where » et de « whereis ».

Where/Whereis :

Ces commandes sont faites pour vous aider à trouver rapidement des programmes et des fichiers de configuration. Par exemple, si vous installez skype et exécutez ces commandes, vous verrez :

```
where skype
```

```
résultat affiché :  
/usr/bin/skype
```

```
whereis skype
```

```
résultat affiché : skype:  
/usr/bin/skype.real  
/usr/bin/skype  
/usr/bin/skype.bak2  
/usr/share/skype
```

Comme vous le constatez, cela renvoie des résultats qui sont liés au programme binaire. Si vous essayez d'exécuter where ou whereis sur un répertoire, vous n'obtiendrez aucun résultat. Ces commandes servent à trouver rapidement les fichiers de configuration d'un programme ou l'emplacement d'un exécutable. Elles ont aussi la propriété supplémentaire de vous donner l'impression d'être un homme des cavernes.

J'espère que vous avez trouvé ces explications claires et que vous essaieriez certaines de ces astuces la prochaine fois que vous rechercherez un fichier ou voudrez reformater un disque. Comme

toujours, si vous avez des questions, des commandes ou des suggestions, vous me trouverez à l'adresse lswest34@gmail.com. Assurez-vous de mettre « C&C » ou « FCM » comme objet de votre courriel pour que je ne le néglige pas.

Pour aller plus loin :

<http://www.linfo.org/mkfs.html>

(ressource utile pour mkfs).

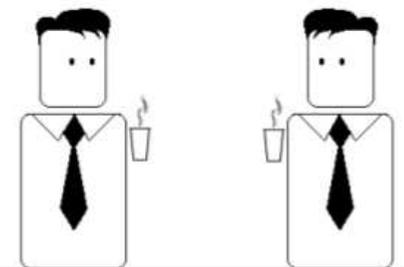
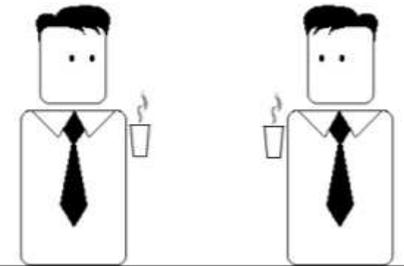
http://tldp.org/HOWTO/Partition/fdisk_partitioning.html (ressource utile pour fdisk).

Les pages de manuel de toutes les commandes sont également un excellent point de départ.

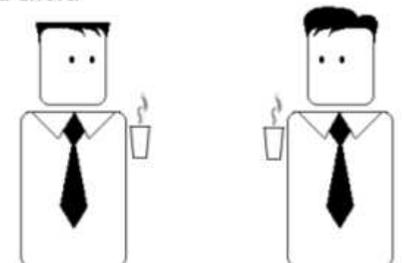


Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à : lswest34@gmail.com.

Je pense que dans la vie de tout employé vient un moment où on devient maximaliste.



On veut un maximum de salaire pour un minimum d'effort.



by Richard Redei



COMMAND & CONQUER

Écrit par Lucas Westermann

Avant de commencer l'article de ce mois, je veux prendre le temps d'expliquer que, dès le mois prochain, j'étendrai Command & Conquer aux applications avec interface graphique et en ligne de commande, aux trucs et astuces, et aux savoir-faire en général. Ceci est dû à plusieurs raisons dont la principale est que je sens que je parlerai à plus d'utilisateurs débutants de cette manière. Cela signifie que, dans le futur, mes articles couvriront les deux façons de faire, graphique et ligne de commande (quand c'est possible). Cela peut aussi nécessiter un changement de titre, mais à moins de trouver quelque chose d'aussi astucieux, il y a de grandes chances qu'il reste le même.

Correction : dans le Command & Conquer du mois dernier j'ai parlé d'une commande « where ». En revenant sur mon système, j'avais fait un alias de « which » en « where » il y a très longtemps, et je l'avais oublié ! Cela signifie que si vous avez des problèmes avec la commande « where », c'est parce que c'était en fait « which ». Désolé de cette confusion ! Et merci à Harold et Sue pour me l'avoir fait remarquer tous les deux.

Cela nous mène à l'article du mois. Je veux décrire juste quelques commandes de base, mais essentielles (de mon point de vue), que tout un chacun, appréciant ou pas la ligne de commande, devrait au moins connaître en raison de leur importance et de leur flexibilité dans les systèmes Linux. Je veux aussi prendre le temps de préciser que, en dépit de votre opinion personnelle sur la ligne de commande, les solutions sont souvent présentées en ligne de commande sur internet, car cela s'applique à un grand nombre de distributions, à l'opposé des méthodes avec interface graphique qui nécessitent que vous ayez un certain environnement de bureau ou un ensemble de programmes spécifiques. Et, même si vous ne choisissez pas par vous-même d'utiliser la ligne de commande, vous pourriez être forcé de le faire à un moment donné, et avoir une compréhension rudimentaire de ce que vous faites est extrêmement utile.

Voici mes commandes « essentielles », ainsi qu'une description de chacune :

- **man** : affiche une page man (page de manuel) pour une commande donnée (usage : man <commande>).

- **vi(m)** : Vi (ou son dérivé Vim) sont des éditeurs de texte qui offrent un tas de fonctionnalités et vous permettent de modifier des fichiers sans démarrer d'application graphique. À cause des instructions pour sauver, sortir et ouvrir des fichiers, les débutants peuvent se sentir plus à l'aise avec « nano ».

- **ls** : liste tous les fichiers d'un répertoire (par défaut seulement les fichiers non cachés, mais les arguments de la ligne de commande peuvent modifier cet affichage).

- **cd** : ça veut dire « change directory » [Ndt : Changer de répertoire] et cela vous permet de naviguer dans l'arborescence des fichiers (usage : cd <répertoire ou chemin>).

- **ping** : un programme utile pour tester l'état du réseau (la connectivité) d'un serveur. (usage : ping -c 5 www.google.com) – mettre l'option -c 5 si vous ne voulez pas pinguer sans fin.

- **iwconfig/ifconfig** : commandes utilisées pour configurer les cartes sans fil et ethernet par la ligne de commande (des clients wpa sont nécessaires pour les réseaux wpa, et dhclient pour obtenir une adresse IP). À cause du grand nombre d'options, merci de voir la page man de iwconfig ou ifconfig

Voici mes commandes « essentielles » et une description de chacune...

pour connaître l'usage.

- **halt** : un programme semblable à « shutdown -h now », mais ne demande pas d'autres informations. (Usage : sudo halt). S'il n'est pas disponible sur votre système, utilisez shutdown.

- **alias** : extrêmement utile pour raccourcir des commandes longues et des chaînes d'arguments que vous utilisez souvent. Usage : alias <nom_alias>="<commande>".

- **rsync/scp** : vous permettent de copier des fichiers d'un hôte distant par ssh (ou local dans le cas de rsync). Rsync offre aussi une barre de progression, de l'information sur les fichiers, et bien plus d'information que ce qui est disponible avec cp.

- **cp/mv** : cp vous permet de copier des fichiers d'un répertoire (ou un répertoire entier) vers un autre, et mv vous permet de déplacer un fichier ou un répertoire (soit pour le renommer,

soit pour le bouger vers un autre endroit).

- **cat** : un programme qui affiche le contenu d'un fichier (utile en combinaison avec `less` ou `more` pour les fichiers longs). Extrêmement utile pour lire rapidement les journaux (un exemple : `sudo cat /var/log/errors.log|tail -150 | more`). Cela montrera les 150 dernières lignes de `/var/log/errors.log` et les affichera dans `more` (qui vous permet de vous déplacer lentement dans l'affichage). Pour plus d'infos sur comment débugger les erreurs, voir C&C du FCM n° 23, page 6.

- **rm** : vous permet de détruire des fichiers ou des répertoires complets (utilisez `rmdir` pour les répertoires vides).

- **su** : vous permet de changer d'utilisateur en ligne de commande (usage : `su <nom_d'utilisateur>`).

- **locate** : vous permet de trouver rapidement des fichiers sur votre ordinateur (lancez `sudo updatedb` auparavant pour avoir de meilleurs résultats).

- **ln** : vous permet de créer des liens physiques ou symboliques de répertoires ou de fichiers (pour les liens symboliques, utilisez l'option « `-s` »). Les liens symboliques sont comme les raccourcis, et les liens physiques sont des références physiques sur le disque vers un autre endroit (pensez à un fichier original comme le moyen d'atteindre un endroit A et à un lien phy-

sique comme un ensemble de chemins vers la même position finale mais à partir d'un endroit B). Ou, pour ceux qui parmi vous jouent au monopoly, les liens physiques sont des cartes « Passez par la case départ » et les liens symboliques « Allez directement à <cet-endroit>, ne passez pas par la case départ ».

- **echo** : comme la commande `php` du même nom, cela affiche simplement toute chaîne qui lui est fournie après. Utilisé le plus souvent pour ajouter rapidement une ligne à un fichier, ou pour le créer.

- **pwd** : `print working directory` est une commande qui affiche l'endroit où vous êtes sur votre système de fichiers.

- **mkdir** : commande pour créer un répertoire.

- **touch** : crée un fichier vide à l'endroit donné (usage : `touch <chemin et nom de fichier>`). Si le fichier existe, `touch` ne le remplacera pas.

- **grep** : vous permet de rechercher une chaîne de caractères dans la sortie d'une commande (ou dans un fichier).

- **find** : une recherche plus lente, mais plus approfondie que `locate`. (Usage : `find <chemin> <arguments>`).

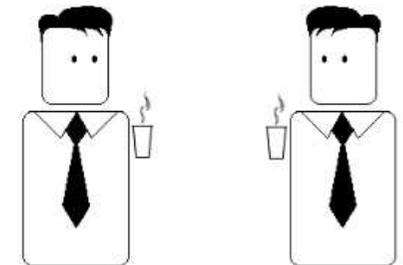
Je me rend compte que ça semble être une longue, longue liste, mais j'y ai inclus des commandes en double, et

certaines ne sont utilisées qu'accompagnées d'autres. De plus, si vous restez coincé dans l'environnement en ligne de commande, ces commandes peuvent vous aider à retourner à l'interface graphique. Il y a une commande que je n'ai pas encore décrite, car elle n'est pas standard. J'installe toujours un navigateur web en texte, juste au cas où. Je préfère personnellement `lynx`, mais `links`, `elinks`, et `w3m` sont aussi très utiles. Même si vous ne connaissez qu'un tiers de cette liste, vous devriez être préparé à résoudre la plupart des erreurs d'un xserver (ou au moins à détecter les erreurs), pour pouvoir rétablir votre environnement graphique. Si vous avez d'autres commandes « essentielles » que vous pensez que je devrais décrire ou mentionner, envoyez-moi un courriel à : lswest34@gmail.com. Et, comme toujours, merci d'inclure « C&C » dans le sujet. J'espère que vous avez aimé cet article, et que vous vous sentez mieux préparés pour tout travail en ligne de commande que vous pourriez avoir à faire.

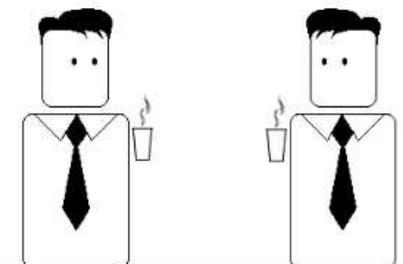


Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à : lswest34@gmail.com.

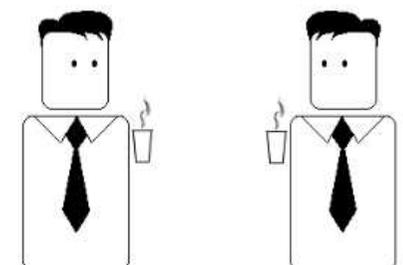
J'ai toujours admiré Picasso.



Pas tellement ses oeuvres.



Mais le fait qu'il arrive à les vendre.



by Richard Redet



Avant de commencer l'article de ce mois-ci, je pense que je dois présenter mes excuses quant à mon utilisation de « cat », car un lecteur a pensé que cette utilisation méritait une récompense pour la « pire utilisation de cat possible ». Je sais que « tail » accepte aussi des fichiers et que donc l'utilisation de « cat » était superflue, mais je voulais donner un exemple de ce que fait « cat », non pas indiquer la meilleure utilisation qu'on peut en faire ! Si je n'avais pas fait comme j'ai fait, il aurait fallu un autre exemple de commande que je trouvais de trop. Excusez-moi si j'ai gêné (ou offensé) quelqu'un en faisant ainsi.

Il y a un an, j'ai commencé à apprendre le japonais (principalement pour relever le défi) et le premier obstacle véritable que j'ai rencontré a été de trouver un système pour saisir des choses en japonais. Comme je n'avais jamais entendu parler de iBus, j'ai décidé d'utiliser SCIM (Smart Common Input Method [Ndt : méthode de saisie intelligente commune]). Aujourd'hui, Ubuntu est livré avec iBus, qui ressemble à SCIM, bien que plus facile à configurer.

Je suppose que de nombreux lec-

teurs du Full Circle ont besoin de faire des saisies en japonais ou en chinois. Voici comment activer la saisie étendue.

Étape 1

Ouvrez la fenêtre de gestion des langues (dans le menu Système → Administration → Prise en charge des langues sous Ubuntu 10.10).

Étape 2

Choisissez « Installer/Supprimer des langues » dans la fenêtre principale (voir ci-dessous).



Étape 3

Parcourez la liste et cochez la case à la fin de la ligne de la langue qui vous intéresse (dans mon cas, il s'agissait du japonais, voir figure 2). Après avoir choisi la langue et fermé la fenêtre, les paquets de langue nécessaires pour la langue choisie (les polices, les dictionnaires, les « locales » [Ndt : paramètres régionaux], etc.) seront installés.



Étape 4

Une fois de retour dans la fenêtre principale du choix des langues, vous devrez choisir « iBus » dans la liste déroulante à côté de « Système de saisie au clavier ».

Étape 5

Déconnectez-vous puis reconnectez-vous pour activer les systèmes et démons supplémentaires requis par iBus (vous pourriez le faire à la main, mais ça pourrait poser des problèmes).

Étape 6

Ouvrez les préférences iBus en allant dans Système → Préférences → Méthodes de saisie au clavier. Puis allez sur « Méthode de saisie » (voir les figures 3 et 4 page suivante) et choisissez votre langue dans la liste déroulante (japonais > Anthy ou une méthode choisie dans le menu chinois). Je ne peux pas recommander une méthode de la section chinoise, car je n'utilise pas la saisie en chinois, mais vous pouvez facilement changer si vous vous apercevez que la méthode choisie ne vous plaît pas. Une fois que vous en avez choisi une, vous

devez cliquer sur « ajouter » pour accepter réellement cette méthode-là. La première en haut de la liste est la langue par défaut (celle qui sera activée/désactivée lorsque vous activez ou désactivez iBus).



Étape 7

Pour utiliser iBus, il vous suffit d'appuyer sur Ctrl + Espace pour l'acti-

ver. Puis, si vous saisissez ローマ字 en japonais par exemple, vous saisirez en rōmaji (c'est-à-dire en alphabet latin), mais les symboles apparaîtront lorsque vous saisissez leur prononciation phonétique.

Pour générer les Kanji (漢字) pour un mot, appuyez simplement sur espace après l'avoir saisi (ainsi にっぽんご devient 日本語).

S'il y a plusieurs possibilités, appuyez à nouveau sur espace pour afficher la liste des Kanji possibles.

Voici tout ce dont vous avez besoin pour commencer à saisir en japonais. J'espère que vous trouverez ceci utile et, comme d'habitude, si vous avez des questions, des suggestions ou des demandes, envoyez-moi un courrier à lswest34@gmail.com. N'oubliez pas d'inclure « Command & Conquer » ou « C&C » dans « Objet » pour que je ne manque pas votre message.



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à : lswest34@gmail.com.

Podcast Full Circle

Dans cet épisode, UDS, Unity et aucune arme...

Dans l'épisode n° 13 :

- **Critique** : Numéro 42 du magazine Full Circle.
- **Actus** : Ubuntu Developers Summit (UDS), Unity interface.
- **Opinion** : Présentant « Ubuntu le film » (apparemment, il ne s'agit pas d'une publicité...)
- **Jeux** : Le jeu qui déferle sur le monde : Minecraft et Ed vous donnent un aperçu du nouveau jeu d'énigmes indépendant « And Yet It Moves » (Et pourtant elle tourne).

Taille des fichiers :

OGG 29,3 Mo.

mp3 23,8 Mo.

Durée : 54 minutes 40 secondes.

Sortie : 6 novembre 2010.

<http://fullcirclemagazine.org/>



Full Circle
Podcast





Dernièrement, j'ai pu voir un large afflux d'installations intrigantes de Conky sur les forums Arch Linux. Cela m'a donné l'idée de partager quelques trucs et astuces que j'utilise dans toutes mes installations. Je ne vais couvrir qu'une seule astuce précise ce mois-ci, mais je vous donnerai des trucs en plus avec lesquels vous pourrez vous amuser. Tout d'abord, pour ceux d'entre vous qui ne savent pas ce qu'est Conky, il s'agit d'un moniteur système en mode texte qui peut être affiché sur le bureau, en sortie dans dzen (populaire parmi certains gestionnaires de fenêtre en mosaïque) ou flottant de lui-même comme panneau.

Par défaut, Conky possède de nombreuses options (depuis afficher l'heure et la date jusqu'à la consommation de mémoire ou l'espace du disque dur). Toutefois, il ne propose pas un moyen d'afficher le nombre de mises à jour disponibles pour votre machine (ce qui

est compréhensible vu le nombre de formats de paquets et de systèmes qui existent). Voilà donc où entre en jeu une autre fonctionnalité de Conky : la capacité d'exécuter des scripts personnalisés et de les afficher dans le logiciel lui-même. Il y a deux variantes de cette commande : l'une avec un intervalle de rafraîchissement (ce dont nous aurons besoin pour le gestionnaire de mise à jour) et l'autre pour une seule sortie (utile si le script se rafraîchit de lui-même). Tous ces paramètres sont contrôlés depuis le fichier `.conkyrc` dans votre dossier personnel [Ndt : `home`]. Pour vous mettre le pied à l'étrier, voilà un `.conkyrc` basique que j'utilise (je lui ai ôté tout script personnalisé, puisque je ne projette pas de tous les partager) : <http://fullcirclemagazine.pastebin.com/SwwHqFSm>

Comme vous pouvez le constater, j'ai pris la liberté de commenter toutes les options de Conky et vous remarquerez peut-être aussi que « `Updates: »` n'affiche rien pour le moment. Dans un ins-

“ j'ai pris la liberté de commenter toutes les options de Conky...”

tant, j'expliquerai comment je suis arrivé à ce script et vous montrerai comment l'implémenter. D'abord, je voudrais vous le présenter en disant que cet exemple n'est pas le plus court (vous pouvez faire la même chose avec `sed` et quelques expressions régulières), mais c'est le plus lisible auquel je puisse penser. Si vous désirez pratiquer un peu de regex, remplacez `grep` et `cut` par `sed`. Enfin, voici le script :

```
#!/bin/bash
updateChecker=`apt-get -s upgrade|grep upgraded,|cut --delimiter=" " -f1`
echo "$updateChecker";
```

Bon, j'ai trouvé cette commande (ce qu'il y a dans « `updateChecker` ») de façon plutôt simple. J'ai saisi :

```
apt-get -s upgrade
```

et j'ai regardé la sortie, puis j'ai cherché la ligne avec le nombre de mises à jour et trouvé un seul mot sur celle-ci. Puis j'ai relancé la commande en la pipant en « `|grep upgraded,` ». Une fois que j'ai été sûr qu'il me donnait la bonne ligne, je l'ai simplement

prise et coupée en champs (délimités, c'est-à-dire séparés par des espaces) et j'ai affiché le premier champ (« `-f1` ») puisqu'il s'agissait du nombre désiré. J'ai à nouveau relancé la commande entière pour être sûr qu'elle retournait ce nombre correctement, je l'ai copiée dans le script `bash` et écrit la ligne `echo` pour avoir le retour. Si vous voulez obtenir un script moins spécifique à Conky, vous devez seulement ajouter « `Updates: »` (sans les guillemets) à la ligne `echo` (avant `$updateChecker`) et supprimer le mot « `Updates: »` du fichier `.conkyrc`. Pour son implémentation dans Conky, tout ce que vous avez à faire est ajuster la ligne de mise à jour comme ceci :

```
 ${font DejaVuSans:bold:size=8}
Updates:${font} ${execi 300
/chemin/vers/le/script}
```

Bien sûr, `/chemin/vers le/script` sera votre propre chemin et le script est rendu exécutable. Pour ce faire :

```
chmod +x /chemin/vers/le/script
```

Ma dernière vérification est de lancer le script depuis le terminal pour m'assurer qu'il fonctionne bien,

mais, ici, vous pouvez simplement sauter cette étape (après tout, Conky vous laissera voir s'il fonctionne bien !).

Avec un peu de chance, cela aura montré à quelques nouveaux utilisateurs que la ligne de commande n'est pas qu'un texte noir et blanc, mais peut aussi ajouter quelque chose à leur installation graphique. Pour ceux d'entre vous qui sont intéressés par un script Bash qui supprime les noyaux en trop (en ne laissant que les deux derniers), allez voir ci-dessous dans la partie « Encore plus de scripts ». Si vous avez trouvé cela utile et/ou intéressant, je serai heureux de partager d'autres scripts dans les mois qui suivent. Et comme toujours, si vous avez des corrections, des questions ou des suggestions, vous pouvez me les envoyer, en anglais, à lswest34@gmail.com. Gardez à l'esprit que « C&C » ou « FCM » doit apparaître dans le champ objet afin que je ne les manque pas.

Encore plus de scripts

Si quiconque se demande pourquoi je possède un gestionnaire de noyau, je l'ai écrit pour un ordinateur en double-amorçage Ubuntu/ Windows, avec Windows comme option

par défaut de Grub, où toute mise à jour du noyau me foutait en l'air les paramètres par défaut. Au lieu d'apprendre au nouvel utilisateur à mettre Grub2 à jour, j'ai simplement mis un lien symbolique sur le bureau pour le script qu'il doit lancer quand la liste devient trop longue.

Mon « gestionnaire de noyau », pour ceux qui le trouveraient intéressant (qui peut aussi être transformé en un script Conky pour afficher le nombre de noyaux installés) : <http://fullcirclemagazine.pastebin.com/6zifBj9W>

Le programme est plutôt bien commenté, mais voilà en substance ce qu'il fait :

- il vérifie les dossiers dans /usr/src et compte combien de dossiers noyaux Linux s'y trouvent ;
- puis il stocke les véritables noms dans une autre variante ;
- s'il y en a quatre (ou moins), il affiche « rien à faire » et quitte ;
- autrement ($\$dossiers > 4$), il affiche combien de noyaux doivent être supprimés et montre lesquels le sont effectivement (pour des raisons de sécurité) ;
- une fois que l'utilisateur a lu cela, on lui demande s'il doit continuer ;
- si l'entrée est « o » [oui], il supprime

les noyaux et attend trois secondes pour s'assurer que tout est fait ;

- il attend jusqu'à ce que l'utilisateur appuie sur Entrée pour quitter (sinon, tout message d'erreur est perdu) ;
- si aucun des états de « if » ne s'applique, il fait savoir à l'utilisateur qu'il y a eu une erreur.

lswest@lswest-VirtualBox

Sun, 12 December 2010

Up Time: 0h 1m 21s
Updates: 148

Cpu: 5%
Ram: 29%
Swap: 0%

Root: 3.72GiB free
Home 3.72GiB free



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à : lswest34@gmail.com.



Podcast Full Circle

Dans cet épisode, Narwhals, Wayland et... j'ai failli oublier, Amnesia !

Dans l'épisode n° 14 :

- * **Critique** : numéro 43 du FCM.
- * **Actus** : Ubuntu 11.04 Alpha1, Wayland, System 76, Android 2.3, des jeux à cotisation dans la logithèque Ubuntu et Flash 10.2 Bêta.
- * **Jeux** : Humble Indie Bundle n° 2 et Amnesia.

Taille des fichiers :

OGG 42,5 Mo
mp3 34,9 Mo

Durée : 1 h 18 mn 58 s

Sortie : 19 décembre 2010

<http://fullcirclemagazine.org/>



Dans l'article de ce mois-ci, je vais traiter de quelques trucs supplémentaires à faire avec Conky pour tous ceux qui voudraient un moniteur système avec une interface un peu plus graphique. Cependant, avant que je ne commence, si quelqu'un se demande pourquoi j'ai créé un script pour les mises à jour au lieu de mettre le code à l'intérieur du fichier `.conkyrc`, vous pourrez aller voir ma réponse dans la rubrique Courriers de ce numéro.

Si vous avez déjà vu une capture d'écran de mon bureau (<http://lswest.deviantart.com/>), vous aurez remarqué que j'ai une section « Music » qui est très visible dans le Conky principal. Si vous vous demandez de quoi je parle, allez voir la capture d'écran la plus récente (affichée à droite) ici <http://lswest.deviantart.com/#/d360pfy>. En fait, ce que je fais dans ce script-là, c'est lire les informations concernant une chanson en provenance du MPD (Music Player Daemon) pour ensuite, en utilisant le nom de l'artiste et de l'album, obtenir une image à partir de albumart.org. Si je ne réussis pas à convaincre le script d'aller la chercher automatiquement, je peux la téléchar-

ger et la sauvegarder dans mon dossier `.covers` avec les bons noms et elle sera utilisée. Quant à l'affichage dans Conky, je demande au script de sauvegarder un exemplaire de la chanson en cours comme `/tmp/cover` et puis d'afficher cette image-là (mise à jour toutes les 15 secondes). Ceci est contenu partiellement dans un argument « if » pour éviter un affichage vide lorsque MPD ne fonctionne pas. J'ai rassemblé un fichier `.conkyrc` de base pour le script ici : <http://lswestfcm.pastebin.com/rrCS0hDt> et le script même se trouve ici : <http://lswestfcm.pastebin.com/iX7Y7W3v>.

Et maintenant, quelques explications. La toute première ligne du fichier `.conkyrc` (« `imlib_cache_size 0` ») dit à Conky de ne pas mettre les images en cache (c'est nécessaire puisque vous les voulez mises à jour). Les autres lignes sont toutes plus ou moins les mêmes que le mois dernier, à l'exception de la partie MPD qui indique tout simplement à Conky l'emplacement de mon daemon. Ceci n'est nécessaire que si vous utilisez MPD pour votre musique. Si vous utilisez autre chose, il faudrait ajuster le script Python pour qu'il lise les informations album et artiste à partir



du système que vous utilisez et remplacer tous les appels spécifiques à MPD dans le fichier `conkyrc`. Les autres sections devraient être plus ou moins explicites, mais voici ce qui se passe après l'en-tête « TEXT » :

Je l'oblige à se servir d'UTF-8 pour qu'il puisse afficher tous les caractères spéciaux qui se présentent.

J'écris mon en-tête « MUSIC ».

J'affiche l'état.

Début de l'argument « If ».

- Exécuter le script `python` toutes les 2 secondes.
 - Afficher `/tmp/cover` à une position (100,330) (format des coordonnées (x,y)) avec une taille de 50px par 50px, mise à jour toutes les 15 secondes.
 - Afficher l'artiste.
 - Afficher le titre.
 - Afficher une barre avec le temps passé et le temps total aux deux bouts.
- Fin de l'argument « If ».**

Je me rends compte que cet article est court et, peut-être, pas

Je l'oblige à se servir d'UTF-8 pour qu'il puisse afficher tous les caractères spéciaux qui se présentent.

trop pertinent (je ne suis pas certain du nombre d'utilisateurs d'Ubuntu qui se servent de MPD), mais je le trouve extrêmement utile et un bon exercice pour quiconque voudrait en apprendre davantage sur Conky. En outre, si vous n'avez pas encore essayé MPD, je le recommande vivement. Puisque c'est un daemon, il sauvegarde son état quand vous arrêtez l'ordinateur, ce qui fait que vous pouvez reprendre votre musique tout de suite après un redémarrage ou continuer une chanson là où vous l'avez arrêtée. Il a pas mal d'interfaces et j'en ai présenté quelques-unes dans un article précédent (à la page 25 du FCM n° 32). Pour un petit exemple d'autres choses que vous pouvez faire avec Conky, voici un lien : <https://bbs.archlinux.org/viewtopic.php?pid=875306#p875306>. C'est un message des forums Arch linux affichant un Conky qui exécute des éléments d'interface graphique à partir de scripts « lua ». Je pense que la prise en charge de lua par Conky est activée par défaut aujourd'hui,

mais il se peut qu'elle soit désactivée dans la version des dépôts Ubuntu, alors il faudrait y penser si vous décidez d'essayer les configurations.

Comme toujours, j'espère qu'au moins quelques-uns d'entre vous ont trouvé cet article utile et j'aimerais avoir votre avis sur les interfaces MPD, les usages de Conky, ainsi que toutes remarques ou toutes idées de sujet d'article que vous pourriez avoir. Vous pouvez m'écrire à lswest34@gmail.com en pensant à mettre « C&C » ou « Full Circle Magazine » dans le sujet, pour que je ne rate pas votre message. Et puis, j'aimerais vraiment mieux si vos courriels étaient en anglais ou en allemand - j'en ai reçu un en français l'autre jour et j'avais beaucoup de mal à le comprendre. Dans ce message, le lecteur était en train de souligner l'existence de « file » - un programme qui affiche des informations concernant le type de fichier (c'est-à-dire JPEG, MP3, etc.) [Ndt : N'oubliez pas que l'équipe de traduction propose de traduire en anglais vos questions et courriels en français. Il suffit pour cela de les envoyer à FCMfr@free.fr. Pour plus de renseignements voir « Écrire pour le FCM », page 12.]



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à : lswest34@gmail.com.



Le Podcast Full Circle

Dans l'épisode n° 15 : Brainstorms, FUD et Media Players

- Critique : FCM n° 44.
- Actus : Idées pour les brainstorms, les indices de téléchargement de la logithèque Ubuntu, Fuduntu, Unity, Android, et plus encore !
- Jeux : Humble Indie Bundle 2, Mass Effect, FreeCiv et Dropbox.

Taille des fichiers :
OGG - 46.9 Mb
mp3 - 40.4 Mb

Durée : 1 h 24 min 34 sec
Sortie : 13 Janvier 2011

<http://fullcirclemagazine.org/>



Dans ce numéro j'aimerais traiter deux sujets : la création d'une liste de tâches en utilisant Conky et Bash et une présentation rapide de Zenity.

Ce sont deux demandes que j'ai reçues. Je vais vous montrer comment j'utilise Conky avec ma liste de tâches et je vous suggérerai quelques ajouts que l'on pourrait apporter. Cependant, je n'entrerai pas dans les détails au sujet de ces ajouts à moins que vous ne m'envoyiez plein de demandes.

Liste de tâches

Je fabrique mes listes de tâches tout simplement en créant un lien symbolique dans mon dossier Dropbox vers un répertoire nommé Rappels. Dans ce répertoire, j'ai plusieurs fichiers triés par thèmes (université, FCM, travail et personnel), puis dans Conky j'appelle un script Python (que j'ai écrit) qui parcourt tous ces fichiers et affiche chaque ligne précédée d'un « - ». Vous pouvez faire cela avec votre langage préféré. Mon script Python se trouve à droite (en haut) pour ceux que ça intéresse.

Assurez-vous de remplacer « Rappels » par le chemin depuis votre répertoire personnel pour que ça fonctionne (la variable « home » se charge de la partie « /home/\$USERNAME », ne précisez donc que ce qui suit. J'indique aussi d'ignorer les fichiers cachés de sauvegarde automatique (qui se terminent par un « ~ »). Je me rends compte que ma méthode est très basique, mais elle fonctionne. Si vous voulez ajouter des dates limites, ajoutez-les simplement dans vos fichiers lorsque vous saisissez les tâches. Cependant, si vous commencez à utiliser des dates, cela devient difficile de trier correctement les éléments. Pour cela, j'ai écrit deux scripts : le premier est à droite, l'autre est en haut de la page suivante.

Ces deux scripts fonctionnent très simplement. Le script `creerTaches.sh` prend tous les éléments d'un fichier (indiqué dans la variable `$fichier`), retire la date (au format mois jour hh:mm) et

```
#!/usr/bin/env python
import os

home=os.path.expanduser("~")

for racine, reps, fichiers in
os.walk(os.path.join(home,"Rappels")):
    for fic in [f for f in fichiers]:
        if(fic.endswith("~")!=True):
            fh=open(os.path.abspath(os.path.join(racine,fic)))
            for ligne in fh:
                print("- "+ligne, end=' ')
            fh.close()
```

creerTaches.sh :

```
#!/bin/bash
fichier=~ /Taches.txt
taches=~ /taches.txt
if [[ -e $taches ]]; then
    `rm "$taches"`
fi
while read ligne; do
    date=`date -d "$(echo "$ligne" | sed 's/\(.\) -.*$/\1/g')" +%s`;
    echo "$(echo "$ligne" | sed -e s/" .*-"/"$date -"/g)" >> "$taches";
done < "$fichier"

if [[ -e "$taches" ]]; then
    temp=`sort -n "$taches"`
    echo "$temp" > "$taches"
fi
```

la remplace par le temps Unix (nombre de secondes depuis le 1er janvier 1970), puis l'écrit dans un fichier (qui sera effacé à chaque lancement du script pour éviter les doublons). Une fois cela fait, il trie le contenu de ce fichier du plus pe-

tit nombre au plus grand (de la date la plus proche à la plus éloignée, c'est-à-dire la prochaine tâche en premier).

Le script `afficherTaches.sh` prend simplement chaque ligne du fichier

afficherTaches.sh :

```
#!/bin/bash
taches=~/.taches.txt
while read ligne; do
    if [[ "$ligne" != "" ]]; then
        date=`date -d@"$(echo "$ligne"|sed -e s/"-[^-]*$"/g)" +%a %b %d %H:%M`
        echo "$(echo "$ligne"|sed -e s/".*-"/"$date -"/g)";
    fi
done < "$taches"
```

taches.txt nouvellement créé, remplace la date Unix par une date normale et l'affiche. Le premier script pourrait devenir une tâche cron (lancée chaque heure ou toutes les deux ou trois minutes si vous souhaitez) et le deuxième pourrait être exécuté depuis Conky comme un script bash normal. Je n'ai pas trop testé ces scripts, mais ils font le boulot demandé. Je ne sais pas s'il existe une méthode plus simple, mais je pense que ces scripts vous permettent d'apprendre pas mal de choses.

Si vous améliorez ces scripts, j'aimerais que vous me le disiez et je serai heureux de pouvoir publier quelques solutions le mois prochain dans C&C.

Zenity

Pour ceux d'entre vous qui ne connaissent pas Zenity, c'est un outil en ligne de commande qui sert à créer des boîtes de dialogue (des éléments gra-

phiques). Comme je sais que de nombreux lecteurs n'osent pas trop utiliser la ligne de commande lorsqu'ils débutent, ceci pourrait être utile à ceux qui essaient d'aider un débutant. Je vais vous montrer en introduction les bases de Zenity et j'écrirai un tutoriel plus approfondi pour le mois prochain.

Zenity peut créer des fenêtres de saisie de texte ou de dates, des fenêtres d'information, des barres de progression, des icônes de notification, des listes déroulantes, des fenêtres de sauvegarde, des cases à cocher, des fenêtres d'erreurs, etc. Voici quelques exemples :

```
<command>| tee > (zenity
--progress --pulsate) >file
```

Cette commande affiche une barre de progression pendant que « tee » lit l'entrée standard STDIN, puis la sauvegarde dans un fichier. Dans ce cas, vous devez envoyer la sortie d'une commande vers tee.

```
zenity
--question
--test
"Question ?";
echo $?
```

Cette commande crée une fenêtre affichant une question et des boutons OK/Annuler.

« echo \$? » renvoie 0 si l'utilisateur appuie sur OK et 1 si l'utilisateur appuie sur Annuler (pratique pour une intervention de l'utilisateur). C'est exactement la même syntaxe pour les avertissements, en remplaçant `--question` par `--warning`.

```
<command>|zenity --text-info
--width <taille en pixels>
```

Cette commande récupère la sortie de la commande du début de ligne et l'affiche dans un champ de texte à l'intérieur d'une fenêtre.

```
saisie=$(zenity --entry
--text "Comment ça va ?"
--entry-text "Saisissez du
texte ici"); echo $saisie
```

Cette commande Zenity crée une fenêtre de saisie et renvoie la saisie dans la variable « saisie » (d'où l'affichage avec « echo »).

```
zenity --error --text "Une erreur
est survenue..."
```

Cette commande crée une fenêtre d'erreur et affiche le texte dans la fenêtre.

Il existe bien d'autres commandes fournies par Zenity, mais ceci devrait préparer tout codeur enthousiaste à l'écriture de la plupart des scripts dont il aurait besoin. Le mois prochain, je prévois d'utiliser quelques-unes de ces commandes pour écrire un script utile. Si quelqu'un veut formuler une demande sur ce que le script devrait faire, envoyez-moi un courriel à : lswest34@gmail.com. S'il vous plaît, indiquez FCM ou C&C comme « Objet » de vos messages, pour que je ne les rate pas ! Tous les commentaires sur cet article ainsi que des demandes générales sont les bienvenus.



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à : lswest34@gmail.com.



Je me rends compte que le mois dernier j'ai dit que j'allais faire un article sur l'utilisation de Zenity dans un script. Cependant, je n'ai pas trouvé de script qui pourrait bénéficier de Zenity, à moins d'être extrêmement compliqué. Au lieu de cela, j'ai décidé que je voulais aller plus loin dans Conky, notamment sa capacité d'utiliser des scripts LUA pour dessiner des éléments graphiques sur le bureau (j'ai choisi des anneaux, mais je suis sûr que d'autres objets sont possibles). Avant de commencer, je veux vous avertir que je ne suis nullement un programmeur LUA et il y a de bonnes chances qu'il existe des moyens plus faciles de faire les changements que j'ai faits, mais je vous montre ce que j'ai trouvé.

Pour ceux d'entre vous qui ne savent pas de quoi je parle, cette capture d'écran de mon profil DeviantArt est un exemple de ce qu'on peut faire avec Conky : <http://lswest.deviantart.com/#/d3ay5fb>

Tout d'abord, assurez-vous que vous avez installé Conky (la version 1.7.2 est dans les dépôts depuis la version 9.10 Karmic Koala). Je suppose que

tout le monde utilise la version 9.10 ou supérieure. Si vous utilisez une version plus ancienne, Launchpad aura probablement un PPA pour vous.

Concernant les widgets que nous allons créer, je vais faire un widget simple pour le lecteur de musique MPD et une horloge (les mêmes widgets visibles dans la capture d'écran ci-dessus). Aussi, comme je n'éprouvais pas le besoin de recréer la roue (ou dans ce cas l'anneau), je vais utiliser le script suivant comme base de ce que nous allons faire maintenant : <http://londonali1010.deviantart.com/art/quot-Rings-quot-Meters-for-Conky-141961783>. Pour ceux d'entre vous qui aimeraient les scripts complets (pour la référence/correction), voyez la section Scripts à la fin de cet article.

Pré-codage

Avant de commencer sur le script lui-même, je vous demande de décider si vous le voulez dans deux instances distinctes de Conky (mon choix) ou dans une instance unique. La raison pour laquelle j'en utilise deux est tout simplement parce que j'ai deux autres instances de Conky sur mon bureau et fusionner les wid-

L'horloge

Configuration de l'horloge.

Pour chaque anneau voulu, il faudra configurer une entrée dans la table des paramètres qui ressemblera à ceci :

```
{
    name='time',
    arg='%I',
    max=12,
    bg_colour=0xffffffff,
    bg_alpha=0.1,
    fg_colour=0xffffffff,
    fg_alpha=0.4,
    x=165, y=170,
    radius=89,
    thickness=7,
    start_angle=0,
    end_angle=360
},
```

gets en un seul aurait donné lieu à des chevauchements entre les instances Conky. Si vous souhaitez en utiliser une seule, vous devrez augmenter la taille minimale et ajuster les valeurs x et y pour chaque widget afin de les placer dans la fenêtre de Conky. x et y se rapportent à la position relative de Conky. Par exemple, si Conky commence à la position (400, 200) ((x, y)), alors un widget placé en (100, 85) sera effectivement à (500, 285) sur votre moniteur. Gardez cela à l'esprit.

En outre, pour utiliser les scripts LUA, vous devez ajouter ce qui suit à votre .conkyrc :

```
lua_load
/home/lswest/conky_testing/rings-v1.2.lua
```

```
lua_draw_hook_pre ring_stats
```

... où la première ligne est, bien sûr, le chemin d'accès réel absolu au script lua et le nom en dessous est le nom de votre fonction principale (si vous écrivez `conky_ring_stats`, ou `ring_stats`, il trouvera la fonction quelle que soit la variation que vous

utilisez dans le script réel).

Le script a des commentaires pour clarifier les entrées, mais je vais vous les expliquer rapidement. Le nom est en fait le nom de la variable Conky (par exemple `#{time }`), args représente les arguments (par exemple `#{time %}`), et il est analysé par le script aux lignes 121 à 131 (sur pastebin), dans la fonction locale `setup_ring`. Il envoie simplement la commande à Conky (après l'avoir mise sous la forme `#{name args}`), récupère le résultat et l'analyse. Il est alors transformé en nombre, et les écarts de l'anneau (entrés dans la variable `max`) sont calculés (ainsi si vous indiquez 360 divisions, chaque division représentera exactement 1° de l'anneau, ou si vous en indiquez 12, alors chacune sera de $2\pi/\max$ (en radians)). Ce n'est pas important si vous ne comprenez pas, il faut garder à l'esprit que pour avoir 12 heures dans l'anneau, vous devez faire 12 divisions. Les 4 variables suivantes sont tout simplement les couleurs de fond et de premier plan et leurs niveaux alpha (transparence). Les variables `x` et `y` fabriquent votre vecteur de position, le rayon est la largeur de l'anneau, on a également l'épaisseur de la ligne ainsi que l'angle auquel le cercle commence, ou `start_angle` (0°), et l'angle où il se termine, ou

`end_angle` (360°), de sorte que nous obtenons un cercle complet.

Pour ceux d'entre vous qui connaissent le format de la commande `date`, vous savez que `%l` est le format de l'heure avec des 0 devant (donc 01, ..., 12). Le format d'une heure sans les 0 devant est `%l`, mais cela n'a pas d'importance pour cette horloge. J'ai aussi réussi à le faire fonctionner correctement avec `%H` (0, ..., 23). J'ai rendu les deux anneaux suivants plus petits et celui des secondes est 2 pixels plus mince. En fin de compte, vous devriez avoir quelque chose qui ressemble (après l'anneau des heures) au code affiché à droite.

Comme vous pouvez le voir, c'est assez simple. Si vous réussissez à faire compter les secondes au centre et sans la date au centre, vous avez terminé. Si vous voulez, comme moi, que les secondes soient sur le cercle intérieur, alors vous devrez ajouter la ligne suivante avant « `cairo_arc(cr, xc, yc, ring_r, t_arc-arc_w, t_arc+arc_w)` » :

```
if pt['arg'] == '%S' then
  cairo_arc(cr, xc, yc, ring_r,
  angle_0, t_arc+arc_w) end
```

Nous commençons simplement à l'angle_0 (12 heures sur l'anneau) et prolongeons la ligne. Ma première réaction est de mettre la ligne d'origine dans une déclaration `else`, mais

ça fonctionne sans cela et cela fait moins de choses à saisir, donc nous allons renoncer à un bon formatage dans ce cas. Si vous souhaitez placer la date dans le centre de l'anneau, c'est un travail de conjectures pour le positionnement, mais voici ce que vous devez ajouter à votre `.conkyrc` :

```
#{goto 115}#{voffset
150}#{time %A}
```

```
#{goto 115}#{time %b %d %Y}
```

La ligne `goto` le déplace vers la droite (vous pouvez aussi utiliser `#{offset <pixels>}`) et `voffset` est le décalage vertical (c'est-à-dire le nombre de pixels dont on descend depuis le haut de la fenêtre conky). J'ai affiché le jour sur la ligne supérieure (`#{time %A}`) et la date sur la ligne en dessous. Si vous voulez changer la façon dont ça s'affiche, regardez la page de manuel de la commande `date` pour trouver les options de formatage dont vous avez besoin.

Widget MPD

D'abord, avant de commencer, il faut savoir que le widget que je décris ici ne fonctionne que pour MPD (Music Player Daemon), car Conky manque de variables pour les autres lecteurs de musique. Je suis sûr que vous pourriez le faire fonctionner de

```
{
  name='time',
  arg='%M',
  max=60,
  bg_colour=0xffffffff,
  bg_alpha=0.1,
  fg_colour=0xffffffff,
  fg_alpha=0.4,
  x=165, y=170,
  radius=79,
  thickness=7,
  start_angle=0,
  end_angle=360
},
{
  name='time',
  arg='%S',
  max=60,
  bg_colour=0xffffffff,
  bg_alpha=0.1,
  fg_colour=0xffffffff,
  fg_alpha=0.4,
  x=165, y=170,
  radius=70,
  thickness=5,
  start_angle=0,
  end_angle=360
},
```

la même façon avec des bouts de codes compliqués, mais je ne pense pas que ça vaille le coup car la plupart des autres lecteurs de musique ont un widget du type « lecture en cours ».

L'entrée `settings_table` pour celui-ci ressemble au code qu'on voit en haut à gauche de la page suivante.

Comme vous pouvez le voir, nous travaillons avec 100 divisions (puis-

```
{
  name='mpd_percent',
  arg='',
  max=100,
  bg_colour=0xffffffff,
  bg_alpha=0.1,
  fg_colour=0xffffffff,
  fg_alpha=0.4,
  x=70, y=170,
  radius=60,
  thickness=7,
  start_angle=0,
  end_angle=360
},
```

que c'est un pourcentage, ça sera une valeur comprise entre 0 et 100). En outre, la variable `arg` est vide, ce qui est important car si on la laisse complètement de côté c'est incompatible avec les fonctions que nous utilisons plus tard (argument manquant). Une fois que vous avez fait cela, j'ai aussi modifié le script pour que le widget disparaisse quand la musique est en pause. Pour ce faire, vous devez apporter les modifications suivantes au script :

Ajoutez cette fonction au début ou à la fin du fichier :

```
function
conky_my_flag(my_arg)
  flag = my_arg
  return ""
end
```

Placez ensuite le texte suivant du script original :

```
local
updates=conky_parse('${updates}')

update_num=tonumber(updates)

  if update_num>5 then
    for i in
pairs(settings_table) do

      setup_rings(cr,settings_table[i])
    end
  end
end
```

dans l'instruction `if` suivante :

```
if tonumber(flag) == 1 then
<text from above>
end
cairo_destroy(cr)
```

de façon à ce que les 11 dernières lignes du fichier ressemblent à ce qui est ci-dessus à droite.

Les changements ci-dessus ne font que détruire le widget si MPD ne fonctionne pas et, sinon, le font fonctionner normalement. La fonction que nous avons créée permet d'assigner une valeur au drapeau global que nous utilisons dans l'instruction `if`. Maintenant, avant que ce script fonctionne, vous devrez ajouter `#{lua my_flag 0}` et `#{lua my_flag 1}` à votre `.conkyrc` de sorte que la fonction soit appelée et que le drapeau soit mis à 0 ou 1 selon si MPD est arrêté (0) ou pas (1). La section `TEXT` de mon `.conkyrc` ressemble au code ci-dessus.

```
if tonumber(flag) == 1 then
  local updates=conky_parse('${updates}')
  update_num=tonumber(updates)

  if update_num>5 then
    for i in pairs(settings_table) do
      setup_rings(cr,settings_table[i])
    end
  end
end
cairo_destroy(cr)
end
```

```
#{lua my_flag 0}
#{if_mpd_playing}
#{lua my_flag 1}
#{execi 2 python /usr/bin/mpd-cover}
#{image /tmp/cover -p 40,138 -s 60x60 -u 15}
#{if_match "${mpd_status}" == "Paused"}
#{offset 137}${voffset 40}${font
DejaVuSans:bold:size=10}Paused
$endif
#{if_match "${mpd_status}" == "Playing"}
#{offset 137}${voffset 20}${font
DejaVuSans:bold:size=10}${mpd_artist}
#{offset 137}${font DejaVuSans:size=9}${scroll 38
${mpd_title}}$font
$endif
$endif
```

Elle permet de définir le drapeau à 0 lorsque `if_mpd_playing` est faux. Sinon, il est mis à 1. Le reste des paramètres gère l'affichage et la position de la pochette de l'album, affiche « pause » si MPD est en pause, ou l'artiste et le titre de la chanson sur deux lignes à la droite de l'anneau si MPD fonctionne. La section `#{scroll`

`38 ${mpd_title}}` fait que le titre défile (le texte se déplace de droite à gauche) s'il fait plus de 38 pixels. Vous pouvez omettre cela, mais je l'ai mis là pour prévoir le cas où le texte est plus long que la largeur de mon Conky. Pour afficher l'image, vous aurez besoin d'ajouter les deux paramètres suivants au-dessus du mar-

queur TEXT quelque part :

`imlib_cache_size 0`

De plus, le script mpd-couverture est ci-dessous, dans la section Scripts. Le script mpd-couverture est écrit pour Python 2.X, mais vous pouvez toujours utiliser le programme 2to3 pour le ré-écrire pour Python 3. Si vous avez des problèmes, faites-le moi savoir. Soyez conscient que certains symboles peuvent causer des problèmes avec le script. J'ai fait très peu (voire pas) de changements, et il était à l'origine pris ici : <https://bbs.arch-linux.org/viewtopic.php?id=112708>.

J'espère que la majorité d'entre vous ont trouvé cela intéressant, et, comme toujours, je suis ouvert aux demandes, suggestions, commentaires d'ordre général et autres questions. Vous pouvez me joindre à l'adresse lswest34@gmail.com, et n'oubliez pas de mettre C&C ou FCM comme « Objet » pour ne pas que je rate votre message. En outre, l'anglais ou l'allemand sont mes langues préférées, parce que sinon je vais devoir compter sur Google Translate. Si quelqu'un améliore les scripts que j'ai énumérés/utilisés ici, n'hésitez pas à envoyer une copie avec une explication des ajouts/modifications, et je le noterai au début du prochain article pour ceux qui seraient intéressés.

Scripts:

<http://pastebin.com/SpC6bcn7>
horloge en anneau Lua ;
<http://pastebin.com/iZFdZAeg> Conky
mpd ;
<http://pastebin.com/zkVVHkYk>
.conkyl_mpd ;
<http://pastebin.com/BDa5MHuR>
conkyrc pour l'horloge ;

<http://pastebin.com/ZX4pLbta> script
mpd-couverture.



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à : lswest34@gmail.com.

u³ UCubed
<http://ucubed.info>

A free event taking place in Manchester on the 2nd April 2011 that brings together the Debian and Ubuntu communities

Visit our website <http://ucubed.info> for tickets and for more information

Do you want to tell people about your new project? Do you want to work with others to develop your ideas?
Do you want to learn more about Linux and Free Software?
Do you want to contribute to free software?

Fun
Ubuntu Debian Crunchbang
Network Participate
Friends Learn
Community Documentation
Contribute Development
Co-working Artwork
Share

lebian <http://ucubed.info> ubuntu <http://ucubed.info> #!



Ce mois-ci, j'ai souhaité partager avec vous quelque chose que j'ai appris récemment. Les sujets que je vais couvrir ne s'appliquent qu'aux lecteurs qui soit utilisent iBus/SCIM et ne sont pas heureux avec, soit ceux qui l'utilisent et sont heureux avec, mais dont les caractères japonais/chinois/etc., n'apparaissent pas dans une police lisible en rxvt-unicode. Je voudrais également prendre un moment pour annoncer que le mois prochain j'espère faire une séance de questions et réponses pour les lecteurs de C&C. Si vous avez des questions concernant Linux en général, la ligne de commande, ou moi en tant qu'auteur, n'hésitez pas à envoyer vos questions avant le 28 avril à lswest34@gmail.com. Je choisirai un tas de questions pour y répondre le mois prochain. Les demandes d'articles sont également les bienvenues.

Comme certains d'entre vous le savent probablement, j'ai écrit un article sur iBus dans le numéro 43 du FCM. Je n'avais pas utilisé iBus, puisque j'étais à l'aise avec SCIM. Toutefois, une mise à jour a récemment désactivé SCIM et j'ai donc essayé

iBus. Ce qui m'a vraiment embêté, c'est que je ne pouvais pas passer facilement de hiragana à katakana, alors j'ai décidé de suivre la suggestion d'un de mes amis et j'ai essayé uim. Étonnamment, uim ne bloque pas mes touches mortes en rxvt-unicode et permet de passer facilement de hiragana à katakana. Ci-dessous, voici comment je l'ai configuré pour l'utiliser.

uim & uim-fep :

On lit sur la page d'accueil (<http://code.google.com/p/uim/>):

« L'objectif de uim est de fournir une plate-forme pour développer une méthode de saisie simple, facilement extensible et avec un code de haute qualité, et un environnement avec une méthode de saisie utile pour les utilisateurs d'ordinateurs de bureau et de plate-formes embarquées. Voyez Qu'est-ce qu'uim ? pour plus d'informations. »

D'abord, vous aurez besoin de l'installer :

```
sudo apt-get install uim uim-gtk2.0 uim-qt uim-qt3 uim-fep uim-anthy
```

Cela devrait fournir le soutien uim pour les terminaux, les applications QT et les applications GTK utilisant anthy. Il y a de nombreux autres paquets offrant des applets, des dictionnaires divers et des utilitaires qui peuvent intéresser certaines personnes.

Une fois que vous l'avez installé, exécutez uim-toolbar-gtk-systray et vous obtiendrez une icône dans la barre des tâches. Faites un clic droit dessus et choisissez Préférences. Modifiez la liste des langues de saisie pour laisser seulement celles dont vous avez besoin et ajustez les raccourcis clavier selon vos préférences. Si vous constatez que l'icône dans la barre des tâches est pratiquement invisible, c'est parce que trop d'information est affichée dans la largeur d'une icône. Pour régler ce problème, ouvrez les Préférences et décochez tout dans la rubrique « barre d'outils », puis réglez simplement sur « Mode de saisie » les boutons de la barre d'outils pour chaque langue que vous utilisez. Ceci les réduira à une seule icône, ce qui la rend lisible à nouveau. Afin de le faire fonctionner, vous aurez également besoin d'ajouter ce qui

suit au fichier /etc/profile (ou .bashrc ou .zshrc) :

```
export XMODIFIERS=@im=uim
export GTK_IM_MODULE="uim"
export QT_IM_MODULE="uim"
```

Une fois que vous avez défini ces variables, vous devrez exécuter la commande suivante dans un terminal :

```
gtk-query-immodules-2.0 > /etc/gtk-2.0/gtk.immodules
```

Cela recréera le fichier gtk.immodules, qui précise aux programmes GTK les types de méthodes de saisie disponibles.

Uim-fep est un processeur Front-End pour les émulateurs de terminaux. Fondamentalement, il vous permet de saisir en japonais dans un émulateur de terminal (rxvt-unicode dans mon cas), sans compter sur uim-xim (qui a tendance à être un gros mangeur de ressources). Pour le faire fonctionner, vous aurez besoin d'ajouter uim-fep à la fin de votre fichier .bashrc ou .zshrc ou du shell que vous

utilisez. Si vous recevez un avertissement indiquant que uim-fep est déjà en cours d'exécution, vous pouvez ajouter « clear » à sa suite, de sorte qu'il cache le message. Une fois qu'il est lancé, vous aurez une ligne à la fin de votre terminal qui ressemble à ceci :

```
anthy-utf8 [- ]
```

L'utilisation du raccourci global pour uim changera l'icône à la fin suivant la méthode d'entrée et vous permettra de saisir du japonais en ligne dans le terminal.

Rxvt-unicode :

Si vous avez le problème que votre japonais est presque illisible dans rxvt-unicode (cela peut s'appliquer aux autres émulateurs de terminaux aussi, mais je ne l'ai pas testé), vous pouvez ajouter ce qui suit à votre .Xdefaults :

```
URxvt.preeditType: On-TheSpot, None
```

```
URxvt.imLocale: ja_JP.UTF-8
```

```
URxvt.font: xft:AnonymousPro:size=11:antialias=true:autohint=false,xft:IPAGo-
```

```
thic:size=11:antialias=true
```

```
URxvt.boldFont: xft:AnonymousPro:size=11:weight=Bold:antialias=true:autohint=false,xft:IPAGothic:size=11:weight=Bold:antialias=true
```

Ceci dit simplement à urxvt de s'attendre à des entrées en japonais depuis uim. Les polices sont en fait une liste de deux, comme vous pouvez le voir. Anonymous Pro est la police terminal que j'utilise pour tout, mais si rxvt-unicode ne trouve pas les symboles pour quelque chose dans cette police, il va passer à la suivante dans la liste (ou une police de repli si ce symbole n'existe pas dans les polices listées). Cela vous permet d'avoir le support de plusieurs langues sans compromettre la lisibilité de l'alphabet latin. En outre, vous pouvez voir des gens utilisant urxvt.* au lieu de URxvt.* - mais cela peut être problématique si vous définissez le nom de votre terminal à partir d'un raccourci (par exemple avec urxvt -name ncmpcpp -e ncmpcpp). La première partie de ces préférences indique au système que nous voulons agir sur la WM_CLASS du programme et que le « urxvt » en minuscules est le premier de la liste, ce qui est défini en utilisant l'argument -name. Si, au contraire, vous utilisez « URxvt », il ne changera pas en fonction de l'option -name. Pour

voir ce que je veux dire, saisissez la commande suivante dans un terminal, et cliquez sur rxvt-unicode.

```
xprop | grep "^WM_CLASS"
```

qui vous donne quelque chose comme ceci :

```
WM_CLASS(STRING) = "urxvt", "URxvt"
```

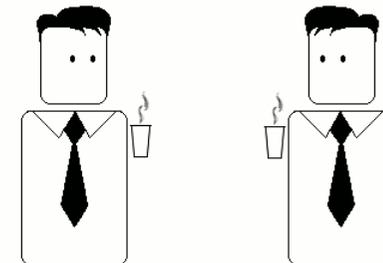
Maintenant vous devriez avoir une installation uim entièrement fonctionnelle et ne devriez pas avoir perdu de fonctionnalités dans votre terminal. Si vous avez des suggestions ou demandes d'articles, n'hésitez pas à m'envoyer un courriel à lswest34@gmail.com. Et n'oubliez pas vos questions ! J'ai besoin de les recevoir avant le 28 avril !



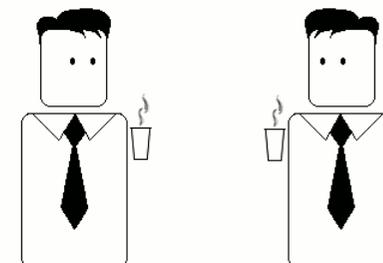
Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez contacter Lucas par courriel à : lswest34@gmail.com.

Un nouveau boulot

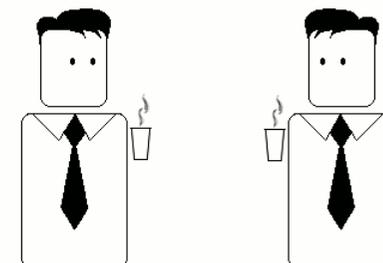
J'ai de sérieuses doutes concernant mes capacités tu sais. Je n'ai presque jamais de bonnes idées.



Cela veut-il dire que je ne deviendrai jamais ingénieur ?



Non, bien sûr que non. Au contraire tu seras parfaitement à ta place.



by Richard Redei



Le 22 avril, un lecteur, John Niendorf, m'a contacté pour demander un article sur Graphicsmagick. En réponse à son courriel, j'ai prévu de vous présenter des cas d'utilisation basique de graphicsmagick (que l'on appellera gm ici), un usage plus avancé (gestion automatisée) et la création de fichiers MIFF contenant un aperçu de toutes les images d'un répertoire.

Si vos besoins ressemblent aux miens, vous trouverez que gm est excellent pour faire de la conversion automatisée ou bien si vous souhaitez créer rapidement une miniature à partir d'une grande image sans ouvrir une application graphique. Avant de lire la liste, sachez que `<quelque chose>*` signifie que le quelque chose peut être répété indéfiniment et que ce qui se trouve entre crochets est optionnel (mais utile à connaître). Alors, sans autre préliminaire, voici une liste des commandes que je trouve utiles ; gardez en tête que ce sont des exemples de base :

gm identify <fichier>

Liste des informations sur le format et la taille de l'image ; affiche éga-

lement le statut du fichier (incomplet, corrompu, etc.).

gm montage [<options><entrée>]* <sortie>

Combine tous les fichiers d'entrée dans un seul fichier de sortie, avec des options de mise en forme (mosaïque, afficher le nom des images en dessous, etc.).

gm mogrify <options> <fichier d'entrée>

Transforme le fichier.

gm convert <options> <fichier d'entrée> <fichier de sortie>

Transforme le fichier (idem que mogrify, sauf que mogrify écrase le fichier).

gm composite <fichier à changer> <fichier de base> [<fichier de masque>] <fichier de sortie>

Fusionne, mélange et masque les fichiers pour créer une nouvelle image.

Ce n'était qu'un résumé très simple des commandes possibles et un exemple de base des arguments que l'on peut utiliser avec. Quant aux options courantes :

-geometry <hauteur>x<largeur>

<+/-><x><+/-><y>

Cette option spécifie la géométrie de la fenêtre de l'image, incluant un décalage x et y. Par exemple : `-geometry 1600x1050+10+10`

-size <hauteur>x<largeur>

Cette option est placée avant le fichier d'entrée et permet de lire les images jpeg dans une petite taille, dans le but de raccourcir le temps de traitement. Très pratique pour la création automatisée de miniatures. Exemple : `-size 170x160`.

-thumbnail <hauteur>x<largeur>

Cette option utilise des options pré-réglées pour créer rapidement une miniature.

-resize <hauteur>x<largeur>

Cette option change l'échelle de l'image pour qu'elle ait la taille demandée.

-gaussian <rayon>{<x><sigma>}

Cette option applique un flou gaussien à l'image. Sigma est l'écart-type. En général, vous n'aurez besoin que de l'option pour le rayon.

-quality <valeur>

Cette option règle la qualité de l'image de sortie (pour les JPG/MIFF/TIFF/PNG).

[Graphicsmagick] est excellent pour faire de la conversion automatisée...

`<valeur>` peut être un entier entre 0 et 100 (où 100 = qualité la meilleure et faible compression).

-crop <largeur>x<hauteur>{<+><x>{<-><y>{<%>

Cette option vous permet de découper l'image à la taille que vous indiquez (et de fournir un décalage).

Cette liste d'options devrait vous suffire pour commencer et expérimenter. Une fois que vous aurez trouvé une commande que vous aimez, avec les bonnes options, vous voudrez peut-être l'appliquer à plusieurs fichiers du répertoire courant. Pour faire cela, vous procéderez ainsi (voyez plus bas pour l'explication des options) :

```
find . -name "*.jpg" | xargs -l -i
basename "{}" ".jpg" | xargs -l -i
gm convert -quality 100%
"{}.jpg" "{}.png"
```

COMMAND & CONQUER

Ici, find . -name "*jpg" renvoie à une liste de tous les fichiers jpg du répertoire courant ; cette liste est envoyée à xargs qui la parcourt ligne à ligne ("-l") et retire le suffixe (".jpg") de la liste ("}") en utilisant la commande basename. Ensuite, la liste est envoyée à nouveau à xargs qui exécute gm convert -quality 100% "{}.jpg" "{}.png", ce qui en gros prend chaque image et la convertit en fichier png. L'étape intermédiaire est nécessaire pour éviter d'avoir des fichiers nommés "*.jpg.png" après la fin du travail. Ceci peut également être utilisé pour découper, éditer ou redimensionner un grand nombre de fichiers.

Pour terminer, je vais vous montrer comment créer un visuel d'un répertoire d'images (un fichier contenant l'aperçu de toutes les images d'un répertoire). Pour créer le fichier, utilisez cette commande :

```
gm convert 'vid:*.jpg'
directory.miff
```

L'extension miff signifie Magick Image File Format (format de fichier d'image magick). Le nom de ce format s'explique par le fait que gm est un logiciel qui fut dérivé d'ImageMagick en 2002. Pour afficher ce fichier par la suite, exécutez simplement la commande :

```
gm display directory.miff
```

Si vous vous demandez en quoi ceci peut être utile, imaginez que vous ayez des milliers de photos sur un PC et que vous en cherchiez une en particulier. Au lieu de travailler sur l'ordinateur en essayant de la retrouver, vous pourriez copier le fichier miff et le parcourir pendant votre temps libre ou l'utiliser pour créer un catalogue d'aperçus.

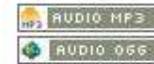
J'espère que vous avez trouvé les astuces de cet article intéressantes et que vous continuerez à en faire bon usage. Si vous avez des demandes ou des questions, vous pouvez me joindre à l'adresse lswest34@gmail.com. Placez les mots « Command and Conquer », « C&C », « Full Circle Magazine » ou « FCM » dans l'espace « Objet », pour que je voie votre message. Et s'il vous plaît, essayez d'écrire les messages en anglais ou en allemand, sinon je dois compter sur Google Traduction.



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à : lswest34@gmail.com.



Full Circle
Podcast



Vos animateurs :

Robin Catling

Ed Hewitt

Dave Wilkins

Audio: Victoria Pritchard

Notez :

00:42	BIENVENUE et INTRO
01:04	DEPUIS LA DERNIÈRE FOIS...
04:55	CRITIQUE du n° 48 du magazine Full Circle
12:05	CRITIQUE : Ubuntu 11.04
57:26	CONTRIBUER
1.26:23	REMARQUES DES AUDITEURS
1.27:19	CONCLUSION



Avant de commencer cet article, j'aimerais partager deux des façons dont John Niendorf utilise gm (avec mes remerciements !). Les voici :

```
alias imgresize='gm mogrify
-resize 640x480 *.jpg *.JPG'
```

```
alias frameall='gm mogrify
-mattecolor yellow -frame
5x5+0+5 *.JPG *.jpg *.jpeg
*.png'
```

Vous pouvez coller ces alias soit dans votre fichier .bashrc soit dans un fichier dédié aux alias. La commande du dessus redimensionne tous les fichiers jpeg à 640×480 et l'autre rajoute un cadre autour de tous les fichiers jpeg et png.

Et maintenant passons à l'article de ce mois. Comme certains d'entre vous le savent peut-être, la saisie des formules mathématiques (pendant des cours magistraux ou en séminaire ou pour n'importe quelle raison) à l'intérieur de programmes comme OpenOffice ou Libreoffice n'est pas la chose la plus facile au monde. Surtout lorsque vous abordez la théorie des ensembles ou d'autres concepts mathématiques avancés avec des lettres grecques, des symboles comme « pour tout » et ainsi de suite. Pour ce genre de

choses, je recommande vivement l'utilisation de LaTeX, car il sait bien formater les formules mathématiques (cf. le numéro 11 pour une présentation de base de LaTeX). Dans cet article, je vais vous présenter quelques paquets mathématiques et quelques trucs et astuces utiles pour bien formater des formules mathématiques. Quant aux logiciels, j'aime bien Texmaker ; les paquets texlive proposés dans les dépôts officiels Ubuntu devraient comprendre tous les paquets mentionnés ici.

Préambule du document

Le préambule c'est, en LaTeX, tout le texte saisi avant la ligne `\begin{document}`. Cela comprend les paramètres du document, les en-têtes, les pieds de pages, l'import de paquets et les paramètres des paquets. Les documents mathématiques typiques que je crée contiennent les paquets suivants :

- tikz** (pour les diagrammes/graphiques, pour lequel je charge la bibliothèque `decorations.markings tikz`);
- amsmath** – propose des améliorations pour toutes les fonctions mathématiques de base ;
- amsfonts** – propose des formatages spéciaux pour les maths (calligraphie mathématique (`\mathcal{}`), bloc de texte mathé-

matique (`\mathbb{}`), etc.) ;

- amssymb** – propose la capacité d'afficher des équations numérotées, des maths en ligne, etc.
- hyperref** (lorsque vous utilisez une table des matières) – permet la création de liens cliquables dans les documents TeX.

Vous trouverez ci-dessous le véritable préambule que j'utilise pour mes notes d'algèbre linéaire (la section document contient seulement deux instructions `include` - et les références à `utf8` et `ngerman` existent parce que mon cours est en allemand).

```
\documentclass[12pt,a4paper]{article}
% page counting, header/footer
\usepackage{fancyhdr}
\usepackage{lastpage}
\usepackage[ngerman]{babel}
\usepackage{tikz}
\usetikzlibrary{decorations.markings}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{graphicx}
\usepackage[utf8]{inputenc}
\usepackage{hyperref}
\addtolength{\oddsidemargin}{-.525in}
\addtolength{\evensidemargin}{-.525in}
\addtolength{\textwidth}{1.5in}
\hypersetup{unicode=true,pdfborder={0 0 0 [0 0]},
linkcolor=blue}
\title{Lineare Algebra}
\author{Lucas Westermann}
\pagestyle{fancy}
\fancyhead{}
\fancyfoot{}
\fancyhead[L,L]{Lineare Algebra}
\fancyhead[R,R]{Lucas Westermann}
\fancyfoot[R,R]{Seite \thepage\ von \pageref{LastPage}}
\fancyfoot[L,L]{\hyperlink{contents}{Inhaltsverzeichnis}}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
\setlength{\headheight}{16pt}
```

TikZ est sans doute le paquet le plus compliqué à utiliser et je vais donc le traiter en premier. Ce qui suit est le code utilisé pour créer le graphique A :

```
\begin{tikzpicture}[node
distance=2cm, auto]

\node (1) {\hat{1}};

\node (2) [right of = 1]
{\hat{2}};

\node (3) [below of = 2]
{\hat{3}};

\draw[decoration={markings,
mark=at position 1 with
{\arrow[ultra
thick]{>}},postaction={decorate}] (1) to node {1} (2);

\draw[decoration={markings,ma
rk=at position 1 with
{\arrow[ultra
thick]{>}},postaction={decorate}] (2) to node {3} (3);

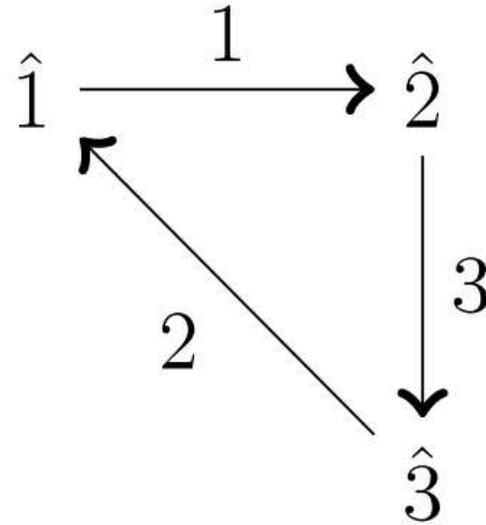
\draw[decoration={markings,ma
rk=at position 1 with
{\arrow[ultra
thick]{>}},postaction={decorate}] (3) to node {2} (1);

\end{tikzpicture}
```

Graphique A

Ce code-ci crée 3 noeuds (appelés 1, 2 et 3). L'information entre les accolades (« } ») est l'étiquette pour le noeud (ce qui est affiché), ainsi si on la laisse vierge nous avons comme résultat un noeud vide. Les trois lignes suivantes « /draw »

créent les lignes entre les noeuds (en se servant du nom des noeuds, qui se



trouve entre les crochets normaux), et sont étiquetés à nouveau par ce qui est entre les accolades.

En utilisant les paquets mathématiques pour créer et aligner des équations :

```
\begin{align*}
(\mathbb{K}_1^1) & \alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma \\
(\mathbb{K}_1^2) & \alpha + 0 = 0 + \alpha = \alpha \\
(\mathbb{K}_1^3) & \alpha \cdot -\alpha = -\alpha \cdot \alpha = 0 \\
(\mathbb{K}_1^4) & \alpha + \beta = \beta + \alpha
\end{align*}
```

```
\beta = \beta + \alpha
\end{align*}
```

Le texte qui en résulte se trouve ci-dessous.

L'environnement align* vous permet d'utiliser des caractères de type tabulation (« & ») pour aligner le texte joliment. C'est particulièrement utile quand vous faites une preuve et vous voulez aligner les équations aux signes égal. Le résultat du `\mathbb{}` est le K bloqué. Le « _ » et le « ^ » désignent l'indice et l'exposant. Si vous avez un ensemble indice/exposant plus long qu'un seul caractère, vous devrez le mettre entre des accolades. `\alpha`, `\beta` et `\gamma` désignent les lettres grecques. `\cdot` est un signe multiplier, les deux barres obliques inversées signifient des sauts de ligne et le reste est évident.

D'autres commandes utiles sont des

$$\begin{aligned}
 (\mathbb{K}_1^1) \alpha + (\beta + \gamma) &= (\alpha + \beta) + \gamma \\
 (\mathbb{K}_1^2) \alpha + 0 &= 0 + \alpha = \alpha \\
 (\mathbb{K}_1^3) \alpha \cdot -\alpha &= -\alpha \cdot \alpha = 0 \\
 (\mathbb{K}_1^4) \alpha + \beta &= \beta + \alpha
 \end{aligned}$$

trucs comme :

« `\forall` » (pour tout - le symbole A à l'envers);
 « `\exists` » (il existe - un E renversé);
 « `\in` » (un E courbé, utilisé lorsqu'on parle d'ensembles);
 « `\cup` » (symbole d'union dans la théorie des ensembles);
 « `\cap` » (symbole d'intersection dans la théorie des ensembles);
 « `\mathcal{}` » (transforme la lettre entre les accolades en cursive – utilisé par mon professeur quand il désigne un ensemble de vecteurs qui sont indépendants linéairement, c'est-à-dire, une base de vecteurs).

J'espère que vous aurez trouvé cet article utile. J'aurais pu traiter plus d'exemples, mais quel que soit leur nombre, cela serait adapté à seulement un petit nombre de cas d'usage. Ainsi, vous devriez penser à ceux-ci comme des exemples de ce que vous pouvez faire avec LaTeX. Voyez la section « Pour aller plus loin »

pour le lien vers un manuel utile. Si vous avez des questions ou des requêtes, envoyez-moi un mail en anglais à lswest34@gmail.com. Veuillez mettre dans le champ « Objet », « Full Circle Magazine », « FCM » ou « C&C » pour être sûr que je le vois.

Pour aller plus loin :

<http://fr.wikibooks.org/wiki/LaTeX>

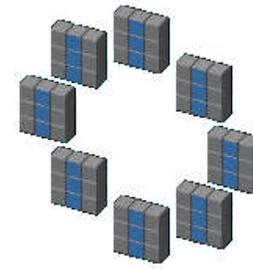
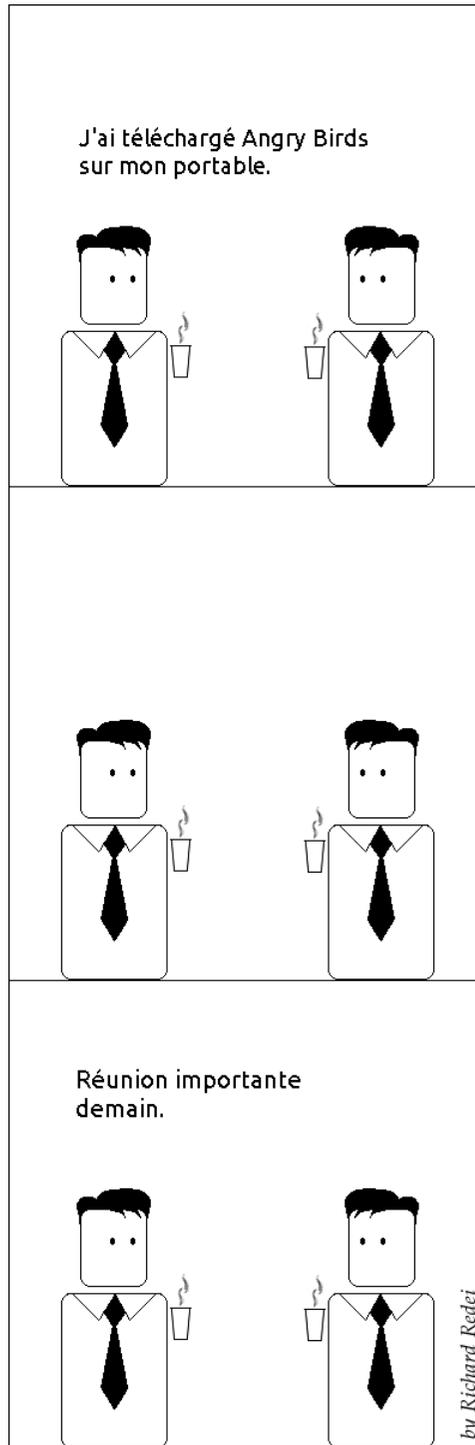
- C'est un Wikibook en français où sont traitées un grand nombre d'utilisations habituelles de LaTeX. Pour tous les autres usages, une recherche rapide sur Google devrait faire l'affaire.

<http://sourceforge.net/projects/pgf/>

- dans le fichier zip se trouve un manuel génial expliquant beaucoup d'usages du paquet TikZ.



Lucas a appris tout ce qu'il sait en endommageant régulièrement son système et en n'ayant alors plus d'autre choix que de trouver un moyen de le réparer. Vous pouvez lui écrire à : lswest34@gmail.com.



Serveur Circle

Server Circle est un nouveau site de questions et réponses géré par des experts en technologie.

Les utilisateurs de tout niveau peuvent poser gratuitement des questions techniques en rapport avec les serveurs. Ils recevront des réponses de la part d'experts de confiance notés par la communauté.

À la longue, vous pouvez gagner des points de réputation, et même des récompenses financières, en contribuant par vos réponses aux questions posées par d'autres personnes.



<http://www.servercircle.com>

Recent	Popular	Unanswered	Rewards	
Which packages can I safely uninstall on Ubuntu Server ?				17 views 0 replies
answer now				
by Squeeze (445 points) in Ubuntu Linux - 0 votes				
How to access a Intel Express 535T Switch Hub				34 views 4 replies
answer now				
by GoldAlchemist (50 points) in Networking - 1 votes				
How to use wired desktop for server to wireless notebook				38 views 2 replies
answer now				
by btinman (50 points) in Linux Servers - 1 votes				
Help with mod_security in Apache please				

N.B. Server Circle n'est ni affilié avec, ni approuvé par, le magazine Full Circle.



COMMENT CONTRIBUER

FULL CIRCLE A BESOIN DE VOUS !

Un magazine n'en est pas un sans articles et Full Circle n'échappe pas à cette règle. Nous avons besoin de vos opinions, de vos bureaux et de vos histoires. Nous avons aussi besoin de critiques (jeux, applications et matériels), de tutoriels (sur K/X/Ubuntu), de tout ce que vous pourriez vouloir communiquer aux autres utilisateurs de *buntu. Envoyez vos articles à :

articles@fullcirclemagazine.org

Nous sommes constamment à la recherche de nouveaux articles pour le Full Circle. Pour de l'aide et des conseils, veuillez consulter l'Official Full Circle Style Guide :

<http://url.fullcirclemagazine.org/75d471>

Envoyez vos **remarques** ou vos **expériences** sous Linux à : letters@fullcirclemagazine.org

Les tests de **matériels/logiciels** doivent être envoyés à : reviews@fullcirclemagazine.org

Envoyez vos **questions** pour la rubrique Q&R à : questions@fullcirclemagazine.org

et les **captures d'écran** pour « Mon bureau » à : misc@fullcirclemagazine.org

Si vous avez des questions, visitez notre forum : fullcirclemagazine.org

Merci de noter :

Les éditions spéciales sont un assemblage des numéros d'origine et les indications peuvent ne pas fonctionner dans les versions actuelles d'Ubuntu.

Équipe Full Circle

Rédacteur en chef - Ronnie Tucker

ronnie@fullcirclemagazine.org

Webmaster - Lucas Westermann

admin@fullcirclemagazine.org

Éditions spéciales - Jonathan Hoskin

Correction et Relecture

Mike Kennedy, Gord Campbell, Robert Orsino, Josh Hertel, Bert Jerred, Jim Dyer et Emily Gonyer

Remerciements à Canonical, aux nombreuses équipes de traduction dans le monde entier et à **Thorsten Wilms** pour le logo du FCM.

Pour la traduction française :

<http://www.fullcirclemag.fr>

Pour nous envoyer vos articles en français pour l'édition française :

webmaster@fullcirclemag.fr

Obtenir le Full Circle Magazine :

Pour les Actus hebdomadaires du Full Circle :



Vous pouvez vous tenir au courant des Actus hebdomadaires en utilisant le flux RSS : <http://fullcirclemagazine.org/feed/podcast>



Ou, si vous êtes souvent en déplacement, vous pouvez obtenir les Actus hebdomadaires sur Stitcher Radio (Android/iOS/web) :

<http://www.stitcher.com/s?fid=85347&refid=stpr>



et sur Tunein à : <http://tunein.com/radio/Full-Circle-Weekly-News-p855064/>

Obtenir le Full Circle en français :

<http://www.fullcirclemag.fr/?pages/Numéros>



Format EPUB - Les éditions récentes du Full Circle comportent un lien vers le fichier epub sur la page de téléchargements. Si vous avez des problèmes, vous pouvez envoyer un courriel à : mobile@fullcirclemagazine.org



Issuu - Vous avez la possibilité de lire le Full Circle en ligne via Issuu : <http://issuu.com/fullcirclemagazine>. N'hésitez surtout pas à partager et à noter le FCM, pour aider à le faire connaître ainsi qu' Ubuntu Linux.



Magzster - Vous pouvez aussi lire le Full Circle online via Magzster : <http://www.magzster.com/publishers/Full-Circle>. N'hésitez surtout pas à partager et à noter le FCM, pour aider à le faire connaître ainsi qu'Ubuntu Linux.