Ubuntu

ITRODUCTION 3
C'EST PARTI ! 3
LES CONDITIONS A REMPLIR 3
QUELLE VERSION D'UBUNTU ? 3
MISE A JOUR 3
UBUNTU, KUBUNTU OU ENCORE ? 4
À PROPOS DE L'AUTEUR 4
ISTALLATION DE LOGICIELS 5
GESTION DES PAQUETS : SYNAPTIC 5
COMMENT UTILISER APT ? 7
ERSONNALISER UBUNTU 10
MODIFIER LES REGLAGES DE L'ECRAN 10
MODIFIER L'APPARENCE DU BUREAU 11
TABLEAUX DE BORD 15
BUREAUX VIRTUELS 16
ACCESSIBILITE 17
PERSONNALISER LE MENU 17
KDE, XCFE & CO. 18
OGICIELS : TRUCS ET ASTUCES 20
LA FIN DES BOITES NOIRES : QUELS SONT LES PROGRAMMES REELLEMENT ACTIFS SUR MON PC ? 20
PILOTES PROPRIETAIRES 25
LOGICIELS PROPRIETAIRES INTERESSANTS 27
BACKPORTS 28
DE QUELS LOGICIELS AI-JE BESOIN ? 29
ESEAU 32
COMMENT FONCTIONNE UN RESEAU ? 32
CONFIGURATION DU RESEAU 36
VERIFIER LA COMMUNICATION RESEAU 38
DETECTION SYSTEMATIQUE DES ANOMALIES 41
ACCES AUX PARTAGES WINDOWS (ET MAC) 42
ACCEDER A LINUX DEPUIS UN PC OU UN MAC ? 43
BUREAU A DISTANCE 44
CONNEXION FTP 46
SECURITE 50
INTEGRITE DES TELECHARGEMENTS 50
GNUPG 52
LE DOSSIER PRIVATE 60
TRUECRYPT 60
CONSEILS DE SECURITE 61
SAUVEGARDES 66
QUELS ELEMENTS SAUVEGARDER ? 66
STRATEGIES DE SAUVEGARDE 66
SAUVEGARDER 67
RESTAURER 70
UTILISER DES CD ET DES DVD 71
LE SHELL –LINUX A L'ETAT PUR ! 72
TRUCS ET ASTUCES 72
COMMANDES UTILES 76
PROGRAMMATION D'UN SCRIPT SHELL 81
LE MOT DE LA FIN 90
INDEX 91
CATALOGUE COMPÉTENCE MICRO 93
CATALOGUE COMPÉTENCE MAC ET COMPÉTENCE PHOTO 94
BON DE COMMANDE 95
UBUNTU 96

ISBN : 978-2-35564-017-9

www.competencemicro.com

Linux

Compétence Micro

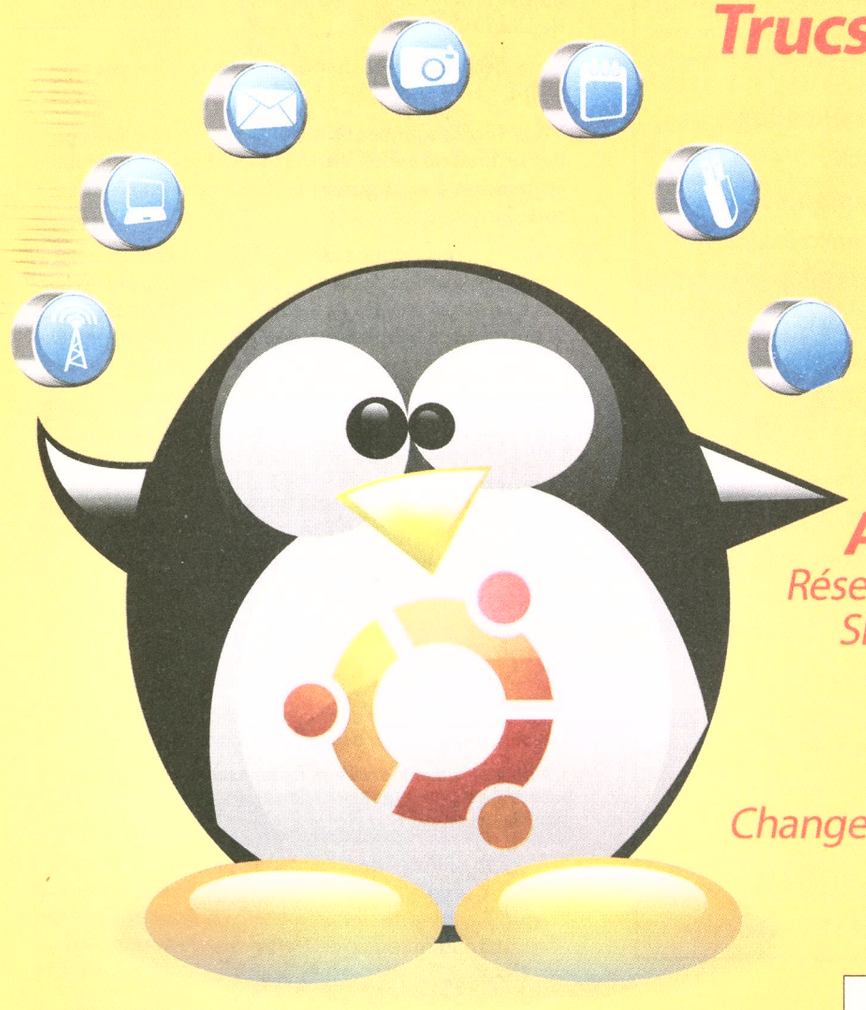
Le plaisir d'apprendre

Numéro 7

LINUX

Ubuntu

Trucs et astuces



Allez plus loin

Réseau Windows et Mac
Shell, scripts, Backups
... et beaucoup plus

Linux Sexy ?

Changez le look de Ubuntu
Thèmes, bureau, ...

Numéro 7 – Linux - Avril / Mai 2009 – France : 6,00 € - DOM : 7,20 € - BEL : 6,00 €

Ubuntu

Trucs et Astuces

Eric Amberg

Traduction : Catherine Viereck
Révision : David Bosman

© Copyright : KnowWare EURL

Société d'édition :
KnowWare E.U.R.L.
70, rue Georges Villette
F-77250 Ecuelles
France
Responsable de publication:
Mikkel Franck
Adresse électronique :
mfranck@competencemicro.com
Dépôt légal : Mars 2009
ISBN: 978-2-35564-017-9
EAN : 9782355640179
ISSN : 1963-2932
Imprimeur :
Imprintsa, Espagne
D.L Num. B.55194-2008
Imprimé en Espagne
Printed in Spain

Vente au numéro :
Voyez page 93 ou cherchez un revendeur sur le site de Compétence Micro.
Contact réservé aux dépositaires de presse :
Média dif - Olivier Le Potvin
Tél. : 33 (1) 01 49 72 59 53
Fax : 33 (1) 01 43 60 05 83
olpotvin@digicia.com

Ce numéro n'est pas un guide d'initiation à Ubuntu ou Linux. Pour cela, je vous renvoie à *Ubuntu un jeu d'enfant*, que nous avons déjà publié chez Compétence Micro. Mais ce n'est pas pour autant un guide qui s'adresse uniquement aux pros.

Au contraire ! A travers une sélection d'astuces adaptées, on propose au débutant comme à l'utilisateur un peu plus expérimenté d'aller plus loin dans la maîtrise de leur ordinateur et l'utilisation de Ubuntu.

Vous n'aimez pas les couleurs et le look par défaut de Ubuntu (moi non plus) ? Vous verrez comment changer cela en utilisant les thèmes, comment ajouter ou modifier l'affichage des polices, modifier les tableaux de bord, etc. Vous personnaliserez votre espace de travail selon vos besoins et vos envies.

Vous trouverez des conseils pratiques pour l'installation de logiciels (pilotes propriétaires, plugins multimédias, backports), pour l'utilisation des outils en ligne de commande, et sur les mises à jour de Ubuntu.

Vous découvrirez le shell et réaliserez votre premier script.

Vous verrez comment utiliser Linux en réseau avec des ordinateurs sous Windows et Mac OSX, et vous apprendrez à configurer manuellement un réseau sous Linux.

L'auteur vous donne enfin quelques informations sur l'organisation du disque dur et des conseils relatifs à la sécurité (GPG et emails, dossier Private,...) et aux bonnes habitudes à prendre pour vous protéger. Vous verrez enfin comment réaliser des sauvegardes fiables de vos données.

Bref, voici *notre* sélection de trucs et astuces qui, je l'espère, rendra votre utilisation de Ubuntu encore plus agréable.

David Bosman

Depuis son vieux portable sous Ubuntu 8.10 ☺

Tous les produits cités dans cette publication sont des marques déposées ou des marques commerciales.

L'auteur et l'éditeur déclinent toute responsabilité pouvant provenir de l'usage des données ou programmes figurant dans cet ouvrage.

Introduction

C'est parti !

Vous avez fait vos premiers pas avec Ubuntu-Linux et avez apprécié l'expérience ? Vous voulez passer à la vitesse supérieure et vous avez besoin pour cela d'un peu d'aide ? Ce guide est précisément ce qu'il vous faut ! Je vous y expliquerai entre autres comment :

- adapter l'interface de votre bureau,
- tirer le meilleur du gestionnaire de paquets Synaptic,
- effectuer des installations à partir de la console grâce à apt-get,
- crypter des fichiers,
- gérer des services et supprimer les éléments inutiles,
- réaliser des sauvegardes régulières de vos fichiers importants,
- écrire vos propres scripts shell,

... et bien plus encore. Vous découvrirez aussi de nombreux trucs et astuces applicables dans des cas de figures divers et variés.

Ce manuel vous permettra de maîtriser votre système Ubuntu. Ubuntu-Linux va vous étonner par l'ampleur de ses capacités !

Les conditions à remplir

Je pars du principe que vous avez déjà installé et utilisé Linux. Pas besoin d'être un gourou, *au contraire*, mais vous devez disposer de connaissances de base en informatique et surtout *d'une bonne dose de curiosité*.

Idéalement, vous aurez lu le guide de Compétence Micro sur Ubuntu (« Linux Un jeu d'enfant »). Sinon, je vous recommande très vivement de le consulter avant d'aborder le présent manuel, mais cela n'est pas obligatoire.

Quelle version d'Ubuntu ?

Canonical (le distributeur d'Ubuntu) publie deux fois par an aux alentours d'avril et d'octobre une nouvelle version d'Ubuntu. Nous utiliserons Ubuntu 8.10, nom de code *Intrepid Ibex* (bouquetin intrépide).

À vrai dire, le « bouquetin intrépide » ne propose pas beaucoup de nouveautés par rapport à la précédente version 8.04 (vous pourrez donc aussi l'utiliser sans trop de surprises), mais certains détails ont été améliorés ou ajoutés. On peut y noter entre autres :

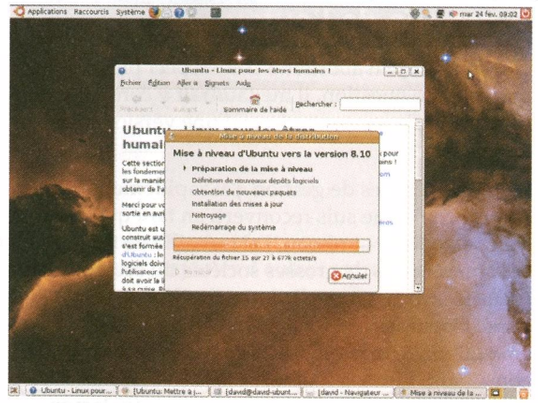
- de meilleures possibilités de réglage pour la résolution de l'écran,
- cryptage des fichiers intégré avec GNUPG,
- OpenOffice.org 2.4,
- logiciel de traitement d'images Gimp 2.6,
- nouveaux thèmes pour le bureau,
- création de connexions UMTS.

Mise à jour

Si vous disposez de Ubuntu 8.04, vous avez deux possibilités pour passer à la version 8.10.

Tout réinstaller (après avoir sauvegardé vos données personnelles). C'est souvent la meilleure option. Téléchargez le CD de Intrepid Ibex et faites une installation normale de Ubuntu.

Faire une mise à jour. L'avantage, c'est que vous ne devez rien réinstaller (mais ça ne vous exonère pas de faire vos backups)! Vous pouvez-même continuer à travailler dans Ubuntu pendant qu'il fait la mise à jour :



Il faudra quand même redémarrer après la mise à jour ;-)

Pour lancer la mise à jour depuis 8.04, ouvrez une console et tapez :

gksu 'update-manager -c'

Plus d'informations ici :

http://doc.ubuntu-fr.org/migration_hardy_intrepid

La mise à jour n'est pas recommandée si vous avez déjà lourdement modifié votre version de Ubuntu.

Ubuntu, Kubuntu ou encore ?

Quelle est la variante d'Ubuntu la mieux adaptée ? Ubuntu avec GNOME en tant qu'interface graphique ? Kubuntu avec KDE ? Xubuntu avec le gracile Xfce ? Edubuntu ? … ? Les variantes ne manquent pas !

Il s'agit d'une question de goût personnel et de connaître vos besoins. Mais aussi des performances de votre machine : Xubuntu sera moins lourde que d'autres et sera mieux adaptée à un ordinateur un peu ancien. Cela dit, il est très aisé de changer d'interface graphique dans Linux. Si un jour vous décidez finalement que le bureau GNOME ne correspond pas vraiment à vos attentes, aucun problème. Vous pouvez à tout moment passer à KDE ou par ex. à Xfce. Je vous expliquerai la procédure.

GNOME étant le bureau standard pour Ubuntu, c'est sur lui que nous nous baserons. Nous adapterons le reste ensemble en temps voulu.

Vous n'avez pas encore installé Ubuntu et vous ne disposez pas du manuel « Linux Un jeu d'enfant » ? Surfez à l'adresse www.ubuntu.com ou www.ubuntu-fr.org pour obtenir plus d'informations sur le téléchargement et l'installation de cette excellente distribution de Linux !

À propos de l'auteur

Qui suis-je ? Je n'aborderai pas l'aspect philosophique de cette question, il me faudrait y consacrer un livre entier, je me contenterai d'une version plus simple :

Après des études de gestion, j'ai rapidement changé de branche et me suis reconverti en tant qu'administrateur système et réseau. J'ai travaillé de nombreuses années dans de grosses sociétés du secteur de la sécurité informatique, l'administration réseau et système. Depuis début 2009, je travaille à mon compte en qualité de formateur informatique, auteur et consultant. Linux a toujours été mon système d'exploitation de prédilection, et je l'utilise également dans le cadre de ma profession.

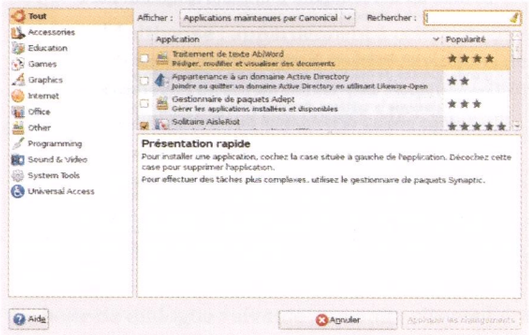
Assez parlé, entrons dans le vif du sujet...

Installation de logiciels

Je vous propose d'abord une petite introduction à la gestion avancée des paquets. Les outils et programmes présentés ici vous seront utiles régulièrement tout au long de ce guide (et de votre travail dans Linux). Posons les bases à partir desquelles s'organisera la suite.

Gestion des paquets : Synaptic

Vous voulez ajouter ou supprimer des programmes ? Allez dans le menu **Applications** puis choisissez la commande **Ajouter/Supprimer...**.



La liste de programmes est organisée par catégories dans lesquelles Ubuntu vous propose uniquement des programmes sélectionnés.

Mais en votre qualité d'utilisateur expérimenté de Linux, vous serez rapidement confronté à la question de l'installation de certaines bibliothèques, certains outils, programmes d'aide ou autres paquets. Autrement dit, des applications qu'Ubuntu ne vous propose *pas* dans la boîte de dialogue *Ajouter/supprimer des applications*.

C'est là qu'intervient *Synaptic*, l'interface de gestion des logiciels d'un système Ubuntu. Il sert d'interface pour les APT (Advanced Package Tools), un ensemble d'outils basés sur Debian pour la gestion des paquets de distribution Linux. Même si son utilisation est très intuitive, observons *Synaptic* d'un peu plus près afin de découvrir également quelques fonctions cachées.

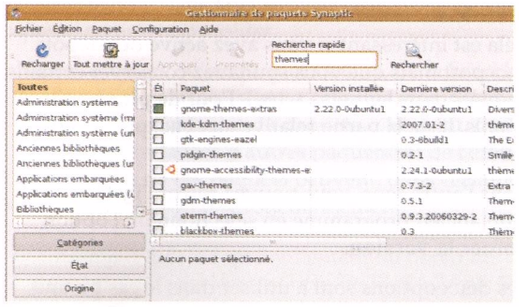
Invoquer Synaptic

Pour invoquer Synaptic, utilisez le menu **Système**, sous-menu **Administration**.

Vu qu'il s'agit d'un programme de gestion système, vous devez d'abord saisir votre mot de passe afin de pouvoir y accéder en tant qu'administrateur.

Sélectionner des paquets

Vous connaissez le nom ou une partie du nom d'un paquet ? Vous pouvez alors le saisir dans le champ de la recherche rapide. *Synaptic* met à jour automatiquement la liste de sélection dans la fenêtre principale :

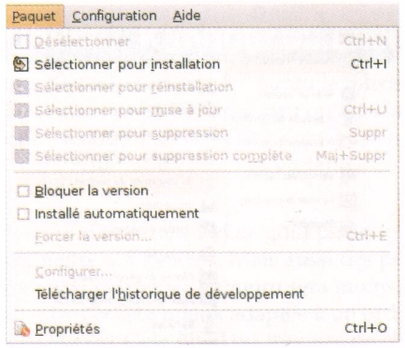


Si le paquet recherché n'est pas trouvé, vous devez éventuellement ajouter d'autres *sources* de paquets (voir plus bas). Vérifiez aussi que `Toutes` est sélectionné dans la liste de gauche : sans cela, seuls les éléments correspondant à la catégorie sélectionnée seront affichés.

Vous avez trouvé votre paquet ? Vous pouvez alors le sélectionner pour l'installation. Si vous le souhaitez, vous pouvez répéter la procédure avec d'autres paquets. Cliquez pour finir sur **Appliquer** afin de lancer l'installation de tous les paquets sélectionnés.

Options des paquets

Vous n'avez probablement rien découvert de neuf jusqu'ici. Nous allons à présent franchir un petit pas de plus : sélectionnez un paquet puis cliquez sur le menu **Paquet**. Les choses commencent déjà à devenir plus intéressantes.



Si vous avez déjà installé un paquet, vous pouvez effectuer ici une nouvelle installation pour par exemple réparer un programme défectueux.

La différence entre **Sélectionner pour suppression** et **Sélectionner pour suppression complète** : dans la première, des fichiers de configuration du programme peuvent être éventuellement conservés, d'où la possibilité d'effectuer une nouvelle installation du programme sur la configuration et retrouver vos réglages.

Si vous sélectionnez l'option **Bloquer la version**, le programme actuellement installé est conservé même si de nouvelles versions du paquet sont disponibles dans les sources de mises à jour.

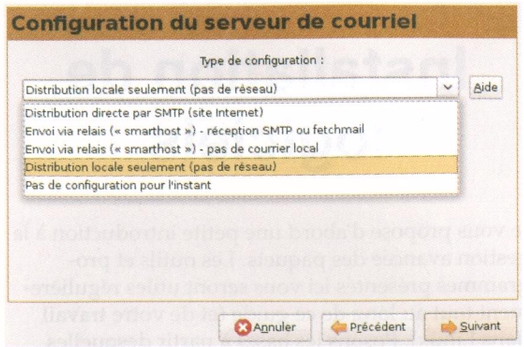
Cela est intéressant si vous avez activé des dépôts backport mais que vous souhaitez conserver certains paquets dans leur état actuel. Pour de plus amples détails, lisez la partie relative aux backports, plus loin.

Dans l'autre sens, vous pouvez forcer l'installation d'un ancien programme en sélectionnant l'option **Forcer la version**.

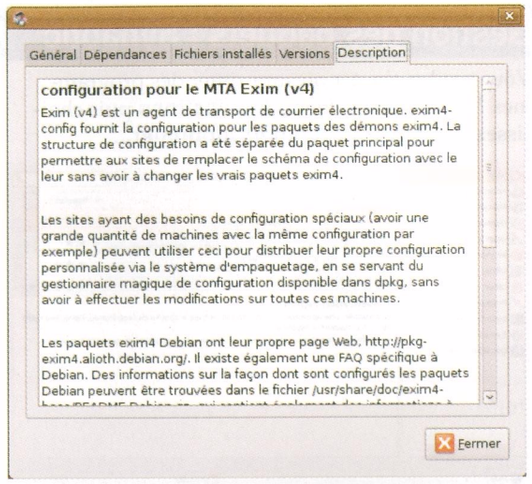
Les deux options sont à utiliser dans le cas où une certaine version spécifique d'un programme devrait être installée et conservée. N'y touchez pas le reste du temps.

Configurer des paquets via Debconf

Le gestionnaire de paquets Debian permet la configuration du paquet d'un programme au cours de l'installation par le biais de *Debconf*. La disponibilité de cette option et les possibilités de configuration dépendent du concepteur du paquet. Si elle est disponible, vous pouvez reconfigurer des paquets installés par le biais de l'option **Configurer**. Cela peut être très utile si vous avez configuré le programme et que vous souhaitez à présent restaurer l'état d'origine. Pour illustrer mes propos, je prends pour exemple le serveur de messagerie *Exim* qui, à vrai dire, n'est pas installé de base avec Ubuntu :



L'affichage des propriétés d'un paquet est une fonction très utile :



Outre le descriptif du contenu du paquet, vous pouvez afficher ici les fichiers d'installation du paquet. D'un simple coup d'oeil, vous découvrez les dossiers dans lesquels se niche le paquet.

La liste des dépendances est un autre élément très intéressant. Elle vous présente les liens existant entre les différents paquets.

> La problématique des dépendances a été heureusement résolue depuis grâce aux très efficaces outils de gestion des paquets. Avant cela, les utilisateurs des systèmes Linux s'arrachaient les cheveux à chaque fois qu'ils voulaient installer un paquet dépendant d'autres paquets. Car ces derniers pouvaient parfaitement avoir encore leurs propres dépendances – et l'on s'éloignait rapidement du sujet…
>
> Actuellement, vous êtes confronté à ce problème uniquement si vous souhaitez installer des paquets n'appartenant pas à la distribution.

Dépôts : les sources de paquets

Les paquets disponibles sous Ubuntu sont prélevés à partir de certaines **sources de paquets** (les dépôts).

Par exemple le CD ou DVD d'installation de Ubuntu, mais aussi des sources Internet.

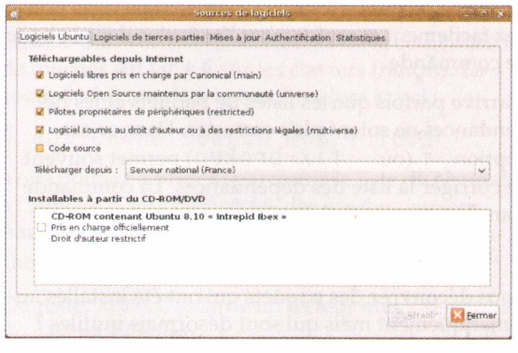
Le contenu d'une source est trié en fonction du nom des **archives**, par ex. *intrepid*, *intrepid-updates* ou *intrepid-security*. Le contenu d'une archive est divisé en **composants**, au nombre de quatre :

- **Main** : englobe tous les paquets répondant aux conditions de licence d'Ubuntu et supportés par l'équipe Ubuntu.
- **Restricted** : contient des logiciels importants et de ce fait supportés par les développeurs d'Ubuntu bien que non couverts par une licence ouverte adaptée qui leur aurait permis d'être intégrés à *Main*. Il s'agit par exemple de certains pilotes de cartes graphiques.
- **Universe** : englobe un large éventail de logiciels gratuits, mais non supportés ni entretenus par l'équipe Ubuntu.
- **Multiverse** : appartiennent à cette catégorie les logiciels soumis à des restrictions de licence et non supportés par l'équipe Ubuntu. Par exemple des codecs MP3, etc.

Pour configurer vous-même les sources de paquets dans Synaptic, suivez la procédure ci-dessous :

Via **Configuration > Dépôts** ou via le menu **Système**, sous **Administration**. Choisissez **Sources de logiciels**.

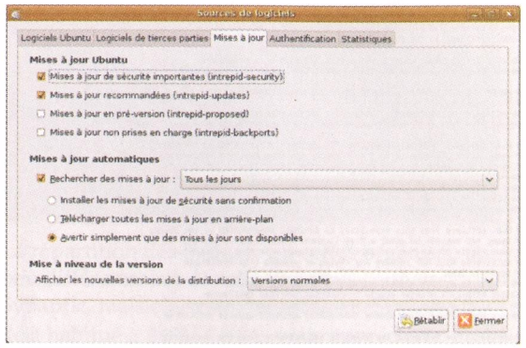
La boîte de dialogue suivante apparaît ; elle contient un onglet particulièrement important nommé **Logiciels Ubuntu**.



Dans cet onglet, sélectionnez uniquement les composants auxquels vous voulez accéder dans les dépôts. La liste déroulante **Télécharger depuis :** vous permet de sélectionner le *miroir* à utiliser en qualité de dépôt. Un miroir est un serveur miroir donnant un accès aux dépôts d'Ubuntu sous la forme rigoureusement exacte de l'original, à savoir le serveur principal.

C'est depuis Gutsy Gibbon (Ubuntu 7.10) qu'est disponible le composant *Partner* mettant à disposition des logiciels commerciaux dont le code source n'est pas disponible. C'est pour éviter de devoir systématiquement télécharger ces logiciels directement à partir du fabricant et pouvoir assurer des mises à jour régulières que cette archive a été créée. Elle est localisée dans l'onglet **Logiciels de tierces parties**.

L'onglet **Mises à jour** vous permet de définir le comportement de la mise à jour d'Ubuntu :



Les préréglages sont généralement adaptés et n'ont pas besoin d'être modifiés.

Vos propres sources de logiciels

En théorie, vous pouvez également ajouter vos propres sources – dans la partie suivante, je vous expliquerai comment procéder.

Notez toutefois qu'un paquet Debian, de la branche Testing par exemple, peut en principe être installé et géré via le gestionnaire de paquets d'Ubuntu, mais que les dépendances d'autres paquets ou de certaines versions de ces paquets peuvent, dans certains cas, empêcher son bon fonctionnement.

> Autrement dit : le fonctionnement ou non d'un paquet issu d'une autre source de logiciels que les dépôts Ubuntu est une pure question de chance !

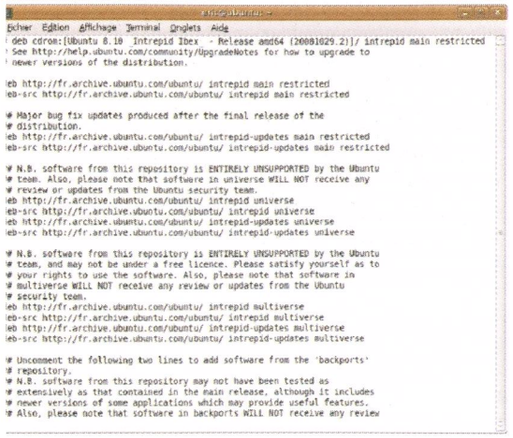
Comment utiliser APT ?

Comme je l'ai déjà mentionné, Synaptic est uniquement une interface graphique pour APT. Cet ensemble d'outils peut bien sûr aussi être utilisé directement. L'avantage est que l'on obtient davantage de transparence qu'avec une interface graphique dans laquelle les événements se déroulent en arrière-plan.

> Affichez une console : **Applications->Accessoires** et choisissez **Terminal**. Le mieux consiste à placer cet élément directement sur votre tableau de bord en le faisant simplement glisser dessus. De cette façon, vous y accéderez plus rapidement.

.es sources de logiciels

:omme promis, je vais vous montrer où se trouve la iste contenant les sources de logiciels d'Ubuntu. Jeez un oeil dans `/etc/apt/sources.list` (par :xemple avec `less`). Vous y trouvez toute une série l'éléments :



'ar exemple :

```
leb http://de.archive.ubuntu.com/ubuntu/
 ntrepid main restricted
```

Une archive compatible Debian commence par `deb` ou `deb-src` pour le code source. S'ensuit l'URL puis les composants.

Si vous souhaitez ajouter des sources de logiciels personnelles (par exemple pour accéder à un logiciel qui n'est pas encore repris dans les dépôts officiels), insérez-les dans ce fichier par le biais de l'éditeur de votre choix (par ex. nano ou vim). Je déconseille toutefois de le faire sauf si vous savez précisément ce que vous faites (voir les raisons énoncées plus haut) !

apt-get

L'outil qui travaille dans l'ombre de *Synaptic* lors de l'installation ou de la désinstallation de paquets se nomme `apt-get`.

> Pour pouvoir utiliser `apt-get`, vous devez ajouter `sudo` devant chaque ligne de commandes (`sudo apt-get` ...). Ou alors saisir une seule fois `sudo -s` dans la ligne de commandes afin de pouvoir travailler avec des droits root durant toute la session de la console.

La syntaxe suivante vous permet d'installer n'importe quel paquet Ubuntu :

```
# apt-get install <nom du paquet>
```

Les signes « plus petit que » et « plus grand que » servent uniquement à délimiter l'emplacement du nom du paquet. Ils ne font pas partie du nom du paquet. Les dépendances du paquet sont automatiquement trouvées et installées.

> Le signe dièse # apparaît automatiquement, vous n'avez donc pas à le saisir ! Dans un terminal/une console, le signe # signifie que vous êtes connecté avec les privilèges de root.
>
> La commande que nous venons de voir signifie donc ceci : vous avez ouvert un terminal, vous êtes connecté en tant que root (par exemple via `sudo -s`) et vous exécutez la commande au niveau d'une invite root, précisément au niveau de ce signe #.

Pour désinstaller un paquet, saisissez ceci :

```
# apt-get remove <nom du paquet>
```

Les fichiers de configuration éventuellement présents ne sont pas supprimés et peuvent être réutilisés lors d'une nouvelle installation. Si vous souhaitez également les effacer, utilisez `purge` à la place de `remove`.

Vous voulez actualiser votre système ? Utilisez alors la commande suivante :

```
# apt-get update
```

Cette ligne actualise la liste des paquets pour APT et identifie les nouvelles mises à jour. Pour installer les mises à jour, saisissez maintenant ceci :

```
# apt-get upgrade
```

Cela permet d'actualiser tous les paquets présents pour lesquels une version plus récente est disponible. Cela revient au même que de passer par le gestionnaire de mise à jour de Gnome. Mais vous pouvez facilement et rapidement l'exécuter dans la ligne de commande.

Il arrive parfois que les listes de paquets et les dépendances ne soient plus correctes. Dans ce cas, l'option `-f` (ou `--fix-broken`) permet souvent de corriger la liste des dépendances. La commande complète ressemble à ceci :

```
# apt-get install -f
```

Vous découvrez des paquets qui ont été installés automatiquement mais qui sont désormais inutiles ? Vous pouvez les supprimer automatiquement par le biais de la commande suivante :

```
# apt-get autoremove
```

En procédant ainsi, vous conservez facilement la consistance de votre système même en cas de dysfonctionnement du gestionnaire de paquets d'Ubuntu.

apt-cache

`apt-cache` vous permet de manipuler le cache d'APT. La fonction la plus importante est la recherche de paquets spécifiques :

```
# apt-cache search <expression recherchée>
```

L'expression recherchée est une chaîne de caractères quelconque figurant dans le nom du paquet ou dans la description succincte du paquet. Vous pouvez aussi utiliser ici ce que l'on nomme des expressions régulières avec lesquelles il est possible de construire des modèles de recherche extrêmement complexes. Un vaste sujet sur lequel on pourrait rédiger encore de longues pages.

> Pour plus d'informations sur les expressions régulières, rendez-vous à l'adresse suivante :
>
> http://www.regular-expressions.info/tutorial.html

Installer un autre lecteur MP3

Intéressons-nous à présent à un exemple simple. Admettons que vous vouliez installer un autre lecteur MP3 en plus du lecteur Amarok déjà présent. Saisissez alors la ligne suivante :

```
# apt-cache search mp3
```

Le résultat est déjà relativement volumineux et compte au total plus de 300 résultats. Pour consulter plus facilement cette liste, vous pouvez la filtrer en utilisant ce que l'on appelle un pagineur. Un pagineur permet d'afficher une liste de résultats par page ou par ligne. Le pagineur par défaut de Linux est `less`. Pour pouvoir faire passer le résultat de `apt-cache search` à `less`, utilisez la touche pipe, le trait vertical (|) accessible via la combinaison de touches **Alt Gr + 6** sur les claviers français. La commande complète ressemble alors à ceci :

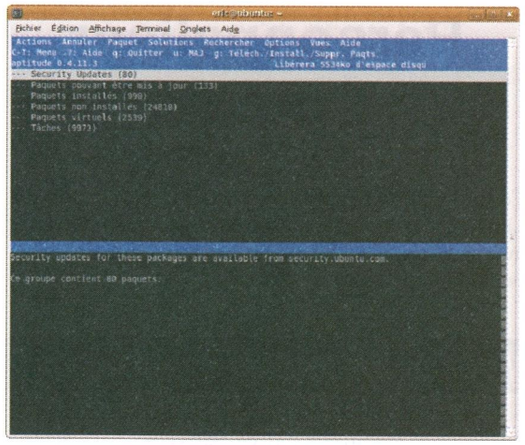
```
# apt-cache search mp3 | less
```

Vous pouvez à présent parcourir la liste de résultats à l'aide de la touche **Entrée**, des touches curseur **haut** et **bas** ainsi que de la touche **Espace**. La touche **q** vous permet de quitter `less`.

Du reste, `xmms2` incarne un lecteur musical inspiré du célèbre Winamp …

aptitude

aptitude est une interface utile d'APT. Il s'agit d'un programme de gestion de paquets exceptionnellement performant, emprunté de Debian GNU/Linux, ne dépendant d'aucune interface utilisateur graphique et qui, de ce fait, peut également être utilisé via une connexion shell distante, comme par exemple SSH. Sous Debian, `aptitude` représente l'outil de gestion de paquets standard. Pour l'invoquer, vous devez saisir son nom dans la ligne de commandes :



La répartition des paquets est certes organisée un peu différemment de celle que vous connaissez de Synaptic, mais elle a son importance. Si vous êtes déjà habitué à `aptitude`, vous apprécierez probablement son utilisation. Appuyez sur **Entrée** pour déployer une des sections présentées. Vous devez ensuite parcourir plusieurs niveaux avant de voir les paquets localisés sous la rubrique concernée. Appuyez sur **+** et **-** afin de sélectionner ou désélectionner un paquet à installer. Les dépendances sont automatiquement recherchées. Appuyez sur **g** pour démarrer l'installation ou la désinstallation.

Pour sortir d'un niveau, appuyez sur **q**. Cette touche vous permet également de quitter `aptitude`.

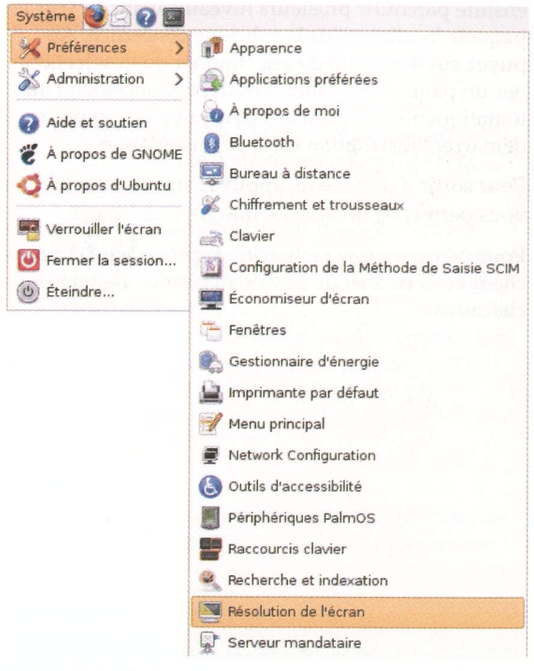
Pour démarrer une recherche, appuyez sur **/**. La touche **n** vous permet de passer au résultat de recherche suivant.

Personnaliser Ubuntu

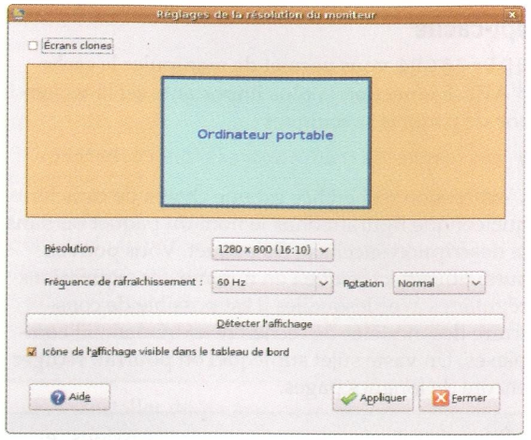
Comme c'est le cas pour la plupart des distributions Linux, la forme et l'apparence d'Ubuntu sont extrêmement variables. Jetons un oeil aux possibilités de personnalisation du système.

Modifier les réglages de l'écran

Si vous utilisez un écran plat, il est essentiel d'utiliser sa résolution native. Sans cela, l'affichage sera beaucoup moins agréable à regarder. Les écrans cathodiques, plus anciens, peuvent utiliser différentes résolutions sans souci. Ce n'est pas le cas des écrans plats. Si l'affichage vous semble médiocre, allez dans **Système/Préférences/Résolution de l'écran** :



La boîte de dialogue qui apparaît vous permet de régler la résolution de l'écran ainsi que d'autres paramètres :



Vous possédez encore un moniteur cathodique (CRT) et non pas un écran plat ? Vous pouvez aussi définir ici la fréquence de rafraîchissement.

> Notez que le mauvais réglage d'un moniteur cathodique peut non seulement perturber l'affichage, mais carrément endommager l'appareil. Fort heureusement, ce problème ne survient plus qu'avec les très anciens écrans ne disposant pas encore des mécanismes de protection appropriés.

Comme son nom l'indique, le bouton **Détecter l'affichage** vous permet de reconnaître votre écran. Pour que cela fonctionne, il faut que la carte graphique soit reconnue et les pilotes installés (par exemple Nvidia). Certaines cartes graphiques sont encore assez mal supportées sous Linux, les fabricants ne faisant pas l'effort de proposer des pilotes. Il vaut donc mieux se renseigner, par exemple sur les forums d'utilisateurs.

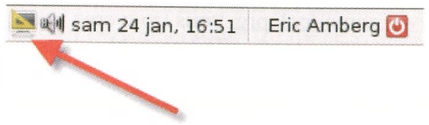
> ubuntu-fr
>
> NDLE : le forum de Ubuntu.fr, la communauté des utilisateurs francophones de Ubuntu, est très actif et accueillant. Mais avant de poser vos questions, faites l'effort de vérifier si elle n'a pas déjà été posée et répondue : les participants du forum sont des bénévoles, pas une hotline gratuite ;-)
>
> http://forum.ubuntu-fr.org

Ubuntu supporte l'utilisation de plusieurs écrans. Cliquez sur l'icône de l'écran concerné pour régler les paramètres souhaités. Toutefois, cet outil ne peut être utilisé que de façon très rudimentaire avec les autres fonctions. S'il s'avère que vous utilisez une carte graphique performante et deux écrans, vous devrez dans tous les cas installer le pilote approprié de la carte graphique et configurer le tout par le biais de l'outil d'administration avec lequel il est livré.

Plus loin, dans la partie relative aux pilotes, je vous expliquerai plus en détail l'intégration et l'adaptation avec des pilotes de cartes graphiques ATI et Nvidia. Je vous montrerai également comment activer certains effets. Recherchez dans l'index les mots-clé ATI ou Nvidia.

Rotation ne fonctionne qu'avec les écrans compatibles qui permettent de faire pivoter l'écran sur un axe. Vous pouvez alors demander à Linux d'afficher le Bureau en mode portrait et non plus en mode paysage, par une rotation de 90°. Vous pouvez aussi sélectionner la valeur *auto* afin que la rotation des contenus s'effectue automatiquement lorsque vous faites pivoter l'écran.

Vous avez coché l'option *Icône de l'affichage visible dans le tableau de bord* ? Une icône en forme d'écran apparaît alors en haut à droite du tableau de bord standard :



Cette icône vous permet à présent aussi d'accéder directement aux réglages de l'écran.

Cliquez sur **Appliquer** et **Fermer** pour activer les modifications et fermer la boîte de dialogue.

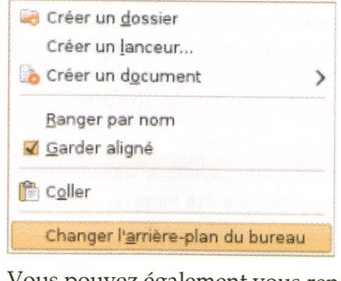
Si malgré tout l'affichage reste médiocre, il faudra sans doute passer par la configuration manuelle de certains fichiers. Cela sort du cadre de ce guide.

Modifier l'apparence du bureau

Le bureau Ubuntu peut être personnalisé de manière très détaillée. Jetons un oeil aux principales possibilités de réglage.

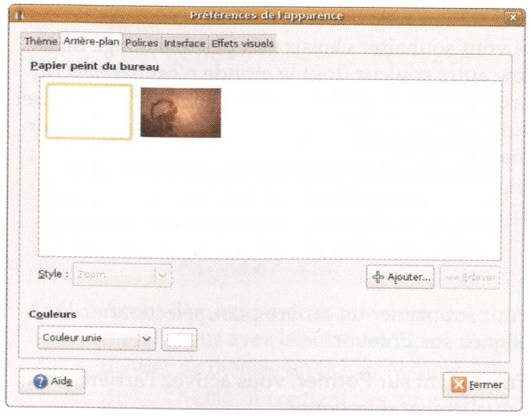
Changer le fond d'écran

Cliquez sur une zone vide de votre bureau et choisissez la commande **Changer l'arrière-plan du bureau** :



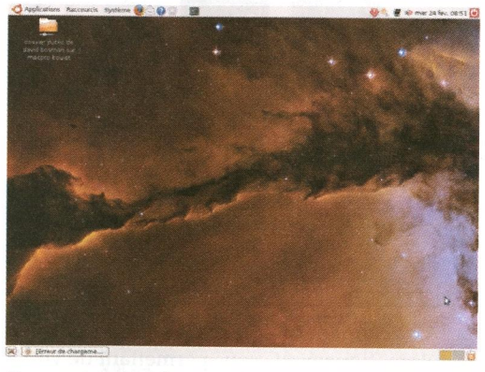
Vous pouvez également vous rendre dans le menu **Système / Préférences / Apparence**.

Affichez l'onglet Arrière-plan s'il n'est pas déjà affiché.



Outre la couleur, vous pouvez également choisir une image pour l'arrière-plan. Le choix est limité, mais rien ne vous empêche d'ajouter vos images !

Cliquez sur **Ajouter** pour sélectionner une nouvelle image d'arrière-plan et l'ajouter à la sélection. Si vous voulez en ajouter plusieurs, il faudra répéter l'opération (ce serait bien de pouvoir ajouter un dossier d'un coup, par exemple). Choisissez la nouvelle image :



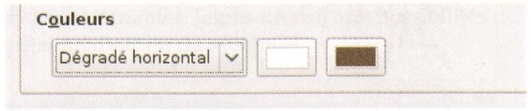
Pour rester dans les tons de Ubuntu, une image disponible sur le site de la NASA.

Les « images par défaut » sont stockées dans le dossier Backgrounds. Mais vous pouvez placer vos images où vous le souhaitez dans votre dossier d'utilisateur.

> Les images de la NASA sont disponibles ici :
>
> www.nasa.gov/multimedia/imagegallery/iotd.html
>
> Vous trouverez à l'adresse http://www.wallpapers.org une très belle page d'accueil pour la recherche d'images d'arrière-plan. Sinon, tapez simplement le mot-clé `wallpaper` dans votre moteur de recherche préféré et sélectionnez un lien.

La liste déroulante **Style** vous permet d'adapter les images ne correspondant pas à la résolution de votre écran.

Si vous souhaitez choisir non pas une image mais une autre couleur (tout le monde n'est pas *fana* du brun/orangé de Ubuntu ;-)), il vous suffit de cliquer sur la case de couleur. La liste déroulante vous permet aussi de sélectionner un dégradé de couleurs :

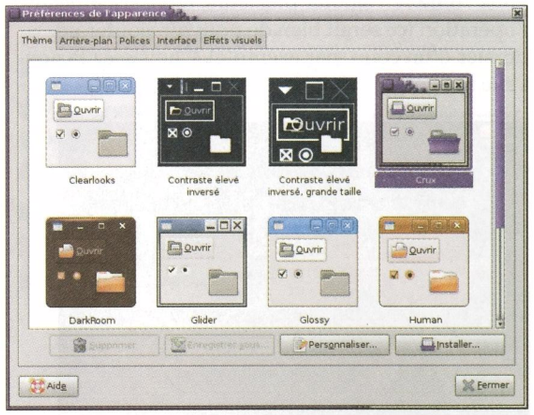


Pour supprimer un arrière-plan, sélectionnez-le et cliquez sur **Enlever**.

En cliquant sur **Fermer**, vous activez l'arrière-plan actuellement sélectionné.

Changer le look des fenêtres : les thèmes

L'onglet **Thème** contient différents thèmes de bureau. Un thème s'applique aux fenêtres et aux icônes.



Dès que vous cliquez sur l'un des thèmes, ce dernier s'active automatiquement, vous permettant de vous faire immédiatement une idée de l'apparence de votre bureau.

Cliquez sur **Personnaliser** pour modifier tel ou tel aspect d'un thème (par exemple pour utiliser les icônes du thème X avec le thème Y)

Installer de nouveaux thèmes

Les thèmes installés par défaut avec Ubuntu ne sont pas très nombreux, il y en a d'autres que vous pouvez installer par le biais du gestionnaire de paquets **Synaptic** ou d'**apt-get**. Pour ce faire, vous devez utiliser les paquets suivants :

- `community-themes` (installe les thèmes *NewWave*, *Dust* et *Kin)*
- `gnome-themes-extras` (installe *Darklooks* et *Unity*)

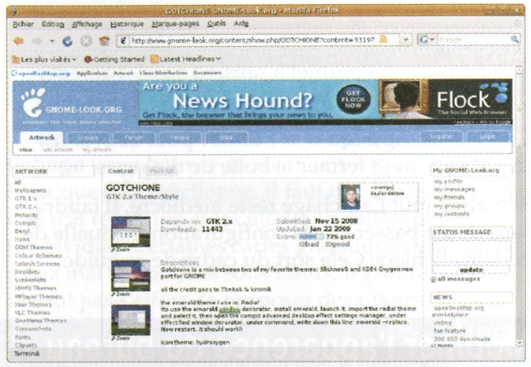
Le mieux consiste à taper les noms dans la recherche rapide de Synaptic. Assurez-vous d'abord que la source de logiciels *Universe* est active. Nous avons déjà abordé ce sujet quelques pages auparavant.

> Notez que ces paquets ne se trouvent pas dans le menu **Applications** sous **Ajouter/supprimer...**, car il ne s'agit pas ici d'applications autonomes. Vous devez donc impérativement utiliser Synaptic ou l'un des outils APT.

Si vous avez installé l'un des paquets ou les deux, vous trouverez les thèmes supplémentaires dans la sélection de l'onglet *Thème*.

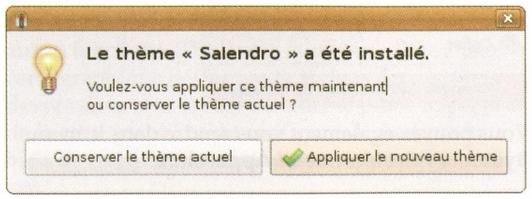
Télécharger des thèmes à partir d'Internet

Pour télécharger d'autres thèmes à partir d'Internet, vous pouvez visiter l'un des nombreux sites communautaires. Par exemple, sous http://www.gnome-look.org, vous trouverez une série de thèmes intéressants, téléchargeables sous la forme de fichiers d'archive (tar.gz) :

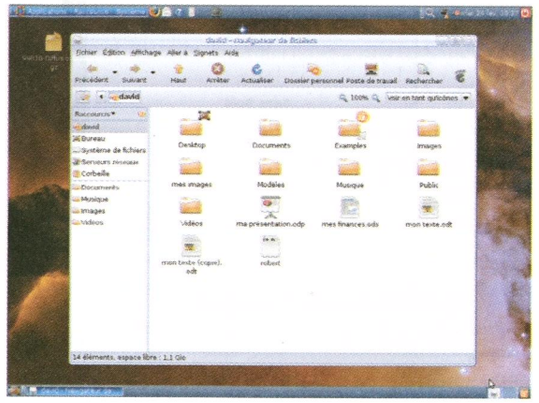


À gauche dans la navigation, choisissez GTK 2.x afin de rechercher les thèmes appropriés – par exemple *Gotchione*. Vous trouverez alors un peu plus bas le lien de téléchargement. Le « fichier thème » correspondant est enregistré par défaut sur votre bureau.

Ouvrez à présent la boîte de dialogue *Apparence* et allez dans l'onglet *Thème*. Faites glisser le fichier d'archive téléchargé et déposez-le dans la fenêtre de sélection. En règle générale, le thème est installé automatiquement. Un message vous informe de la réussite de l'installation. Je suis ravi que tout se soit bien déroulé, mais j'ai fini par choisir un autre thème :



Parfois il faudra redémarrer Ubuntu pour que tout le thème soit appliqué :



Si un thème est installé de cette façon, il est alors enregistré uniquement pour cet utilisateur dans son dossier personnel sous `./themes`. Mais il n'est disponible que pour cet utilisateur. Les autres utilisateurs n'y ont pour le moment pas accès.

Ajouter un thème pour tous les utilisateurs

Je vais à présent vous expliquer comment intégrer un thème pour tous les utilisateurs. Cela devient un peu plus technique.

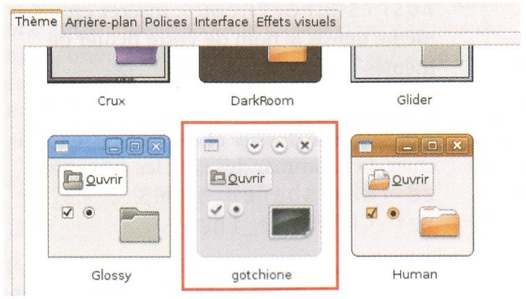
> Si vous souhaitez par contre obtenir des informations générales sur la personnalisation des thèmes, rendez-vous également à l'adresse http://www.techotopia.com/index.php/Configuring_the_Ubuntu_GNOME_Desktop_Background_and_Themes#Changing_the_GNOME_Desktop_Theme – cette fois-ci en anglais.

En votre qualité d'utilisateur normal, vous ne disposez d'aucun droit en écriture pour le dossier des thèmes `/usr/share/themes`. Nous devons nous rendre dans la console et copier le fichier à l'emplacement approprié. Procédez de la façon suivante :

1. Ouvrez un **Terminal** en le sélectionnant à partir du menu **Applications**, sous **Accessoires**.
2. Pour la suite de la procédure, vous aurez besoin à certains moments des privilèges d'administrateur. Vous devez donc saisir `sudo -s` puis votre mot de passe.
3. Allez tout d'abord dans le dossier de votre bureau contenant actuellement le fichier. Saisissez la ligne suivante : `cd /home/nom_utilisateur/Bureau`. Remplacez ici nom_utilisateur par le nom d'utilisateur sous lequel vous vous êtes enregistré.
4. Copiez à présent le fichier au bon emplacement. Saisissez pour cela la ligne suivante : `cp ./fichier_theme /usr/share/themes`. Insérez en même temps le nom de votre fichier de thèmes (par ex. `93197-gotchione.tar.gz`).
5. Allez dans le dossier et décompressez le fichier. Les commandes sont les suivantes : `cd /usr/share/themes ; tar -xzf <fichier_theme>`.
6. Avant de pouvoir utiliser le nouveau thème, vous avez besoin que figure dans le dossier de thèmes un fichier nommé `index.theme`. Il contient des informations sur le nom et la structure de ce thème. Comme exemple, il est préférable que vous commenciez par copier le fichier d'un autre thème, par ex. celui d'Human. Écrivez : `cp Human/index.theme gotchione/`. Pour que cette commande fonctionne, vous devez vous trouver dans le dossier `/usr/share/themes` et avoir décompressé le fichier d'archive du thème *gotchione*.
7. Adaptez maintenant le fichier `index.theme` au nouveau thème. Vous avez besoin pour cela d'un éditeur. Ouvrez le fichier avec par ex. `nano index.theme`. Dans l'exemple, vous pouvez éditer le fichier sans connaissances approfondies de la syntaxe. Pour commencer, nous pouvons nous limiter à la règle de base suivante : remplacer chaque occurrence de *Human* par *gotchione* ou le nom de votre thème. Supprimez toutes les lignes inutiles. Le fichier peut ensuite ressembler à ceci :

```
[Desktop Entry]
Type=X-GNOME-Metatheme
Name=gotchione
Comment=Mon premier thème personnel
Encoding=UTF-8

[X-GNOME-Metatheme]
GtkTheme=gotchione
IconTheme=gotchione
MetacityTheme=gotchione
CursorTheme=gotchione
```

Le fichier une fois enregistré, vous trouverez le nouveau thème dans la boîte de dialogue **Préférences > Apparence**, onglet **Thème** :



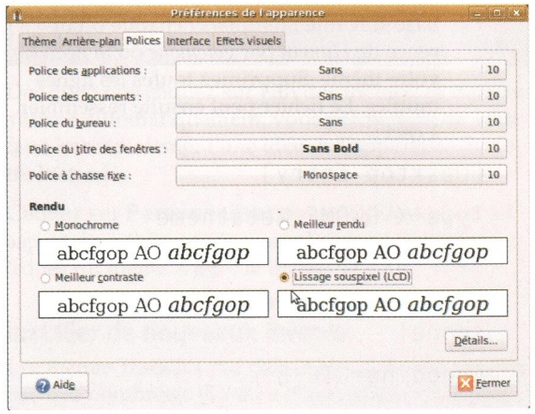
Un thème contient souvent différents fichiers d'archive que vous devez télécharger et installer séparément. Pour ce faire, copiez les fichiers concernés en fonction de leur contenu dans `/usr/share/themes`, `/usr/share/icons` ou `/usr/share/fonts`.

Je vous renvoie à ce propos aux liens mentionnés plus haut qui vous fourniront une introduction de base à l'intégration de thèmes GNOME. En cas de doute, une recherche dans Google vous permettra certainement de trouver des réponses !

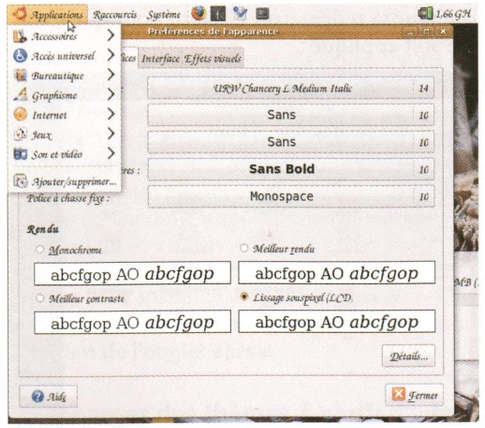
Taille et famille de polices

Vous êtes maître du rendu des polices et de leur taille d'affichage. C'est très pratique par exemple si vous trouvez les textes des menus trop petits pour être lisibles.

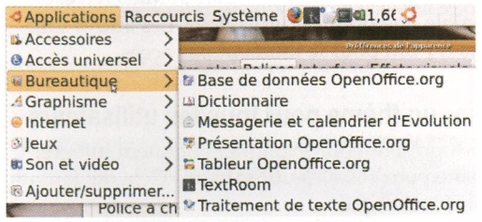
Allez dans les **préférences | Apparences** et affichez l'onglet **Polices** :



Pour chaque élément de la liste, vous pouvez changer la famille et la taille de la police :

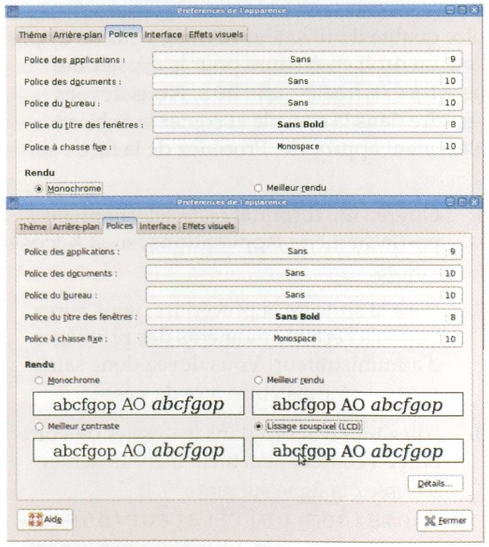


Cliquez sur la liste désirée pour modifier les valeurs. C'est très pratique quand on n'a pas une très bonne vue, ou lorsque la résolution de l'écran est si grande que le texte paraît minuscule :



Lissage

En bas de cet onglet, le **Rendu** permet de définir le lissage des polices à l'écran. Sans lissage, les polices sont un aspect en escalier un peu abrupt. Le lissage « triche » et atténue cet effet, rendant les polices plus douces à l'écran :



Sans et avec lissage.

Pour les ordinateurs portables ou les ordinateurs connectés à un écran plat, **Lissage de souspixel (LCD)** est généralement le mieux adapté.

Cliquez sur **Détails** pour accéder à des réglages plus fins.

Installer des polices Microsoft

Même si vous utilisez OpenOffice.org, vous souhaiterez certainement installer les polices Microsoft TrueType. Pourquoi ? Pour pouvoir afficher et modifier de la même façon les documents créés sous Microsoft Office !

Notons également que la majorité des pages Web ont été optimisées pour les polices Microsoft, de telle sorte que sous Linux, ces pages sont souvent affichées différemment. En effet, lorsque la police d'origine est indisponible, le navigateur la remplace par une police analogue présentant les mêmes caractéristiques.

Pour installer quelques unes des principales polices Microsoft, vous devez installer le paquet `msttcorefonts` via Synaptic ou `apt-get`. Pour pouvoir utiliser ces polices, vous devez ensuite :

Soit vous déconnecter et vous reconnecter,

Soit saisir la commande `sudo fc-cache -fv`.

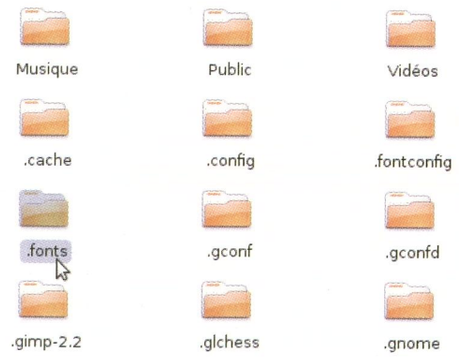
Les polices d'un système Ubuntu-Linux se trouvent généralement dans les sous-dossiers correspondants sous `/usr/share/fonts`. Pour ajouter d'autres polices, vous pouvez saisir le mot-clé `fonts` dans le gestionnaire de paquets Synaptic et sélectionner les paquets de polices appropriés à partir de la longue liste proposée.

> Pour plus d'informations sur l'intégration de polices dans Ubuntu, je vous conseille de consulter le site Web suivant : https://wiki.ubuntu.com/Fonts.

Installer d'autres polices ?

Linux est compatible avec les polices TrueType (nettement moins avec les modèles OpenType).

Pour en installer, ajoutez-les dans le répertoire `~/.fonts` de votre dossier utilisateur. C'est un dossier invisible, pour l'afficher allez dans **Affichage > Afficher les fichiers cachés** (**ctrl+h**) :

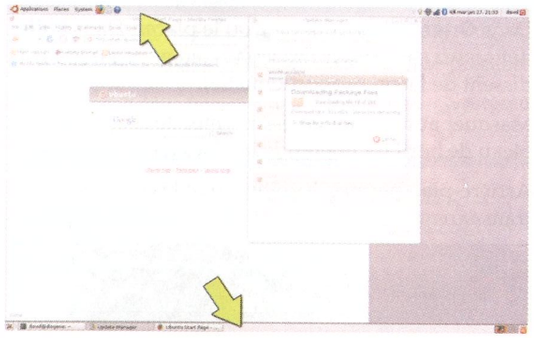


S'il n'existe pas créez-le.

Après y avoir fait glisser vos polices TrueType, dans la console, tapez `fc-cache -fv` pour recharger le cache des polices.

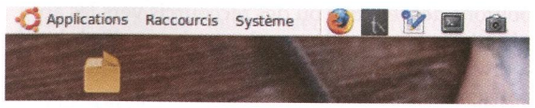
Tableaux de bord

Les Tableaux de bords peuvent être déplacés et personnalisés. Vous pouvez également en ajouter et en supprimer. Ce sont ces bandes grises :



Par défaut celle du haut contient le menu Ubuntu, des raccourcis vers des applications, la date et l'heure, l'état du réseau, etc. Celle du bas un accès rapide vers le Bureau, les boutons des fenêtres, les bureaux virtuels et la Corbeille.

La première chose à faire, à mon avis, c'est d'ajouter des raccourcis vers les applications les plus souvent utilisées :

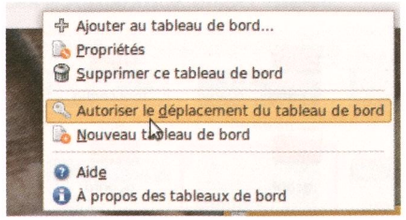


Faites simplement glisser l'icône du programme depuis le menu Applications quelque part sur tableau de bord. Vous pourrez la placer soigneusement plus tard.

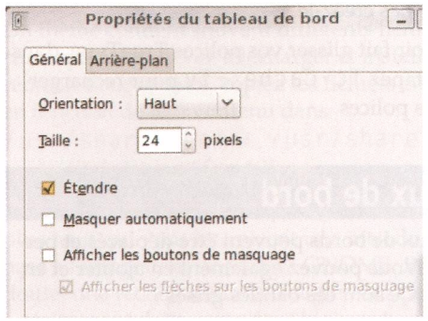
Emplacement et taille

Vous pouvez aussi changer leur position à l'écran. Pour les bouger, vous devez d'abord les déverrouil-

ler. Faites un clic droit sur une partie vide et cliquez sur Autoriser le déplacement du tableau de bord :



Pour le déplacer, vous pouvez à présent le faire glisser sur un autre bord de l'écran. Ou le faire via un clic droit pour afficher ses Propriétés :



Dans **Orientation**, indiquez où le placer. Dans **taille** définissez sa hauteur (les icônes (en général) grandissent de façon proportionnelle).

Masquer automatiquement permet de réduire la tableau de bord (et libérer de la place à l'écran).

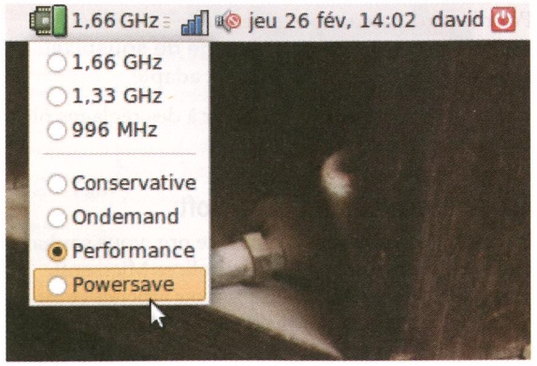
Arrière-plan permet de définir quelle couleur (ou transparence) utiliser :

Un peu moins lisible, mais plus discret et laissant apparaître le papier-peint.

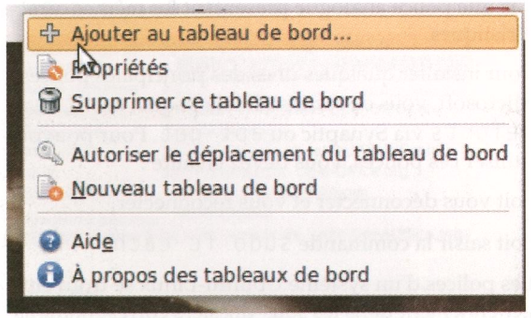
> Vous n'êtes pas limité à 2 tableaux de bord. N'hésitez pas à en ajouter ou en supprimer.

Ajouter au tableau de bord

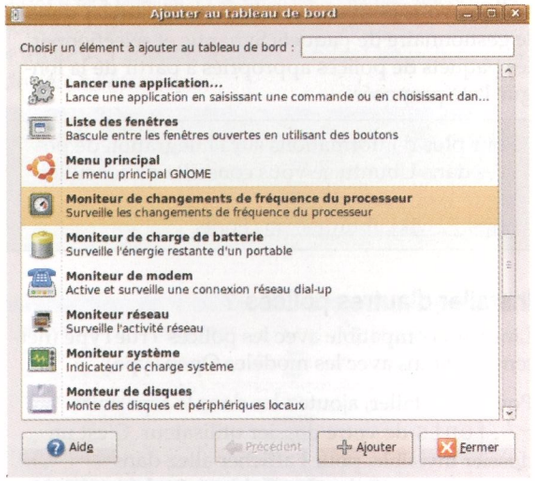
Les tableaux de bord ne servent pas qu'à afficher les boutons des fenêtres ou des raccourcis vers les applications. Gnome dispose de nombreux modules (ou widgets) que vous pouvez afficher : météo, finance, utilisation du processeur, etc. la liste est longue.



Ici le Widget **Moniteur de changement de fréquence** qui permet de savoir à quelle vitesse tourne le processeur. Pour l'afficher, faites un clic-droit sur une partie vide du tableau de bord et choisissez :



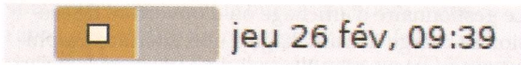
Faites dérouler la liste et choisissez le Widget :



Il ne reste qu'à le faire glisser sur le tableau de bord.

Bureaux virtuels

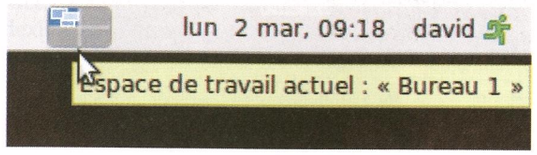
Gnome est livré avec deux Bureaux. Oui, même si vous n'avez qu'un seul écran. Trouves sur un des Tableaux de bord cette drôle d'icône :



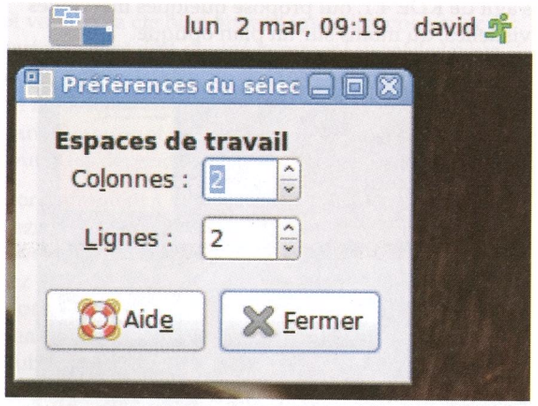
Elle indique qu'il y a deux bureaux virtuels. Le premier est actif, pas le second. Cliquez simplement dessus pour l'afficher. Il est vide.

> Les bureaux partagent tous le même papier-peint.

Sur mon poste j'ai créé quatre bureaux. Seul le premier est utilisé à ce moment :



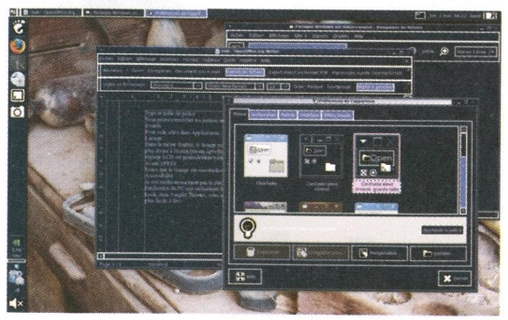
C'est très simple, faites un clic-droit sur le bureau virtuel et choisissez Préférences : définissez le nombre de lignes et de colonnes :



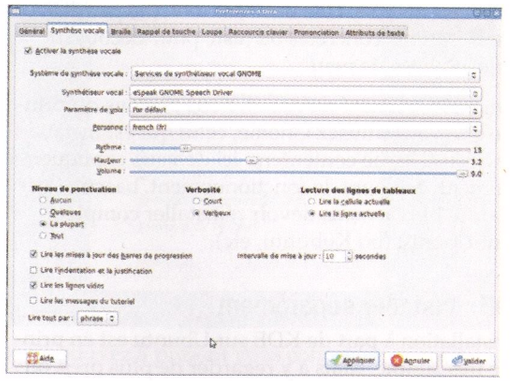
Accessibilité

Je n'ai malheureusement pas la place d'entre dans le détail, mais sachez que Linux dispose d'outils qui facilitent l'utilisation du PC aux utilisateurs handicapés.

Par exemple, au niveau de la personnalisation du Look, dans l'onglet Thèmes, vous trouvez des modèles à fort contraste, qui rende le contenu bien plus facile à lire:



D'autres outils d'accessibilités sont disponibles. Notamment le Lecteur d'écran Orca qui offre une synthèse vocale dans plusieurs langues (dont le français) qui lira à voix haute ce qui est affiché à l'écran, et une loupe pour grossir l'écran.

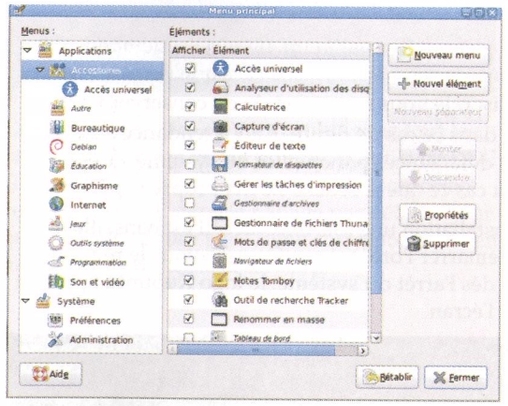


Vous pouvez régler la vitesse ainsi que la hauteur de la voix à voter goût (ainsi que pas mal d'autres paramètres) dans les préférences de Orca.

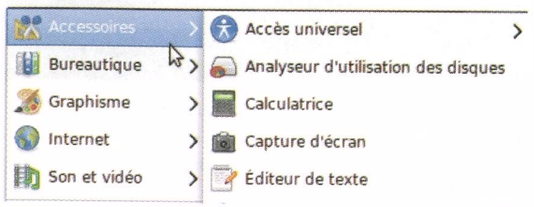
Personnaliser le Menu

Par défaut, on ne peut pas modifier le menu Ubuntu (Application, etc.). Mais c'est quand même possible.

Pour cela, il faut passer par Applications | Système | Préférences | Menu Principal:



Décochez ce que vous souhaitez masquer. Vous pouvez également faire glisser les menus et sous-menus pour les réorganiser. Ici, j'ai placé le menu **Accès universel** dans un sous-dossier de **Accessoires** :



KDE, Xcfe & Co.

Tous les goûts sont dans la nature ! Il n'est pas rare qu'un utilisateur ayant initialement utilisé GNOME souhaite ensuite changer de bureau, pour KDE ou Xcfe. Ou l'inverse. Une décision motivée par des considérations concrètes ou juste pour découvrir quelque chose de neuf.

Comme mentionné au début, nous partons du principe que vous utilisez Gnome, celui qui est installé par défaut. Je me contenterai ici de vous expliquer comment, en cours de fonctionnement, basculer sur un autre bureau sans devoir réinstaller complètement Ubuntu (ou Kubuntu, etc).

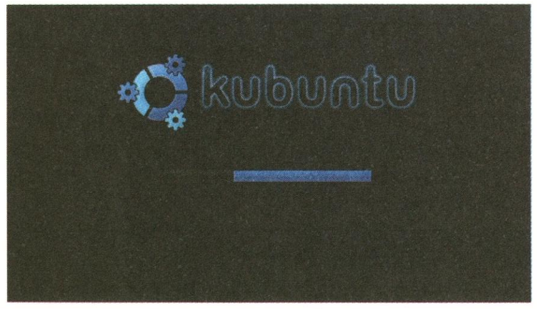
KDE : installer séparément

L'installation à part de KDE sur Ubuntu est en principe très simple. Mais à y regarder de plus près, on finit rapidement par tomber sur un os. Cela est surtout dû au fait que KDE et GNOME sont basés sur différentes bibliothèques et différents outils : GTK 2 pour GNOME et QT pour KDE. Chacun de ces environnements nécessite de nombreux paquets de support avec dépendances.

Mais passons aux actes ! Recherchez dans Synaptic le paquet `kubuntu-desktop` et sélectionnez-le pour installation. Vous voyez s'afficher une très longue liste de dépendances que je vous conseille de ne pas modifier. Vous pouvez ensuite démarrer l'installation.

Au cours de l'installation, il vous est demandé d'indiquer le *gestionnaire d'affichage* que vous souhaitez utiliser. Je vous recommande de conserver ici `gdm`, car dans ce cas de figure, `kdm` a tendance à générer des défaillances ponctuelles du système – c'est en tout cas ce que je retiens de ma propre expérience.

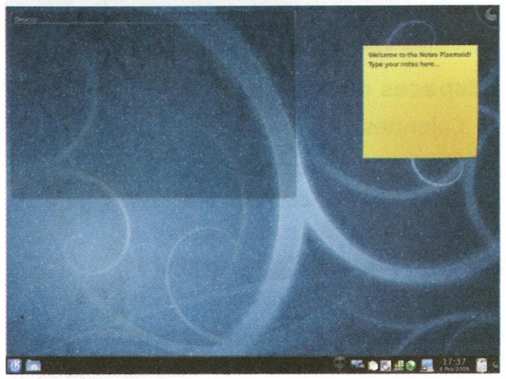
L'installation une fois effectuée, il est conseillé de redémarrer l'ordinateur. Comme vous le remarquerez dès l'arrêt du système, le logo Kubuntu apparaît sur l'écran.



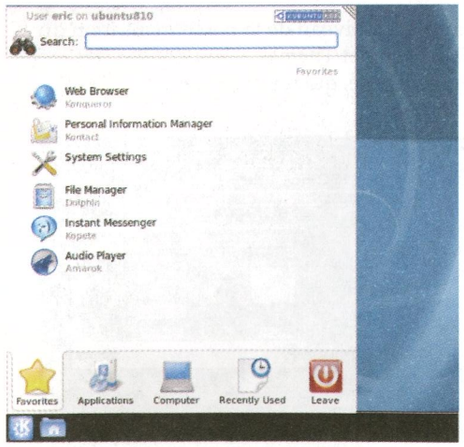
Le redémarrage une fois effectué, vous voyez d'abord s'afficher le gestionnaire d'affichage GNOME `gdm`, donnant l'impression que rien n'a été modifié.

Le gestionnaire d'affichage ou d'ouverture de session se charge de l'affichage d'une interface graphique permettant à l'utilisateur de se connecter au système. Cet écran vous offre également la possibilité de choisir la session à démarrer. Cliquez sur **Options**, puis sur **Sélectionner une session**. Dans la fenêtre qui s'affiche, sélectionnez **KDE** :

Une fois que vous avez saisi votre nom d'utilisateur et le mot de passe correspondant, votre première session KDE est démarrée. Sous Ubuntu 8.10, il s'agit de KDE 4.1, qui propose quelques merveilles visuelles, du moins sur un plan optique.



Le bouton K-Start vous permet d'ouvrir le menu de démarrage. **System Settings** vous permet de configurer les paramètres du bureau, comme par exemple la langue du pays et la résolution de l'écran. Vous pouvez également quitter la session, éteindre ou redémarrer l'ordinateur à l'aide de **Leave**.



Notez qu'en installant le bureau Kubuntu, vous n'avez pas installé simultanément les mêmes paquets que ceux qui l'auraient été avec un CD Kubuntu d'origine. Vous pouvez bien entendu disposer des mêmes fonctionnalités en effectuant quelques modifications, mais l'installation des paquets des versions spécialisées d'Ubuntu et Kubuntu est effectuée et optimisée en fonction du bureau standard correspondant.

Xfce

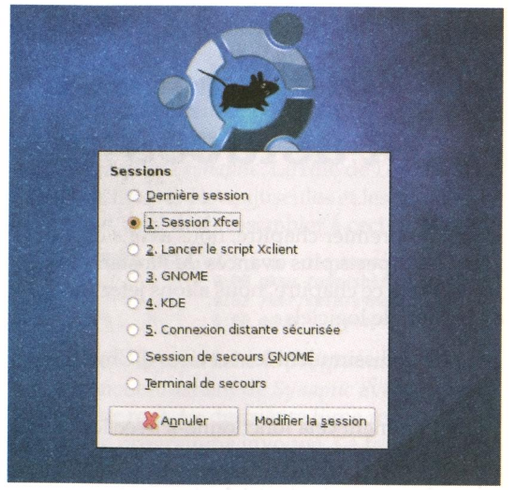
Xfce est un environnement de travail léger et très flexible nécessitant beaucoup moins de ressources que GNOME ou KDE. Il est donc particulièrement adapté aux anciens ordinateurs.

Tout comme GNOME, il se base sur la bibliothèque graphique GTK. Depuis Ubuntu 6.06 (Dapper Drake), Xfce est devenu un composant officiel d'Ubuntu, sous le nom Xubuntu. Dans la version 8.10, Ubuntu livre la version 4.6 de Xfce.

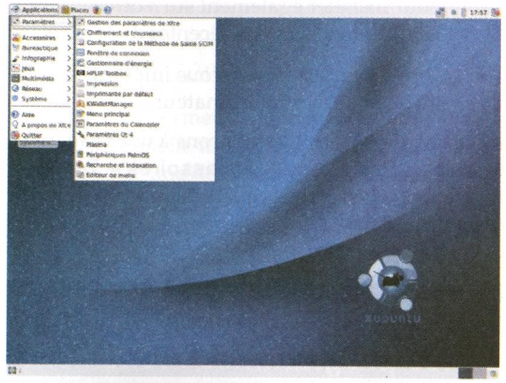
Si vous avez été bien attentif jusqu'ici, vous avez certainement déjà deviné comment procéder pour installer Xfce séparément... effectivement : il suffit de rechercher l'expression `xubuntu-desktop` dans Synaptic et d'installer ce paquet en incluant toutes les dépendances.

Contrairement à l'installation de KDE, vous ne devez pas sélectionner de gestionnaire d'affichage à ce niveau là. L'installation une fois effectuée, redémarrez votre ordinateur. Le gestionnaire d'affichage ou d'ouverture de session s'adapte au nouvel environnement de travail installé et se présente en tant que Xubuntu :

Attention : avant de pouvoir démarrer Xfce, vous devez d'abord le sélectionner via **Session**. Cliquez ensuite sur **Modifier la session**.



Vous pouvez à présent vous connecter en tant qu'utilisateur et admirer votre nouvel environnement de bureau :



Curieusement, Xfce est compatible avec tous les programmes et un grand nombre d'outils de KDE et GNOME. Et sur un plan visuel, il se comporte comme GNOME dans le réglage par défaut. Pour quitter Xfce, cliquez sur l'icône correspondante en haut à droite du tableau de bord.

Logiciels : trucs et astuces

Au cours du premier chapitre, nous avons déjà découvert des aspects plus avancés de l'installation de logiciels. Dans ce chapitre, nous allons jeter un oeil à une sélection de logiciels :

- Que dissimulent les services d'Ubuntu-Linux ?
- Comment cela fonctionne-il avec les pilotes propriétaires ?
- Quels sont les logiciels payants intéressants ?
- Comment utiliser les dépôts backports pour installer également sur votre système Ubuntu des versions récentes de logiciels ?
- Quels logiciels devez-vous impérativement installer sur votre ordinateur ?

Dans ce chapitre, nous travaillerons à nouveau avec la console (**Applications > Accessoires > Terminal**) que vous avez certainement depuis placée sur votre tableau de bord.

La fin des boîtes noires : quels sont les programmes réellement actifs sur mon PC ?

La commande `ps` vous présente la liste des processus sur votre ordinateur. L'option `a` vous permet non seulement d'afficher vos processus personnels, mais aussi l'ensemble des processus du système. Quant à l'option `x`, elle permet d'afficher également tous les processus qui ne sont pas liés à un terminal (tty). Une fois n'est pas coutume, ces options doivent être indiquées sans le tiret (-), cette commande provenant d'un autre dérivé d'UNIX nommé BSD.

Et c'est parti ! Tapez la commande `ps ax` sur la ligne de commandes. Vous voyez apparaître une liste relativement longue de processus actuellement actifs sur votre ordinateur :

```
eric@ubuntu810: ~
Fichier  Édition  Affichage  Terminal  Onglets  Aide
 5125 ?        Ss     0:00 /usr/bin/pulseaudio -D --log-target=syslog
 5126 ?        S      0:00 /usr/lib/pulseaudio/pulse/gconf-helper
 5128 ?        S      0:00 /usr/lib/libgconf2-4/gconfd-2
 5134 ?        Ss     0:00 /usr/bin/seahorse-agent --execute /usr/bin/gnome-sess
 5136 ?        S      0:00 /usr/lib/gvfs/gvfsd
 5143 ?        Ssl    0:00 /usr/lib/gvfs//gvfs-fuse-daemon /home/eric/.gvfs
 5147 ?        S      0:00 /usr/lib/gnome-session/helpers/gnome-keyring-daemon-w
 5152 ?        Ssl    0:00 /usr/lib/gnome-settings-daemon/gnome-settings-daemon
 5153 ?        S      0:01 /usr/bin/metacity --replace
 5178 ?        S      0:02 gnome-panel
 5181 ?        S      0:01 nautilus --no-desktop --browser
 5183 ?        Ssl    0:00 /usr/lib/bonobo-activation/bonobo-activation-server -
 5193 ?        Ss     0:00 gnome-screensaver
 5198 ?        S      0:00 /usr/lib/gvfs/gvfs-hal-volume-monitor
 5200 ?        S      0:00 /usr/lib/gvfs/gvfs-gphoto2-volume-monitor
 5203 ?        S      0:00 /usr/lib/gvfs/gvfsd-burn --spawner :1.4 /org/gtk/gvfs
 5206 ?        S      0:00 /usr/lib/gnome-applets/trashapplet --oaf-activate-iid
 5210 ?        S      0:00 /usr/lib/gvfs/gvfsd-trash --spawner :1.4 /org/gtk/gvf
 5216 ?        S      0:00 /usr/lib/fast-user-switch-applet/fast-user-switch-app
 5220 ?        S      0:00 /usr/lib/gnome-applets/mixer_applet2 --oaf-activate-i
 5227 ?        S      0:00 tracker-applet
 5228 ?        S      0:00 nm-applet --sm-disable
 5229 ?        S      0:00 bluetooth-applet
 5230 ?        S      0:00 update-notifier
 5232 ?        SNl    0:00 trackerd
 5234 ?        S      0:00 python /usr/share/system-config-printer/applet.py
 5235 ?        Sl     0:00 /usr/lib/evolution/2.24/evolution-alarm-notify
 5245 ?        Sl     0:01 /usr/bin/VBoxClient
 5273 ?        Ss     0:00 gnome-power-manager
 5313 ?        S      0:00 /usr/lib/notification-daemon/notification-daemon
 5335 ?        S      0:00 /usr/bin/perl /usr/share//system-tools-backends-2.0/s
 5365 ?        S      0:01 /usr/bin/python /usr/share/jockey/jockey-backend
 5409 ?        Rl     0:00 gnome-terminal
 5411 ?        S      0:00 gnome-pty-helper
 5412 pts/0    Rs     0:00 bash
 5429 pts/0    R+     0:00 ps ax
eric@ubuntu810:~$
```

Cette liste dépendant des programmes installés, elle sera naturellement légèrement différente chez vous. Nous allons maintenant nous intéresser à d'importants services.

Services ?

Les services, également nommés daemons (démons) dans le jargon technique des programmeurs, sont des programmes ou des processus fonctionnant continuellement en arrière-plan et proposant certains services. En règle générale, l'utilisateur n'utilise pas directement ces programmes : ils sont le plus souvent manoeuvrés par le biais d'autres programmes ou processus.

> On entend par **processus** tout programme fonctionnant dans la mémoire vive. Les processus peuvent faire partie intégrante d'applications mais également de services. Les applications comme les services peuvent invoquer plusieurs processus.

Un démon (nous allons à présent respecter la terminologie Linux) peut être lancé soit de manière classique, via un appel sur la ligne de commandes, soit par un autre processus, par exemple un démon, qui invoque alors ce que l'on appelle un processus enfant. Un démon invoque souvent ses propres processus enfants afin d'être disponible simultanément à différents endroits. C'est le cas en particulier lorsqu'un démon met à disposition des services réseau de type serveur que l'on sait très sollicités. Le serveur Web Apache invoque alors par exemple entre 5 et 30 processus enfants (selon la configuration) afin de répondre le plus rapidement et efficacement possible aux requêtes.

Init : la mère de tous les processus

Jetons un coup d'oeil à la liste des processus affichée par ps ax :

```
eric@ubuntu810:~$ ps ax
  PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:01 /sbin/init
    2 ?        S<     0:00 [kthreadd]
    3 ?        S<     0:00 [migration/0]
    4 ?        S<     0:00 [ksoftirqd/0]
```

Le premier processus affiché est `/sbin/init`, en abrégé : `init`. Il possède l'identifiant ID 1 et, sur les systèmes Linux, représente le premier processus lancé au démarrage du système. Il initie (d'où le nom) l'invocation d'autres processus.

Si vous souhaitez afficher la hiérarchie des processus sur votre système, il vous suffit de saisir la commande `pstree` :

```
eric@ubuntu810:~$ pstree
init─┬─NetworkManager
     ├─acpid
     ├─atd
     ├─avahi-daemon───avahi-daemon
     ├─bluetoothd
     ├─bonobo-activati───{bonobo-activati}
     ├─console-kit-dae───63*[{console-kit-dae}]
     ├─cron
     ├─cupsd
     ├─2*[dbus-daemon]
     ├─dbus-launch
     ├─dd
     ├─fast-user-switc
     ├─gconfd-2
     ├─gdm─┬─gdm─┬─Xorg
     │     │     └─x-session-manag─┬─bluetooth-apple
     │     │                       ├─evolution-alarm───{evolution-alarm}
     │     │                       ├─gnome-keyring-d
     │     │                       ├─gnome-panel
     │     │                       ├─metacity
     │     │                       ├─nautilus
     │     │                       ├─nm-applet
     │     │                       ├─python
     │     │                       ├─seahorse-agent
     │     │                       ├─tracker-applet
     │     │                       ├─trackerd───2*[{trackerd}]
     │     │                       ├─update-notifier
     │     │                       └─{x-session-manag}
     ├─6*[getty]
     ├─gksu───synaptic
```

Vous constatez une fois de plus que `init` est le premier de tous les processus. Il invoque une série de processus, à partir desquels d'autres processus sont invoqués. L'illustration montre très clairement que le gestionnaire de sessions gdm X invoque plusieurs processus.

Gérer les processus

Il arrive qu'un processus se bloque et ne fonctionne plus correctement. Une commande aux allures martiales, `kill`, permet d'arrêter un tel processus. Prenons un petit exemple.

Commencez par lancer le gestionnaire de paquets Synaptic. Si nécessaire, ouvrez à nouveau une fenêtre de terminal et tapez la commande suivante sur la ligne de commandes :

```
# ps ax | grep -i synaptic
```

(Rappel : vous ne devez pas saisir le signe dièse. Il sert uniquement à indiquer que vous êtes connecté en tant que root et que la commande sera exécutée sous cet utilisateur.)

Cette ligne de commandes affiche d'abord la liste complète des processus et utilise pipe (`|`) – que nous connaissons déjà – pour rediriger le résultat vers une commande nommée `grep`. Cette dernière effectue une recherche du mot indiqué juste après le grep, dans ce cas *synaptic*. Le rôle de l'option `-i` consiste à ignorer les majuscules et les minuscules. Le résultat final doit ressembler à ceci :

```
root@ubuntu810:~# ps ax | grep -i synaptic
 5306 ?        S      0:00 gksu /usr/sbin/synaptic
 5307 ?        Ss     0:02 /usr/sbin/synaptic
 5810 pts/0    R+     0:00 grep -i synaptic
```

La première ligne affiche le programme `gksu`. Il assure le fonctionnement de *Synaptic* avec les droits de l'utilisateur root. La deuxième ligne représente visiblement le programme invoqué. La troisième ligne se contente d'afficher une nouvelle fois la commande qui servait à générer cette liste de résultats.

Nous récupérons à partir de la première colonne l'identifiant du processus (5307) que nous utilisons à présent pour arrêter le programme. Ce que nous faisons avec la ligne suivante :

```
# kill -15 5307
```

L'option `-15` permet au processus indiqué d'obtenir ce que l'on appelle un signal TERM. Il obtient en quelque sorte une demande de la part du système l'invitant à s'auto-interrompre. Il s'agit de l'option par défaut, vous n'êtes donc pas obligé de l'indiquer.

Mais si un processus s'est bloqué, il peut éventuellement ne plus réagir à un signal TERM. Dans ce cas, nous devons tuer réellement le processus (devenu totalement inutile) en envoyant un signal KILL, par le biais duquel le système se charge lui-même de détruire le processus. La procédure est la suivante :

```
# kill -9 <ID du processus>
```

Pourquoi faire preuve de clémence ? C'est très simple : avant son arrêt, un processus interrompu avec TERM peut éventuellement nettoyer des fichiers et des processus temporaires. Avec KILL, le processus est interrompu immédiatement sans qu'il ait la possibilité d'effectuer une dernière tâche, ce qui génère des résidus de fichiers – donc : n'oubliez pas d'être aimable ;-)

> **En outre** : si vous quittez le processus père, les processus enfants seront automatiquement interrompus, eux aussi. Fort heureusement, les choses se passent différemment dans la vie réelle...

Quels sont les programmes exécutés sur mon ordinateur ?

Jetons un rapide coup d'oeil à d'importants services système afin de vous procurer un petit aperçu de la vie intérieure de votre ordinateur.

Processus du noyau

Si vous observez les premiers éléments de la liste obtenue avec `ps ax`, vous remarquerez qu'ils sont entourés de crochets.

```
root@ubuntu810:~# ps ax
  PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:01 /sbin/init
    2 ?        S<     0:00 [kthreadd]
    3 ?        S<     0:00 [migration/0]
    4 ?        S<     0:00 [ksoftirqd/0]
    5 ?        S<     0:00 [watchdog/0]
    6 ?        S<     0:00 [migration/1]
    7 ?        S<     0:00 [ksoftirqd/1]
    8 ?        S<     0:00 [watchdog/1]
    9 ?        S<     0:00 [events/0]
   10 ?        S<     0:00 [events/1]
   11 ?        S<     0:00 [khelper]
   51 ?        S<     0:00 [kintegrityd/0]
   52 ?        S<     0:00 [kintegrityd/1]
   54 ?        S<     0:00 [kblockd/0]
```

Ces crochets indiquent qu'il s'agit de processus du noyau. Ils sont gérés par le noyau dans une zone spéciale de la mémoire et sont séparés soigneusement de ce que l'on appelle l'espace utilisateur contenant les autres processus. La bonne nouvelle est que vous n'êtes pas concerné par les processus noyau et que vous n'avez pas non plus à les gérer – presque jamais !

Les processus Terminal

Un peu plus bas figurent les processus getty.

```
4112 tty4     Ss+    0:00 /sbin/getty 38400 tty4
4113 tty5     Ss+    0:00 /sbin/getty 38400 tty5
4118 tty2     Ss+    0:00 /sbin/getty 38400 tty2
4119 tty3     Ss+    0:00 /sbin/getty 38400 tty3
4120 tty6     Ss+    0:00 /sbin/getty 38400 tty6
```

Il s'agit des processus mettant à disposition les différentes consoles terminal pour Linux.

Dans chaque système Linux standard, vous pouvez ouvrir une console via la combinaison de touches **Ctrl+Alt+F2** jusqu'à **F6**, ce qui vous permet ensuite d'utiliser la ligne de commandes. À vrai dire, la touche **Ctrl** vous sert uniquement de complément pour sortir de l'interface utilisateur graphique, car rien que dans l'environnement de bureau, **Alt** est souvent déjà affectée à d'autres fonctions. Dans les consoles terminal, vous pouvez donc aussi utiliser **Alt+<Touche de fonction>**.

Par exemple pour quitter la fenêtre du terminal, vous pouvez taper **Alt+F7**.

> L'origine de la multiplicité des consoles terminal est historique : à l'époque, aucune interface utilisateur graphique n'était disponible pour UNIX et la majorité des utilisateurs préféraient de toute façon passer par la ligne de commandes. C'est d'ailleurs toujours le cas : la plupart des administrateurs UNIX et Linux préfèrent travailler avec la ligne de commandes plutôt qu'avec les outils graphiques des environnements de bureau. Nombreux sont les systèmes serveur sur lesquels aucune interface utilisateur graphique n'est installée ! Précisons que cette installation est souvent inutile, les serveurs Linux étant généralement administrés par SSH, un shell distant sécurisé.

Sous Ubuntu, vous n'aurez normalement pas à travailler via ces consoles terminal, l'environnement de bureau vous proposant aussi en parallèle non pas une mais de nombreuses fenêtres terminal.

Services de connexion

syslogd et *klogd* sont deux autres services importants chargés de la journalisation de certains événements. syslogd est le démon général pour la journalisation des événements système, tandis que klogd s'occupe des messages du noyau.

Les fichiers journal correspondants sont affichés sous `/var/log`. Le démon *klogd* enregistre dans le fichier dans `kern.log` et *syslogd* dans `messages` ou `syslog`.

> Il s'agit d'un outil précieux permettant d'éviter d'éventuels problèmes. Si quelque chose dysfonctionne sur votre système, vous pouvez voir sous `/var/log/` si l'un des fichiers journal contient un élément intéressant – Linux est réputé pour la qualité de sa journalisation.

Si vous voulez voir les messages de journalisation entrés lors du dernier démarrage du système, il vous suffit de saisir la commande `dmesg` ou d'afficher le fichier `dmesg`. Cela peut être très utile, en particulier lorsque vous voulez savoir par ex. si un appareil a été ou non reconnu par le noyau. Si vous recherchez par ex. votre carte réseau (`eth0`), cela peut se présenter par ex. sous la forme suivante. Vous remarquerez à nouveau la présence de `grep` :

```
eric@ubuntu810:~$ dmesg | grep -i eth0
[    4.523920] eth0: registered as PCnet/FAST III 79C973
[   34.036044] eth0: link up, 100Mbps, full-duplex
```

Par conséquent, cette carte a donc été identifiée et activée. Mais revenons-en à la liste des processus obtenue avec `ps ax`.

Communication réseau automatique

Un peu plus loin dans la liste figurent les deux lignes suivantes – notez que le nom de l'ordinateur doit certainement être différent chez vous :


```
4486 ?        Ss     0:00 avahi-daemon: running [ubuntu810.local]
4487 ?        Ss     0:00 avahi-daemon: chroot helper
```

Avahi est un programme permettant à un ordinateur Linux de communiquer sur le réseau sans qu'aucune configuration réseau manuelle ne soit nécessaire. Certaines configurations sont ici effectuées automatiquement. À l'heure actuelle, rares sont les programmes supportant *Avahi*.

Démons D-Bus et HAL

D-Bus est un canal de communication interne appelé Bus par l'intermédiaire duquel différentes applications peuvent communiquer entre elles. Associé à HAL (Hardware Abstraction Layer), il sert à la reconnaissance automatique de matériels. Les services se répartissent en différents sous-services :

```
4441 ?        Ss     0:00 /sbin/klogd -P /var/run/klogd/kmsg
4464 ?        Ss     0:00 /bin/dbus-daemon --system
4486 ?        Ss     0:00 avahi-daemon: running [ubuntu810.local]
4487 ?        Ss     0:00 avahi-daemon: chroot helper
4533 ?        Ss     0:00 /usr/sbin/cupsd
4597 ?        Ss     0:00 /usr/sbin/hald
4600 ?        Ssl    0:00 /usr/sbin/console-kit-daemon
4601 ?        S      0:00 hald-runner
4683 ?        S      0:00 hald-addon-input: Listening on /dev/input/event3 /dev/input/event1
4691 ?        S      0:00 hald-addon-acpi: listening on acpid socket /var/run/acpid.socket
4695 ?        S      0:00 hald-addon-storage: polling /dev/scd0 (every 2 sec)
```

Plus bas dans la liste des processus se trouvent d'autres processus appartenant à cette rubrique :

```
0:00 /usr/bin/dbus-launch --exit-with-session /usr/bin/pulse-session /usr/bin/seahorse-agent --execute
0:00 //bin/dbus-daemon --fork --print-pid 6 --print-address 9 --session
```

En outre : si vous souhaitez afficher une liste des processus avec la structure parent-enfant, vous pouvez aussi compléter la commande `ps` avec l'option `-f`. Dans l'exemple ci-dessus, la liste de résultats ressemblerait alors à ceci :

```
4597 ?        Ss     0:04 /usr/sbin/hald
4601 ?        S      0:00 \_ hald-runner
4683 ?        S      0:00     \_ hald-addon-input: Listening on /dev/input/event3 /dev/input/eventi
4691 ?        S      0:00     \_ hald-addon-acpi: listening on acpid socket /var/run/acpid.socket
4695 ?        S      0:00     \_ hald-addon-storage: polling /dev/scd0 (every 2 sec)
```

Comme vous pouvez le constater, le démon `hald` invoque d'autres sous-processus.

Le démon d'impression

Entre les enregistrements relatifs aux démons D-Bus et HAL figure une ligne `/usr/sbin/cupsd`. Il s'agit du démon d'impression. CUPS (Common Unix Printing System) est le système d'impression standard des ordinateurs Linux et fonctionne de la même façon sur quasiment tous les systèmes.

Communication réseau sans fil

Les lignes suivantes de la liste des processus sont placées sous le signe de la communication réseau :

```
4742 ?        Ss     0:00 /usr/sbin/bluetoothd
4806 ?        Ss     0:00 /usr/sbin/NetworkManager
4814 ?        S      0:00 /sbin/wpa_supplicant -u -f /var/log/wpa_supplicant.log
4816 ?        S      0:00 /usr/sbin/nm-system-settings --config /etc/NetworkManager/nm-system-settings.conf
```

La fonction de `bluetoothd` est relativement facile à comprendre – il gère les connexions Bluetooth ! Le rôle de `NetworkManager` est tout autre. Il vous permet d'accéder aux appareils réseau de votre système par un simple clic de souris sur l'icône correspondante du tableau de bord. Considérez les deux ordinateurs en haut du tableau de bord.

`WPA_Supplicant` est un composant logiciel responsable du cryptage WPA dans un réseau sans fil.

Le programme `nm-system-settings` aide le gestionnaire réseau à mettre à disposition des informations relatives à des appareils et des réseaux sans fil.

> Par ailleurs, les composants Wireless et Bluetooth fonctionnent même si vous n'avez pas installé de matériel correspondant.

Sound!

Sous Linux et plus précisément depuis le noyau 2.6 (soit depuis un certain temps déjà !), c'est ALSA (Advanced Linux Sound Architecture) qui est utilisé par défaut en tant qu'architecture sonore. Depuis peu, le démon `Pulseaudio` est chargé de tout ce qui a trait aux données audio du système. Cela permet à ALSA mais également à d'autres systèmes ou sources sonores d'envoyer des données en toute transparence. `Pulseaudio` utilise pour ainsi dire ces données en tant que couche d'abstraction au-dessus du matériel.

> Notez qu'il existe encore des cartes sons qui fonctionnent mal ou pas du tout. Malheureusement.

GFVS

Une autre nouveauté : le démon GFVS.

```
0:00 /usr/lib/gvfs/gvfs-hal-volume-monitor
0:00 /usr/lib/gvfs/gvfs-gphoto2-volume-monitor
0:00 /usr/lib/gvfs/gvfsd-burn --spawner :1.5 /org/gtk/gvfs/exec_spaw/0
0:00 /usr/lib/gnome-applets/trashapplet --oaf-activate-iid=OAFIID:GNOME_Panel_TrashApplet_Factory --oaf
0:00 /usr/lib/gvfs/gvfsd-trash --spawner :1.5 /org/gtk/gvfs/exec_spaw/1
```

Il s'agit ici d'un système de fichiers virtuel de GNOME qui permet l'accès aux fichiers par le biais de différents protocoles de communication réseau tels que par ex. SMB, SFTP, DAV ou NFS.

Mais il existe encore bien d'autres processus – qui varient selon les logiciels présents sur votre ordinateur. Cette brève incursion aura au moins eu le mérite de vous offrir un aperçu de la liste de processus de votre système, qui est de prime abord très opaque.

Administration des services

Allez dans **Système | Administration | Services**. Avant de pouvoir procéder à des modifications, vous devez vous authentifier : cliquez sur le bouton **Déverrouiller**.

Je vous conseille d'agrandir cette boîte de dialogue : cliquez sur l'un des bords à l'aide du bouton gauche de la souris et faites-le glisser tout en maintenant le bouton enfoncé.

Ici, vous pouvez voir les services qui sont démarrés. Plus il y a de services, plus cela consomme de ressources (et cela prend de temps au démarrage). Vous pouvez en supprimer certains.

De quels services ai-je besoin ?

Posons la question dans l'autre sens : de quels services n'avons-nous *pas* impérativement besoin ? Voici quelques réponses possibles :

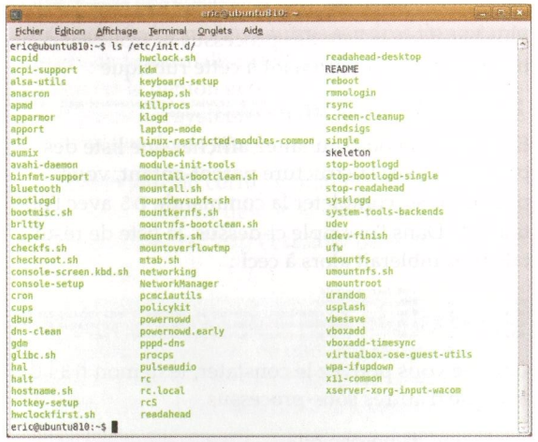
- **Gestion des périphériques bluetooth** (bluetooth) : nécessaire uniquement si vous souhaitez également utiliser des appareils Bluetooth.
- **Gestion de l'énergie** (acpid et apmd) : les deux sont surtout intéressants pour les ordinateurs portables, ils peuvent être désactivés pour les PC de bureau.
- **Découverte de service DNS multicast** (avahi-daemon) : comme mentionné précédemment, ce service est prévu pour la communication automatisée, mais il n'est actuellement compatible qu'avec peu d'applications. En outre, il est toujours préférable de configurer soi-même le réseau.
- **Support de rapports de défaillance automatiques** (apport) : ce service est également optionnel.
- **Gestion de touches spéciales** (hotkey-setup) : sert à déterminer les raccourcis clavier sur certains portables. Inutile pour les ordinateurs de bureau.

Selon l'installation, cette liste contiendra d'autres services. Certains sont présents par défaut, mais non activés. Vous pouvez et devez désactiver les services non utilisés, chaque service représentant également un danger potentiel au niveau de la sécurité. Mais attention, la désactivation de services peut aussi entraîner la perte de fonctionnalités nécessaires. Ne faites pas n'importe quoi.

Services : tenants et aboutissants

Vous avez peut-être à présent désactivé ou bien activé l'un ou l'autre service par le biais de la boîte de dialogue **Réglages des services** – mais que se cache-t-il réellement là derrière ?

Commençons par mentionner le dossier `/etc/init.d/` présent sur chaque système Linux. Il contient ce que l'on nomme des scripts d'initialisation pour les services disponibles sur votre système :



Vous pouvez invoquer vous-même une grande partie de ces scripts d'initialisation. La majorité des scripts comprennent les paramètres suivants :

start – démarre le service

stop – arrête le service

restart – redémarre le service (actif)

reload – relit les fichiers de configuration

status – indique l'état du service

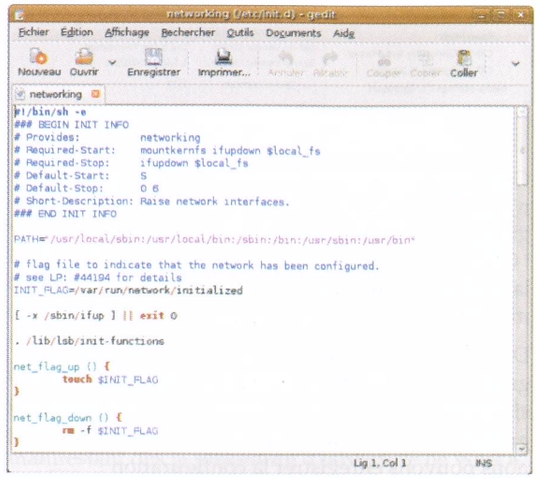
La ligne suivante vous permet donc d'arrêter le service réseau :

```
# /etc/init.d/networking stop
```

Suite à cela, vous ne pouvez plus communiquer par le biais de votre carte réseau. Invoquez à nouveau le script Init, cette fois-ci avec le paramètre `start` afin de réactiver le réseau.

Dans les coulisses, le script Init invoque la commande `/sbin/ifdown` ou `/sbin/ifup` afin de gérer les interfaces réseau.

Vous pouvez d'ailleurs aussi afficher les scripts d'initialisation à l'aide d'un éditeur de texte (par exemple *nano* ou *gedit*) ou d'un pagineur (par exemple *less*), il s'agit de scripts shell. L'illustration suivante présente un extrait du script mentionné ci-dessus :



Un peu plus loin dans le manuel, je vous expliquerai comment, avec peu de moyens, il est possible d'écrire ses propres scripts shell. Vous comprendrez alors – du moins dans les grandes lignes – la manière dont les scripts d'initialisation sont construits.

Mais qui donc invoque ces scripts d'initialisation au moment opportun ? Cette responsabilité incombe à la mère de tous les processus, `/sbin/init`. Tout comme le fichier de configuration `/etc/inittab`, les services appropriés sont invoqués au démarrage du système et interrompus à son arrêt.

C'est là le rôle de ce que l'on appelle les runlevels (que l'on peut traduire approximativement par « niveaux d'exécution »). Vous les trouverez sous `/etc:rc0.d`, `rc1.d`, ... jusqu'à `rc6.d` ainsi que `rcS.d`. Ils sont invoqués ou examinés de la même façon que le fameux runlevel dans lequel est localisé Linux. Ubuntu démarre par défaut dans runlevel 2. Considérons donc `rc2.d` :

Vous trouverez dans ce dossier une multitude de liens symboliques (Symlinks) commençant par S<numéro> et contenant ensuite le nom du service. S est l'initiale de Start (démarrage) tandis que les services à interrompre commencent par la lettre K (pour kill). Les numéros définissent l'ordre de traitement. Si l'on observe attentivement par ex. le lien `S28NetworkManager` (avec -l), ce dernier indique `/etc/init.d/NetworkManager` :

```
eric@ubuntu810:/etc$ ls -l rc2.d/S28NetworkManager
lrwxrwxrwx 1 root root 24 2008-12-21 20:51 rc2.d/S28NetworkManager -> ../init.d/NetworkManager
```

`Init` sait à présent que ce script doit être invoqué avec le paramètre `start`.

Tandis que les runlevel sont divisés avec plus de précision sous d'autres distributions, les runlevel 2 à 5 sont identiques sous Ubuntu et Debian. Sous openSUSE et Fedora, le runlevel standard est 5.

Par ailleurs, la commande `init <numéro du runlevel>` vous permet de changer de runlevel. Ainsi, `init 6` générera un redémarrage et `init 0` un arrêt du système. Pour lancer le mode de gestion du système, tapez `init 1`. Pour que tout cela fonctionne, vous devez être connecté en tant que root sur la console (par ex. via `sudo -s`).

Waouh, voilà déjà une multitude d'informations approfondies ! Revenons-en à des notions un peu plus accessibles.

Pilotes propriétaires

La gestion des pilotes des périphériques sous Linux est un beau casse-tête : la plupart des fabricants fournissent uniquement des pilotes Windows, de telle sorte que la communauté Open Source se trouve dans l'obligation de développer et de mettre à disposition elle-même les pilotes adéquats.

Il existe des pilotes Open Source pour quasiment tous les périphériques – même les cartes graphiques. Toutefois, de nombreuses fonctions ne sont pas supportées par les pilotes Open Source. La fonctionnalité 3D ainsi que la majorité des effets graphiques sont par ex. indisponibles.

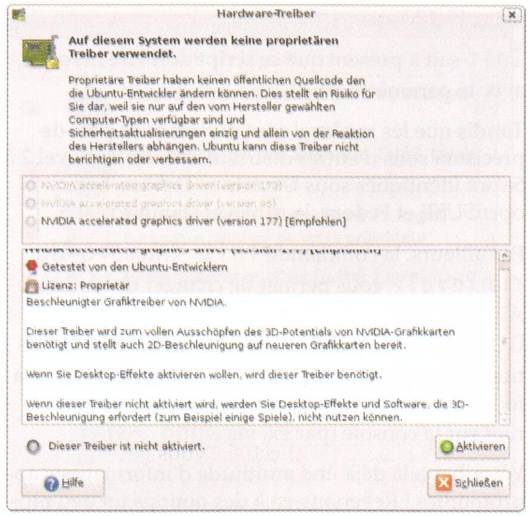
Pas d'inquiétude, vous allez obtenir de l'aide ! Nvidia et ATI proposent tous deux des pilotes propriétaires pour les systèmes Linux. Mais il ne s'agit pas de sources ouvertes, c'est-à-dire sans code source, et ils ne sont pas sous licence GPL, la licence publique GNU des logiciels gratuits.

Oui je sais, il ne faut jamais critiquer les cadeaux offerts, alors arrêtons de faire les difficiles – les pilotes mis à disposition nous permettent quand même de bénéficier de la fonctionnalité 3D et des effets optiques.

Installer des pilotes propriétaires

Si Ubuntu reconnaît une carte graphique pour laquelle un pilote propriétaire est disponible, vous voyez apparaître ceci :

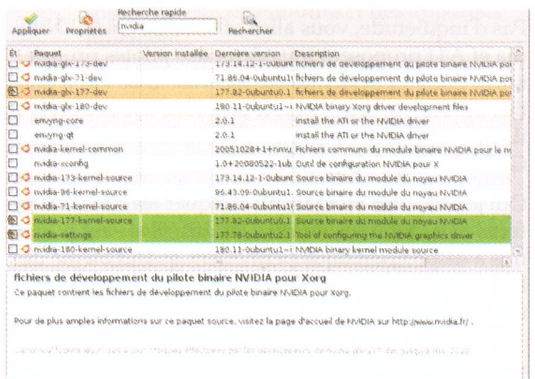
Cliquez sur l'icône pour afficher la boîte de dialogue correspondante.



Sélectionnez le pilote recommandé ou souhaité et cliquez sur **Activer**. Le pilote est téléchargé et activé, vous devrez certainement vous déconnecter puis vous reconnecter une fois pour pouvoir utiliser la fonctionnalité.

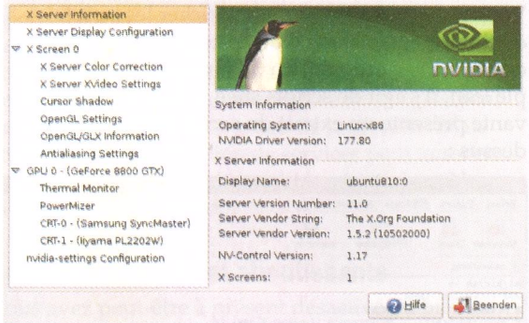
> Ces pilotes ne sont pas installés par défaut car ils ne sont pas libres. Leur code source n'est pas disponible ce qui va à l'encontre de la philosophie de Linux et des Logiciels Libres. C'est donc à l'utilisateur de décider s'il veut ou non utiliser ces logiciels.

Vous pouvez également passer par *Synaptic*. Recherchez le fabricant de votre carte graphique (par exemple Nvidia) et installez le bon pilote à partir de là :



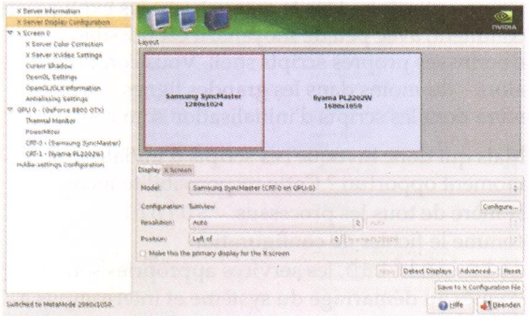
La suite de la procédure dépend du fabricant. Pour Nvidia, rendez-vous à présent sur la console et saisissez `sudo nvidia-xconfig` pour intégrer le pilote Nvidia.

La boîte de dialogue **Préférences Nvidia** se trouve dans **Système | Administration**.



Cela n'est malheureusement pas très utile car nous n'avons pas la possibilité d'enregistrer notre configuration (l'outil n'a pas accès aux privilèges de root) ! On peut bien entendu procéder aux réglages pour la séance en cours, mais ils seront perdus au moment de la déconnexion... La guigne ?

Détrompez-vous ! Nous pouvons invoquer le programme de gestion par le biais de la console en saisissant `sudo nvidia-settings`. Le programme démarre alors avec les privilèges root et nous pouvons enregistrer la configuration :

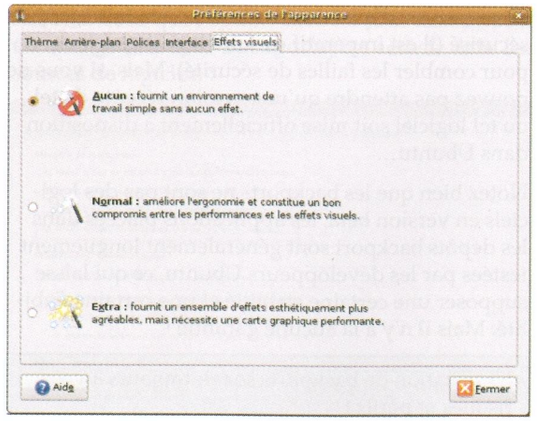


Le programme de gestion propriétaire propose une multitude de possibilités de configuration. Si vous avez deux écrans, ces derniers seront généralement reconnus automatiquement. Même la résolution est définie automatiquement. L'option **Position** vous permet de régler la position des écrans l'un par rapport à l'autre. Plusieurs tentatives sont généralement nécessaires avant l'obtention de l'affichage souhaité.

Vos réglages une fois terminés, n'oubliez pas de cliquer sur **Save to X Configuration File** afin d'enregistrer votre configuration et pouvoir la restaurer lors de la prochaine session.

Activer des effets

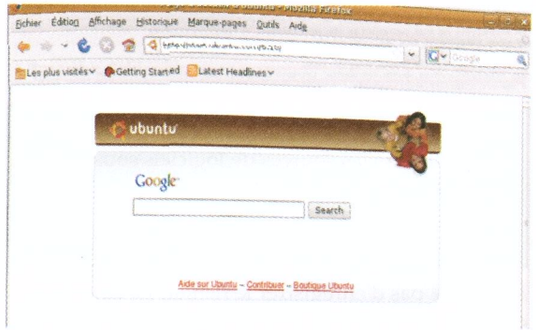
Dans le menu **Système | Préférences** vous retrouvez le menu **Apparence** que vous connaissez déjà. Allez dans l'onglet **Effets visuels** afin d'activer les effets par le biais de l'option *Normal* ou *Extra*.



L'ordinateur doit disposer d'une certaine configuration, capable de supporter les effets graphiques. Si votre machine est un peu ancienne, désactiver les effets lui donnera un peu de peps.

Cliquez sur **Fermer** pour fermer la boîte de dialogue, puis amusez-vous à ouvrir une fenêtre de navigateur avec Firefox. Un effet d'optique amusant se manifestant dès la procédure d'ouverture ne vous a certainement pas échappé : l'impression que la fenêtre saute en quelque sorte dans l'image.

Un effet encore plus amusant se produit lorsque vous essayez de faire glisser une fenêtre sur l'écran afin de la déplacer.



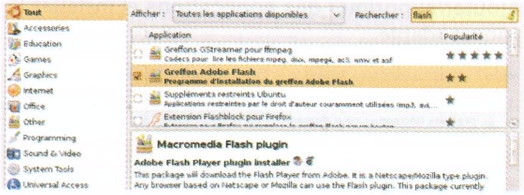
Si vous avez choisi l'option *Extra*, vous pouvez à présent admirer les caractéristiques d'une « fenêtre caoutchouc » se déformant en fonction des mouvements. Un effet d'optique sympa mais superflu dans le cadre d'une utilisation pratique.

Logiciels propriétaires intéressants

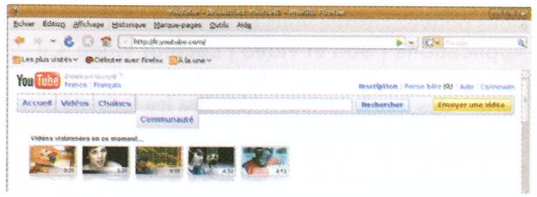
Par défaut, Ubuntu contient uniquement des logiciels placés sous une licence Open Source adaptée, généralement la GPL. Mais il existe certains programmes propriétaires populaires auxquels les utilisateurs Linux non plus ne veulent pas renoncer. En voici une sélection.

Plug-in Flash

Flash est pour ainsi dire le standard pour les films Internet et les animations. Pour pouvoir aussi profiter des vidéos Flash sous Linux via le navigateur Firefox, installez le plug-in Flash. Il vous suffit de vous rendre dans le menu **Applications**, sélectionner la commande **Ajouter/supprimer**, rechercher *Flash* et installer le paquet correspondant.



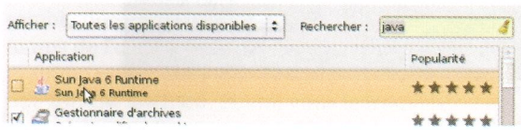
L'installation une fois effectuée, vous pouvez visionner des vidéos Flash dans le navigateur Firefox. Pour faire immédiatement un essai, vous pouvez vous rendre par ex. sur le célèbre site Web http://www.youtube.com. Les vidéos proposées sont au format Flash.



Il existe des lecteurs flash « libres », mais ils ne sont pas 100% compatibles avec les contenus flash.

Toutes les applications disponibles

Si vous ne trouvez pas les paquets, vérifiez que la liste déroulante Afficher est bien sur Toutes les applications disponibles :



Ubuntu ne me proposera Java que dans ces conditions.

Codecs vidéo et audio

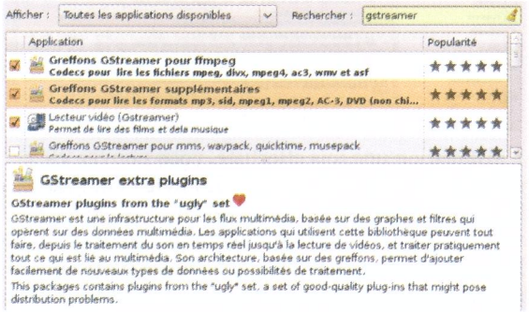
Le projet FFmpeg se compose d'une série de programmes et bibliothèques pour le traitement de matériels vidéo et audio. L'une des raisons de sa popularité est qu'il contient quelques codecs normalement protégés.

Un *codec* est un petit morceau de programme permettant la lecture d'un format vidéo ou audio. Il existe des pays dans lesquels certains codecs sont protégés par des brevets. C'est le cas en particulier aux États-Unis : certains codecs ne peuvent pas être utilisés librement et sont donc accessibles dans le dépôt *non-free* d'Ubuntu.

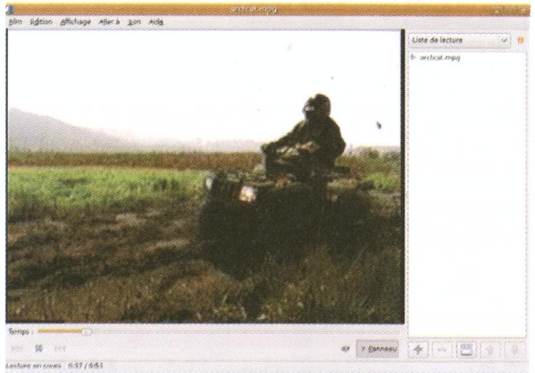
Pour pouvoir lire les formats audio et vidéo MP3, MPEG1, MPEG2, MPEG 4, DIVX et bien d'autres encore, installez les plug-ins pour *GStreamer*.

> *GStreamer* est un projet de *Freedesktop.org* conçu pour permettre l'utilisation de formats audio et vidéo indépendamment de la plateforme utilisée.

Recherchez *gstreamer* dans la boîte de dialogue **Ajouter/supprimer des applications** et installez le paquet adapté.



Vous pouvez ensuite vous rendre dans le menu **Applications | Son et vidéo | Lecteur vidéo**. Ce rôle est endossé par le programme **Totem**. Peu importe – dès lors le lecteur vidéo lira les formats mentionnés.



Backports

Backports ?

« Backporter » ou rétro-porter consiste à proposer pour une version plus ancienne d'un logiciel une modification initialement prévue pour une version plus récente.

Bien qu'encadrés par des développeurs Ubuntu expérimentés, les backports ne contiennent ni support ni mises à jour de l'équipe officielle d'Ubuntu. Vous êtes donc seul maître à bord. Pour la plupart des utilisateurs, il vaut donc mieux suivre les mises à jour de Ubuntu et, quand c'est nécessaire, changer carrément de version de Ubuntu. De cette façon, vous vous assurez d'avoir un ordinateur à la fois à jour et sécurisé (il est impératif de faire les mises à jour pour combler les failles de sécurité). Mais, si vous ne pouvez pas attendre qu'une version récente de tel ou tel logiciel soit mise officiellement à disposition dans Ubuntu…

Notez bien que les backports ne sont pas des logiciels en version beta, les applications placées dans les dépôts backport sont généralement longuement testées par les développeurs Ubuntu, ce qui laisse supposer une certaine stabilité et une certaine fiabilité. Mais il n'y a là aucune garantie !

> L'utilisation de backports se fait toujours à vos risques et périls !

Accéder aux backports

L'important est que vous ayez activé les dépôts *Updates* et *Security*, certains paquets du dépôt backports étant basés dessus. En outre, les composants *Main*, *Restricted*, *Universe* et *Multiverse* doivent être activés pour chaque dépôt.

Pour activer le dépôt backport officiel, ajoutez la ligne correspondante dans `/etc/apt/sources.list` ou retirez le marqueur de commentaire (#) devant les lignes concernées :

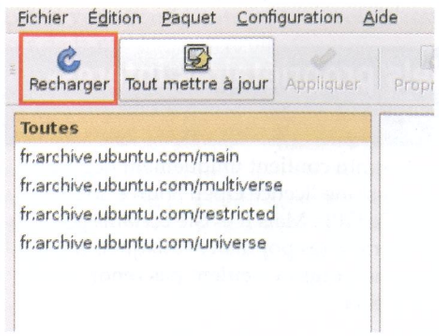
```
## Uncomment the following two lines to add software from the 'backports'
## repository.
## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
deb http://fr.archive.ubuntu.com/ubuntu/ intrepid-backports main restricted universe multiverse
# deb-src http://fr.archive.ubuntu.com/ubuntu/ intrepid-backports main restricted universe multiverse
```

> Vous pouvez également saisir cette ligne dans *Synaptic*, dans les sources de logiciels sous **Logiciels de tierces parties**. La procédure présentée ici est toutefois la plus simple car vous n'avez rien à taper.

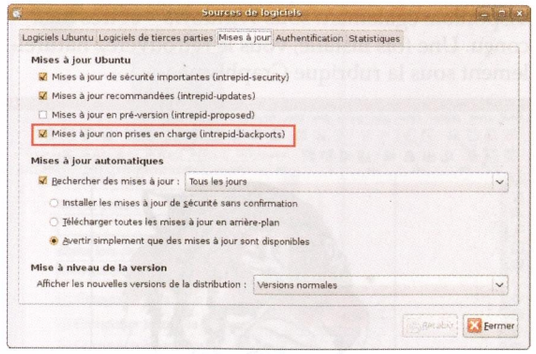
N'oubliez pas d'enregistrer le fichier. La liste de paquets du gestionnaire de paquets doit être ensuite actualisée, par exemple en tapant :

```
# apt-get update
```

Ou encore en chargeant à nouveau le cache des paquets via *Synaptic* :



Assurez-vous que l'option **Mises à jour non prises en charge (intrepid-backports)** est activée dans les sources de logiciels :

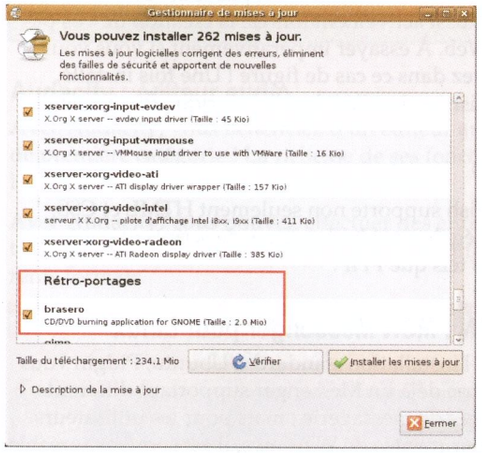


Les paquets backport sont ensuite prêts à être installés.

Notez que toutes les applications ne disposent pas de backports. Ainsi, à l'heure actuelle, bien qu'il soit sorti depuis plusieurs semaines, OpenOffice.org 3 n'est pas disponible pour Ubuntu 8.x. Il est possible de l'installer, mais vous devez le faire seul.

Installer des backports

Les backports sont traités comme des mises à jour :



Si vous souhaitez remplacer le paquet par le backport, il vous suffit de laisser l'encoche de la mise à jour. Pour tous les paquets dont vous souhaitez encore conserver la version actuellement installée sur le système, vous pouvez bloquer l'option correspondante pour chacun d'entre eux. Nous avons déjà étudié la procédure au début du manuel, dans le cadre des options des paquets.

De quels logiciels ai-je besoin ?

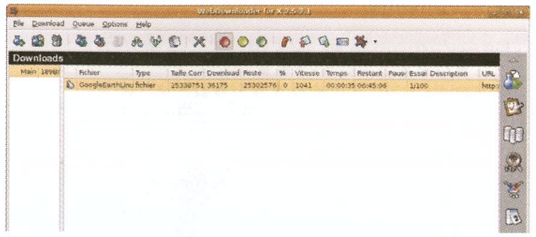
Ce chapitre vous semblera un peu dogmatique de prime abord. Je ne vous dresserai en aucun cas la liste des logiciels dont vous avez besoin ou non – les exigences auxquelles doit répondre l'environnement de travail sont multiples et propres à chacun.

Nous nous contenterons de jeter un oeil aux logiciels qui ne sont pas installés par défaut sur votre système mais susceptibles de vous intéresser. Faites votre choix !

D4X : gestionnaire de téléchargement

D4X fournit un gestionnaire de téléchargement certes moins pratique que le gestionnaire de Windows, mais qui peut se révéler très utile pour les gros téléchargements. Il est en mesure de limiter les taux de transferts et d'organiser les téléchargements qui peuvent être interrompus et poursuivis à tout moment.

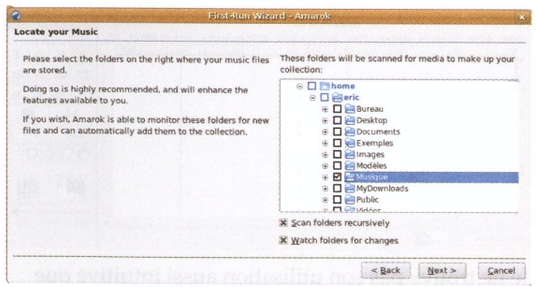
Recherchez D4X dans **Ajouter/supprimer des applications** et installez le programme. Il est ensuite disponible dans le menu **Applications**, sous-menu **Internet** :



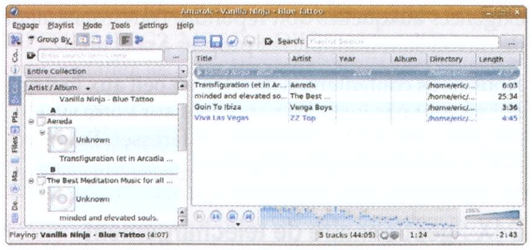
Dans le menu **File**, sous-menu **New Download** ou **Add a Download**, vous pouvez saisir une URL afin de démarrer un téléchargement.

Amarok : lecteur audio

Le lecteur audio *Amarok* fait partie des outils de prédilection des utilisateurs d'Ubuntu. Même s'il semble déconcertant au premier abord, prenez le temps de l'observer d'un peu plus près. Si le programme est installé, vous le trouverez dans la rubrique **Sound & Video**. Sinon, vous le trouverez dans la liste des applications sous **Ajouter/supprimer des applications**.



Au premier démarrage, un assistant vous aide à le configurer. *Amarok* peut non seulement lire vos fichiers audio mais également les organiser.

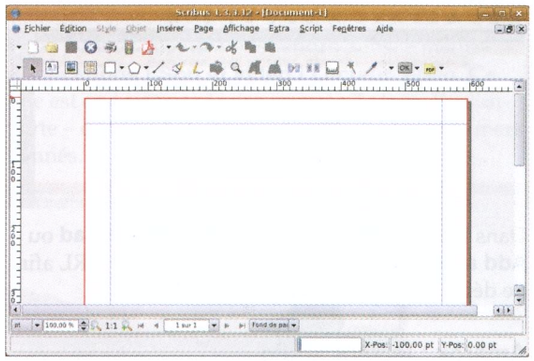


Si nécessaire, au moment de l'installation le support du format MP3 est installé.

Scribus : logiciel de PAO

Scribus est un programme de mise en page Open Source parfait pour les bulletins d'information, posters, brochures, etc. L'installation se déroule comme à l'accoutumée, en recherchant le nom du programme dans Ajouter/Supprimer.

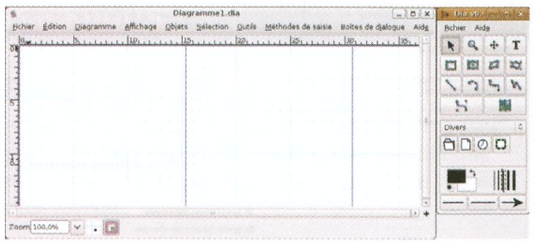
Vous trouverez ensuite le programme dans la rubrique **Graphisme**.



Ce programme performant ne laisse rien au hasard et son utilisation est très intuitive – bon amusement !

Dia : diagrammes et bien plus encore

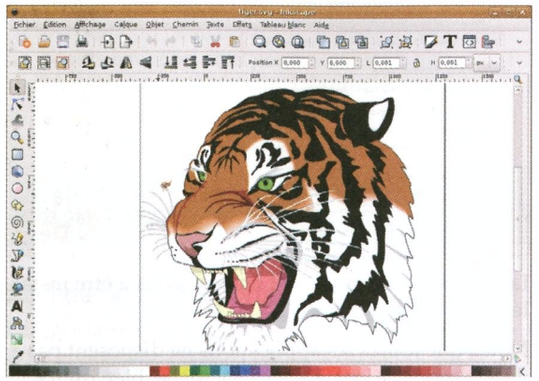
Dia vous permet de créer des diagrammes techniques ainsi que des structures UML. Installez-le comme vous en avez maintenant l'habitude et retrouvez-le dans la rubrique Graphisme.



Je ne trouve pas son utilisation aussi intuitive que celle de nombreux autres programmes ; mais avec un peu de patience, vous n'aurez aucun mal à le maîtriser.

Inkscape

Inkscape vous aide à créer et éditer des graphismes vectoriels. Il s'agit d'un programme très populaire – disponible également sous Windows – et très bien conçu. Une fois installé, vous le retrouverez naturellement sous la rubrique Graphisme :



Bluefish : création de pages Web

La majorité des éditeurs sous Linux – y compris les plus simples – supportent la coloration de la syntaxe. Cette aptitude est utile aussi bien au moment de la lecture que de la création de scripts ou documents HTML. Mais ce n'est pas tout : Bluefish vous aide aussi activement lors de la création de vos pages Web. À essayer impérativement si vous vous trouvez dans ce cas de figure ! Une fois installé, Bluefish est localisé dans la rubrique **Programmation**.

00000

Bluefish supporte non seulement HTML et CSS, mais également les langages de scripts orientés serveurs tels que PHP.

aMSN : MSN Messenger pour Linux

Dans l'installation standard d'Ubuntu, Pidgin vous propose déjà un Messenger supportant différents services de messagerie ; mais pour les utilisateurs inexpérimentés de Windows, il est souvent agréable que les caractéristiques spécifiques du logiciel soient conservées sous Linux. C'est pourquoi aMSN s'inspire du MSN Messenger de Windows. Sa popularité prouve qu'il répond aux goûts des utilisateurs.

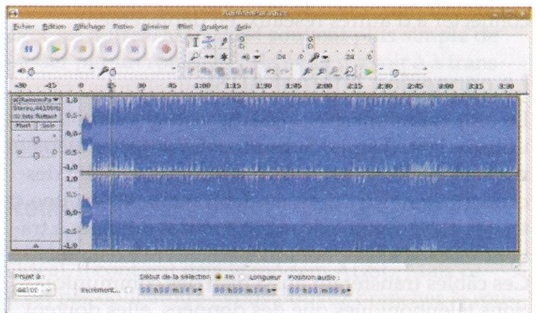
Le programme ne nécessite pas de longues explications, même pour les utilisateurs inexpérimentés de Windows – alors bon amusement sur les chats de discussion !

Skype est également disponible sous Linux.

Audacity : éditeur audio

Avec Audacity, vous bénéficiez d'un éditeur audio de première catégorie ! La richesse de ses fonctions lui permet de jouir d'une grande popularité.

Avec Audacity, vous pouvez effectuer des enregistrements, les éditer et les sauvegarder dans différents formats.



Il peut aussi gérer les fichiers MP3. En outre, il supporte de nombreux effets.

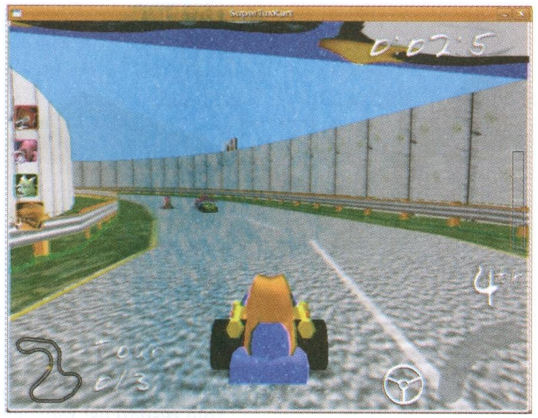
Pingus : place à l'amusement !

Connaissez-vous les Lemmings ? Cet ancien jeu Amiga du siècle dernier faisait l'objet d'un véritable culte dans les années 80 ! Ressuscitons-le l'espace de quelques instants ! Pingus est un clone gratuit de Lemmings.

Avec ce jeu, vous êtes assuré de passer un excellent moment de détente – laissez-vous envoûter par le charme des pingouins !

SuperTuxKart : à fond sur le champignon !

Vous avez fini par y prendre goût et recherchez à présent d'autres jeux divertissants sous Linux ? *SuperTuxKart* est un jeu de course très populaire.



Alors, montez à bord, mettez les gaz et à vous le podium !☺

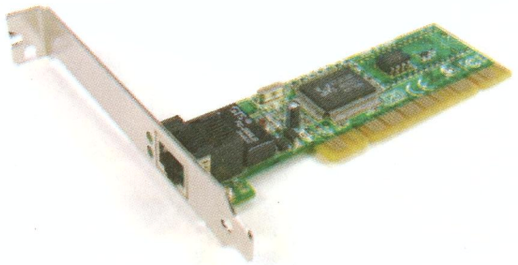
Réseau

Linux est surtout connu pour ses capacités réseau. Ubuntu ne fait pas exception. Aujourd'hui, il est de plus en plus rare de n'avoir qu'un seul ordinateur chez soi (ou au bureau) : plusieurs ordinateurs se partagent généralement une connexion Internet et les utilisateurs apprécient de pouvoir échanger des données entre eux.

Dans les pages suivantes, je vais vous expliquer comment tirer le meilleur des capacités réseau de votre Ubuntu et comment rechercher efficacement les problèmes afin de détecter les erreurs au niveau de la communication réseau.

Comment fonctionne un réseau ?

Actuellement, une carte réseau est intégrée dans quasiment tous les ordinateurs. Qu'il s'agisse d'une carte Ethernet ou WiFi :



Ici, une carte Ethernet. Cette carte contient un port de connexion RJ 45 auquel le câble réseau est raccordé. Il s'agit souvent d'un câble CAT 5.

L'autre extrémité de ce câble est généralement branchée à un commutateur (en anglais switch ou hub). Ces appareils disposent de plusieurs ports de connexion afin de permettre le raccordement d'autant d'ordinateurs ou d'autres composants réseau.

Il s'agit de la technologie la plus courante pour les réseaux locaux, également nommée **LAN** (initiales de *Local Area Network*). Dans la plupart des cas, le taux de transfert s'élève à 100 Mbit/s, (Fast Ethernet) mais atteint aujourd'hui de plus en plus fréquemment 1 Gbit/s (Gigabit-Ethernet).

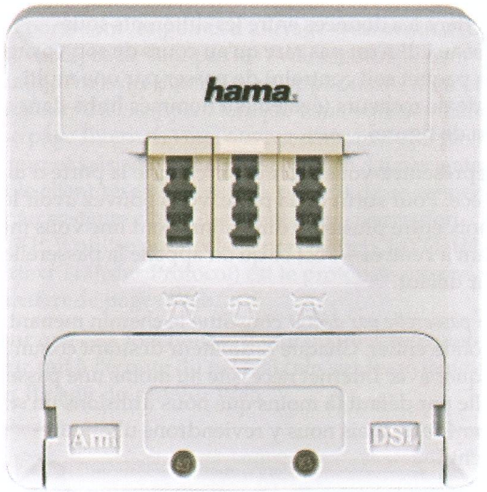
Mais les réseaux sans fil (WLAN, pour Wireless LAN : réseau local sans fil) ne sont pas loin derrière avec le WiFia ,b, g ou n. Ces lettres indiquent la vitesse du réseau , sa vitesse *théorique*, s'entend. C'est-à-dire celle annoncée sur la boîte, jamais celle que vous aurez réellement. La vitesse sera également impactée par la distance à couvrir, les murs et autres obstacles. Ainsi que par la présence de réseaux concurrents autour du vôtre.

L'ancien standard **802.11b** avec son débit maximal de 11 Mbit/s a pris depuis de la bouteille. C'est le **802.11g** avec un débit maximal de 54 Mbit/s qui est aujourd'hui sur le devant de la scène ; par ailleurs, un autre standard (**802.11n**) d'un débit maximal de 300 Mbit/s est déjà proposé (il est livré en standard sur tous les ordinateurs Macintosh, par exemple sur lesquels il est également possible d'installer Linux). Le WiFi est moins rapide que l'Ethernet, mais pour le partage d'une connexion Internet, cela suffira largement. La différence se fait uniquement sentir sur le transfert de gros fichiers qui prendront *beaucoup* plus de temps en WiFi que via un câble.

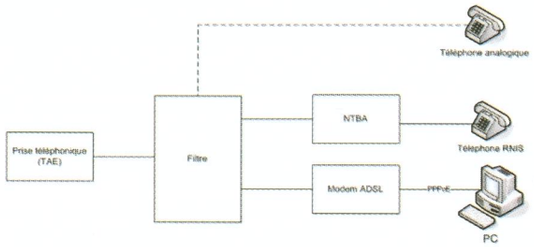
L'avantage du WiFi, c'est qu'on est débarrassé d'un fil sur l'ordinateur et que l'on peut facilement se connecter à d'autres réseaux, en déplacement.

La connexion Internet ADSL

La connexion Internet ADSL est actuellement la plus utilisée. DSL (Digital Subscriber Line), généralement nommé ADSL (A pour Asynchronous, le débit descendant étant supérieur au débit montant) est une technologie capable de transférer des données à travers les câbles téléphoniques en cuivre existants. Ces câbles transférant aussi bien des communications téléphoniques que des données, elles doivent transiter à des fréquences différentes. C'est là le rôle de ce que l'on appelle le **filtre (splitter)**, directement branché à la **prise téléphonique** (TAE).



Le modem ADSL sert de terminal ADSL. Un ordinateur est branché au **modem ADSL**, raccordé via Ethernet ou WiFi selon le cas. Cet ordinateur a besoin d'un logiciel pour pouvoir communiquer avec le modem ADSL via le protocole correspondant (**PPPoE**, Point-to-Point-Protokoll over Ethernet).



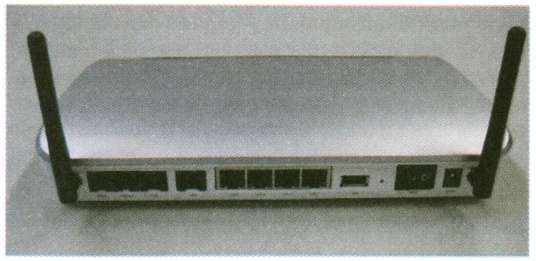
Dans ce cas, un seul ordinateur est raccordé directement à Internet – les autres ordinateurs doivent passer par cet ordinateur pour accéder à Internet. Cet ordinateur partage donc sa connexion Internet avec les autres. Mais il est souvent plus simple de passer par un routeur : cela évite d'avoir à à transformer un ordinateur qui devra rester allumé en permanence pour permettre aux autres de se connecter à Internet.

Routeur ADSL

Il existe différents types de routeurs ADSL. Dans l'idéal, il contient déjà le modem ADSL. Mais ce n'est pas obligatoire : vous pouvez brancher le routeur à votre modem.

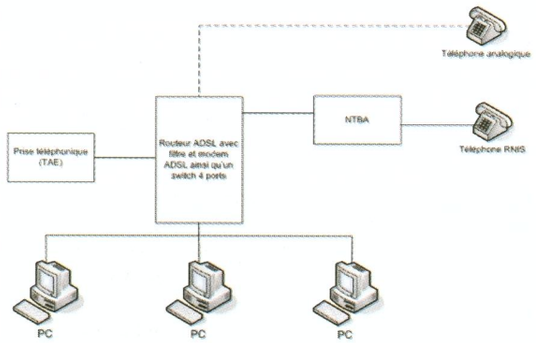
Dans tous les cas, il offre aux ordinateurs du réseau local la possibilité d'accéder à Internet directement et sans configuration logicielle supplémentaire. Il se comporte comme une *passerelle* tout à fait classique. Je reviendrai sur cette notion un peu plus tard, lorsque nous parlerons des routeurs.

Un routeur ADSL contient généralement un switch comportant jusqu'à quatre ports de raccordement pour ordinateurs locaux. En outre, il permettra aussi très souvent une connexion sans fil :



Notez qu'en France on utilise aussi beaucoup les « box » livrées par nos fournisseurs d'accès. La box est un modem, routeur, souvent aussi point d'accès sans fil et quelques autres choses en bonus : TV, téléphone, etc. Nous ne parlerons pas de tout cela. Nous allons nous concentrer sur le réseau et le partage de fichiers.

Un routeur ADSL simplifie considérablement la configuration d'une connexion Internet pour un réseau local. Si le routeur contient le filtre ADSL ainsi que le modem ADSL, le schéma de l'interconnexion ressemble alors à ceci :



Dans ce scénario, les PC n'ont besoin d'aucun logiciel ADSL, mais peuvent accéder tout à fait normalement à Internet via TCP/IP... normalement ? Quelle est l'importance de ce TCP/IP ?

TCP/IP ?

Si vous connaissez cela, n'hésitez pas à sauter cette section. Elle donne les bases du fonctionnement d'un réseau (et d'Internet).

IP – Le protocole Internet

Actuellement, la version 4 d'IP représente encore le protocole standard pour la communication sur Internet. La version IPv6 se trouve déjà dans les starting blocks, mais elle aura encore besoin d'un certain temps avant de s'imposer. C'est pourquoi nous consacrons notre brève introduction à IPv4.

IP est chargé du transport des données sur le réseau local et Internet. Ce sont les mécanismes d'IP qui

permettent l'envoi de paquets de données de la source jusqu'à la destination.

L'adresse IP

La base de toute communication : l'adresse de l'expéditeur et du destinataire – comme dans la vie réelle. Si vous appelez par son nom une personne à portée de voix, elle réagira. Si vous lui envoyez une lettre, elle lui sera délivrée grâce à son adresse postale.

C'est exactement la même chose avec les adresses IP. Il s'agit de quatre octets séparés par des points. Exemple d'une adresse valide : 217.14.245.16.

Chaque noeud de réseau (c'est ainsi que sont nommés les composants réseau actifs, tels que les ordinateurs, imprimantes ou routeurs) ne se contente pas d'avoir une adresse IP : il est également localisé dans un sous-réseau bien précis. Ce dernier est indiqué par ce que l'on appelle le masque de sous-réseau, et se compose lui aussi de quatre octets. Il peut se présenter par ex. sous la forme suivante : 255.255.255.0. Ces chiffres indiquent la partie de l'adresse IP propre au réseau et celle de ce que l'on appelle l'hôte. Exprimé plus simplement : le réseau est localisé au niveau des 255. Imaginons l'adresse IP 192.168.5.1 avec le masque de sous-réseau 255.255.255.0 : **192.168.5** désignerait la partie réseau, tandis que le dernier octet (ou byte en anglais), dans ce cas **1**, représenterait la partie propre à l'hôte ou l'adresse de l'hôte. Dans notre exemple, seul le dernier octet est disponible pour l'adressage de l'hôte, vous pouvez donc adresser aux hôtes les valeurs de 1 à 254 (255 étant réservé, voir plus bas).

Connaissances de base : un octet équivaut à 8 bits. Ces derniers peuvent prendre 2 valeurs : 0 ou 1. Chaque bit possède une certaine valence. Si le bit est défini (1), la valeur est ajoutée à la valeur totale. 255 est le nombre maximal que l'on peut obtenir avec 8 bits (tous les bits sont définis). Sous la forme binaire : 1111 1111. Un masque de sous-réseau contient généralement un certain nombre de bits définis en partant de la gauche. Pour 255.255.255.0, on obtient ceci : 1111 1111.1111 1111.1111 1111.0000 0000.

Qu'est-ce que cela veut dire ? Que signifient ici sous-réseau et hôte ? C'est très simple : tous les ordinateurs d'un même sous-réseau peuvent communiquer directement entre eux. Par contre, si deux interlocuteurs ne se trouvent pas dans le même sous-réseau, ils doivent être reliés entre eux par le biais d'un routeur. Les interlocuteurs d'un réseau sont appelés des *hôtes (hosts)*.

Routeur

Les routeurs sont en quelque sorte les panneaux indicateurs du réseau. Ils contiennent des paquets de données qui ne peuvent pas être transmis au niveau du réseau local (voir plus haut). Les routeurs transmettent les données entre les différents sous-réseaux. Il n'est pas rare qu'au cours de son voyage un paquet soit contraint de passer par une multitude de routeurs (également nommés **hubs** dans ce cas de figure).

Représentez-vous un routeur comme la porte d'une pièce. Pour sortir de la pièce, vous pouvez avoir le choix entre plusieurs directions, dont une vous menant à l'entrée – c'est ce qu'on appelle la passerelle par défaut.

La *passerelle par défaut* constitue le chemin menant au monde entier. Chaque ordinateur désirant communiquer avec Internet nécessite au moins une passerelle par défaut (à moins que nous utilisions un *serveur Proxy*, mais nous y reviendrons ultérieurement).

Un routeur fait toujours partie d'au moins deux réseaux, souvent davantage. Autrement dit, il dispose d'une adresse IP dans chaque réseau connecté.

Si un routeur réceptionne un paquet de données contenant une adresse de destination localisée en dehors du réseau local, il transmet le paquet. Pour ce faire, il consulte sa table de routage. Si le réseau dans lequel est localisée l'adresse de destination est relié directement au routeur, le paquet est livré directement. Dans le cas contraire, sa table de routage contient un enregistrement dont la signification est la suivante :

si tu reçois un paquet destiné à l'adresse X, envoie-le au routeur suivant avec l'adresse Y.

Le paquet est transmis ainsi sur Internet jusqu'à ce qu'il parvienne finalement à un routeur en contact avec le réseau de destination et en mesure de livrer le paquet.

D'ailleurs chaque ordinateur compatible TCP/IP contient une table de routage – pas uniquement les routeurs. Un peu plus loin, je vous expliquerai comment afficher la table de routage de votre ordinateur.

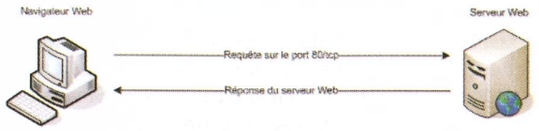
TCP et UDP

Maintenant que nous avons parlé de l'adressage et de la recherche du chemin sur le réseau, nous pouvons passer au niveau supérieur. C'est ici que se trouvent les protocoles de transport. TCP (Transmission Control Protocol) est le protocole de transport le plus important. Il est utilisé pour la plupart des transferts, car il crée une session avec le système cible, ce qui lui permet de contrôler et éventuellement corriger le transfert des données.

Outre TCP, on utilise également le protocole de transfert des données UDP (User Datagram Protocol). Il est plus rapide mais aussi moins fiable car il n'établit pas de session avec le système cible.

Des ports sont utilisés dans les deux cas de figure. Les ports sont les points d'ancrage d'un ordinateur. En réalité, ce ne sont pas les ordinateurs qui communiquent entre eux mais les applications d'un ordinateur. Un exemple : lorsque vous voulez afficher une page Internet, vous ouvrez d'abord votre navigateur et saisissez l'adresse souhaitée. Le navigateur est le client réseau. Il essaie à présent de se connecter au système cible. S'agissant d'une connexion HTTP, il utilise pour cela le port 80/tcp. HTTP (Hypertext Transfer Protocol) est le protocole propre au transfert de pages Web.

Pour que cela fonctionne, il faut qu'un service serveur soit à l'écoute du port 80/tcp sur le système cible. Il s'agit du serveur Web.



Il existe de nombreux services chargés de l'écoute de ce que l'on appelle les « ports reconnus » ou réservés (Well Known Ports). Ce sont les ports inférieurs à 1 024, qui incluent :

- FTP : 21/tcp
- SSH : 22/tcp
- Telnet : 23/tcp
- SMTP : 25/tcp
- DNS : 53/udp
- HTTP : 80/tcp
- POP3 : 110/tcp
- NetBIOS : 137/udp, 138/udp
- SMB : 139/tcp et 445/tcp
- IMAP : 143/tcp
- HTTPS : 443/tcp

La plupart de ces services ne vous disent certainement rien pour le moment, mais ce n'est que provisoire.

Le service serveur n'est pas le seul chargé de la surveillance d'un port – chaque client réseau dispose également d'un port qui lui est réservé. Ces ports commencent à 1 024 et sont attribués à la volée par le système d'exploitation, l'utilisateur n'a pas à intervenir.

DNS – Le Domain Name System

Lorsque vous saisissez une adresse dans le navigateur, cette dernière suit généralement le schéma suivant : www.xyz.com. Attendez un moment ! N'avions-nous pas défini que les ordinateurs communiquaient entre eux par l'intermédiaire d'adresses IP ?

Cette constatation est également vraie. Mais www.xyz.com étant beaucoup plus facile à retenir que par ex. 217.35.135.33, un service de traduction nommé DNS a été imaginé. Ce service convertit ce que l'on appelle le nom de domaine (www.xyz.com) en une adresse IP. Vous devez pour cela indiquer au client l'adresse IP d'un serveur DNS. Il s'agit généralement du routeur ADSL, car il sert en même temps de serveur DNS aux clients locaux. Le système DNS est assez complexe, mais il suffit ici de savoir que sous sa casquette de serveur DNS, le routeur ADSL peut retrouver le nom par le biais d'autres serveurs DNS sur Internet et renvoyer l'adresse IP correspondante au client demandeur. Le client (par ex. le navigateur Web) s'adresse ensuite réellement au serveur via l'adresse IP.

DHCP – Adresses IP dynamiques

DHCP (Dynamic Host Configuration Protocol) permet de transmettre automatiquement aux clients une configuration IP adaptée. Cette configuration est nommée **Lease** et inclut généralement les éléments suivants :

- Adresse IP
- Masque de sous-réseau
- Passerelle par défaut (facultatif)
- Serveur DNS (facultatif)

Le client est alors configuré pour utiliser DHCP, ce qui se fait généralement par le biais d'une simple case à cocher. Au démarrage du système, il demande alors une configuration IP en « hurlant » sur le réseau, ce que l'on nomme dans le jargon technique un *broadcast*(diffusion générale). Si un serveur DHCP se trouve à portée, ce dernier lui livre le Lease. Le client dispose alors d'une configuration IP valide et peut communiquer sur le réseau.

En règle générale, le routeur DSL endosse à nouveau ici la fonction du serveur DHCP.

Lors de l'installation d'Ubuntu, DHCP est activé par défaut de façon à ce qu'Ubuntu tente d'atteindre un serveur DHCP au démarrage du système.

ICMP

ICMP (Internet Control Message Protocol) sert au transfert de messages d'états et d'erreurs. Il est utilisé par ex. lorsque vous lancez une commande ping afin de déterminer si un interlocuteur est joignable sur le réseau. Nous reviendrons sur ce sujet un peu plus tard.

Vous avez maintenant appris une multitude de choses concernant le réseau – il est temps de passer à l'action en mettant ces connaissances en pratique ! C'est parti !

Configuration du réseau

Bonne nouvelle : dans la majorité des cas, Linux se débrouillera tout seul ; il n'y aura rien à configurer (sauf peut-être le clé du réseau sans fil à donner). Il trouvera le serveur DHCP, obtiendra une adresse IP et, si le réseau est connecté à Internet, pourra y accéder. C'est très agréable.

Sauf quand ça ne marche pas. Ce qui arrive, parfois.

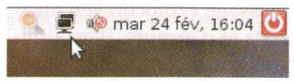
> Normalement, si vous vous connectez via Ethernet il ne devrait y avoir aucun souci.
>
> Avec Ubuntu, la quantité de cartes réseau *sans fil* reconnue semble également plutôt élevée. Tous mes PC fonctionnent sans aucun problème en réseau. En cas de carte non reconnue, la meilleure solution consiste à chercher de l'aide sur Internet… à partir d'un autre ordinateur, ou du même ordinateur, si vous pouvez le démarrer sous Windows .

Si vous essayez de vous connecter à un réseau non DHCP, voilà les informations dont vous aurez besoin :

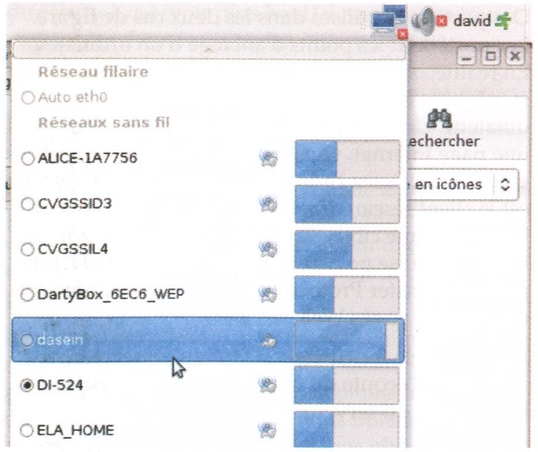
1. Adresse IP : elle doit être unique et faire partie du réseau local.
2. Masque de sous-réseau : le masque de sous-réseau correct nous permet de définir la partie réseau et hôte de l'adresse IP.
3. Passerelle par défaut : la porte donnant vers l'extérieur afin de pouvoir communiquer avec Internet.
4. Serveur DNS : le service de noms de domaine sans lequel la communication Internet ne peut se faire.

Se connecter à un réseau

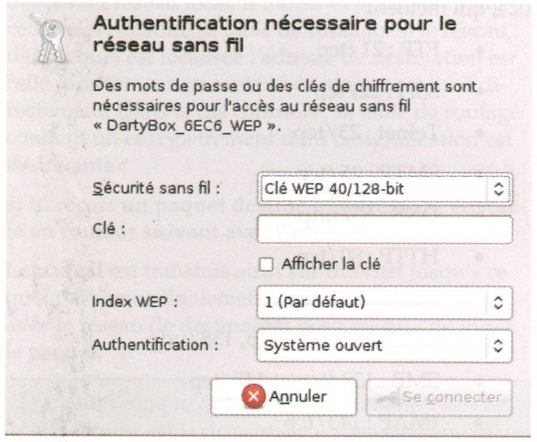
Si vous disposez d'un serveur DHCP sur votre réseau (un routeur ADSL, une box configurée en routeur, etc.), vous n'avez rien d'autre à faire pour une connexion Ethernet : Ubuntu récupère automatiquement une configuration IP à partir du serveur DHCP.



Cette icône m'indique que je suis connecté. Si elle affiche une croix rouge, cliquez dessus pour choisir soit votre réseau Ethernet, soit votre réseau sans fil :

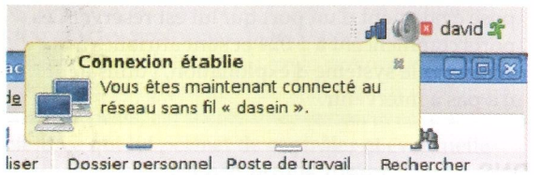


Dasein, c'est le nom de mon réseau sans fil. La longueur de la barre colorée indique la force du signal (et donc la qualité et la vitesse du réseau). Si votre réseau sans fil est un réseau sécurisé (et il devrait l'être sauf si vous désirez que tout le monde, *n'importe qui*, puisse s'y connecter et y faire ce qu'il veut sur Internet), Ubuntu vous demandera la clé :



Cliquez sur Se connecter et patientez le temps que la connexion s'établisse, la petite icône indique que l'outil est au travail :

Et voilà :

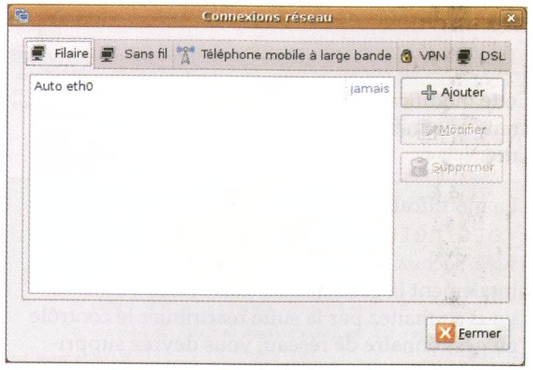


Gestionnaire réseau

La plupart du temps, c'est la seule chose dont vous aurez besoin. Le seul aspect du gestionnaire réseau dont vous devriez vous soucier.

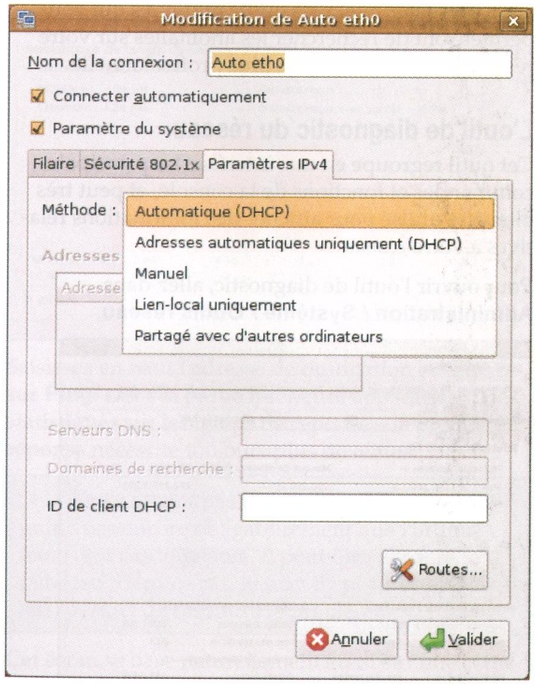
Si vous souhaitez adapter ces réglages, allez dans le menu **Système/Préférences/Network Configuration**.

Le *Gestionnaire réseau* permet de gérer vos connexions réseau. L'avantage ici est qu'il est également possible de gérer les réseaux sans fil, les connexions ADSL et VPN.

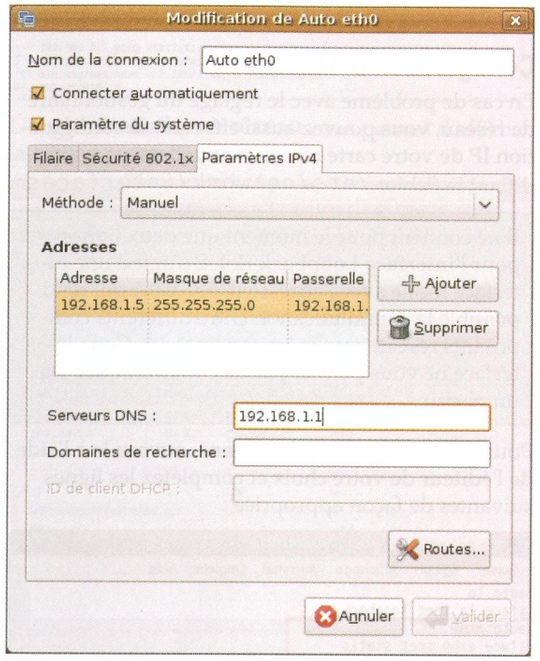


Considérons pour le moment la connexion avec fil. Tant que vous disposez d'un serveur DHCP dont la carte réseau se sert pour définir votre configuration IP, tout va bien. Mais gare à vous si vous intervenez manuellement ! Dans ce cas, le gestionnaire de réseau deviendra rapidement capricieux et non coopératif.

Le bouton **Modifier** vous permet d'adapter la carte réseau sélectionnée :

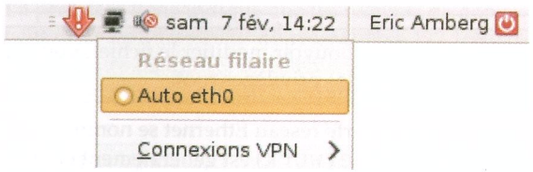


Vous pouvez par ex. définir si l'interface doit être configurée automatiquement (c-à-d. via DHCP) ou manuellement. Si vous avez choisi le mode manuel, vous pouvez saisir la configuration de l'adresse IP dans l'onglet **Paramètres IPv4** – pas très pratique mais possible. Cliquez pour cela sur **Ajouter** et saisissez les valeurs souhaitées.



Comme vous ne tarderez certainement pas à le constater, cette boîte de dialogue n'est pas encore très sophistiquée.

Vous devez ensuite réactiver l'interface configurée. Pour ce faire, cliquez sur l'**icône du gestionnaire de réseau** située sur le tableau de bord, puis cliquez sur l'interface :



Malheureusement, le gestionnaire de réseau ne conserve les modifications de la configuration que le temps de la session. Au redémarrage, tout est perdu – c'est très ennuyeux ! C'est également l'une des raisons pour lesquelles je ne suis pas encore vraiment convaincu par le gestionnaire de réseau.

Une autre raison est que, si vous modifiez directement un fichier de configuration de l'interface, le gestionnaire de réseau perd totalement pied. Mais si vous voulez réaliser une configuration réseau manuelle, qui doit être conservée même après un redémarrage, aucun autre choix ne s'offre à vous.

La documentation actuellement disponible sur le gestionnaire de réseau n'est malheureusement pas un modèle. Nous devons donc nous débrouiller tout seul pour le moment.

Configuration IP manuelle

En cas de problème avec le réglage du gestionnaire de réseau, vous pouvez aussi effectuer la configuration IP de votre carte réseau manuellement, en modifiant le fichier `/etc/network/interfaces`.

Il ne contient pour le moment que deux lignes pour l'**interface Loopback** (lo). Il s'agit d'une interface locale avec l'adresse IP 127.0.0.1 qui rend possible la communication entre différents composants réseau d'un même ordinateur. Cette interface ne vous permet pas de communiquer via un réseau.

Pour adapter le fichier mentionné, ouvrez-le à l'aide de l'éditeur de votre choix et complétez les lignes suivantes de façon appropriée :

```
root@ubuntu810: ~
Fichier Édition Affichage Terminal Onglets Aide
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
  address 192.168.1.5
  netmask 255.255.255.0
  network 192.168.1.0
  broadcast 192.168.1.255
  gateway 192.168.1.1
```

Notez que pour pouvoir modifier le fichier, vous devez disposer des privilèges root.

Votre première carte réseau Ethernet se nomme eth0. Le réseau (`network`) est généralement écrit sous la forme de zéros dans la partie propre à l'hôte. Dans le cadre du broadcast (diffusion générale), précédemment mentionné, il s'agit de la dernière adresse IP du réseau, dans ce cas 192.168.0.255.

Enregistrez les modifications et fermez l'éditeur. Vous devez ensuite redémarrer le service réseau. Saisissez donc la commande suivante :

```
# service networking restart
```

Si tout s'est bien déroulé, votre configuration IP est à présent enregistrée et appliquée.

Pas si vite ! Il manque encore quelque chose ! Vous n'avez pas saisi d'adresse IP pour le serveur DNS. Vous pouvez réparer cet oubli en adaptant le fichier `/etc/resolv.conf` :

```
eric@ubuntu810: /etc
Fichier Édition Affichage Terminal Onglets Aide
nameserver 10.0.2.3
```

Cette modification de la configuration est activée immédiatement, aucun redémarrage n'est nécessaire.

La modification manuelle du fichier `/etc/network/interfaces` une fois effectuée, sachez que le gestionnaire de réseau rejette totalement la gestion de la carte réseau **eth0**. Si vous souhaitez par la suite réattribuer le contrôle au gestionnaire de réseau, vous devrez supprimer du fichier nommé l'ensemble des enregistrements propres à **eth0**.

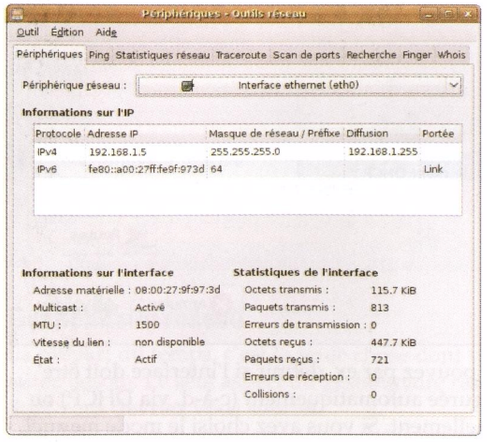
Vérifier la communication réseau

Si votre réseau est connecté à Intenret, lancez Firefox et affichez une page Web, par exemple le site Google. C'est un bon test « pas professionnel » pour savoir si ça marche. Dans les pages qui suivent, je vais vous présenter quelques outils pratiques qui vous permettront de rechercher les anomalies sur votre réseau de manière un peu plus professionnelle.

L'outil de diagnostic du réseau

Cet outil regroupe en une interface les principales commandes et fonctions de la console, et peut très bien être utilisé pour afficher des informations relatives à votre réseau.

Pour ouvrir l'outil de diagnostic, allez dans **Administration / Système / Outils réseau**.

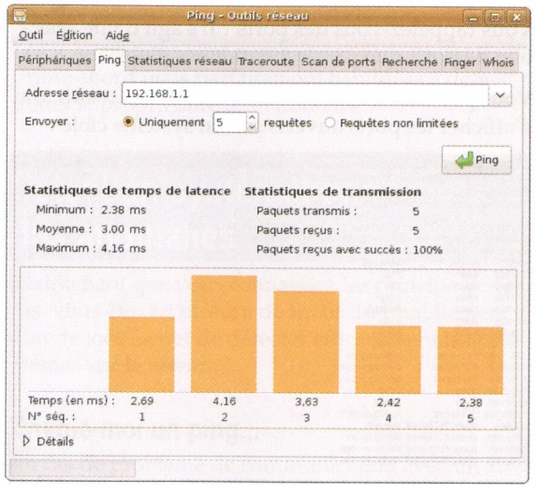


Dans l'onglet **Périphériques**, vous pouvez sélectionner le périphérique réseau (par exemple eth0, généralement votre carte Ethernet) afin d'afficher la configuration IP. Ignorez les lignes relatives à IPv6, elles ne sont normalement pas encore utiles.

Cet écran se base sur le résultat de la commande `ifconfig` que vous pouvez également saisir dans la console :

```
eric@ubuntu810: ~
Fichier Édition Affichage Terminal Onglets Aide
eric@ubuntu810:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:9f:97:3d
          inet adr:192.168.1.5  Bcast:192.168.1.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:fe9f:973d/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:648 erreurs:0 :0 overruns:0 frame:0
          TX packets:641 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:449645 (449.6 KB) Octets transmis:99549 (99.5 KB)
          Interruption:11 Adresse de base:0xc020

lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          Packets reçus:10 erreurs:0 :0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          Octets reçus:684 (684.0 B) Octets transmis:684 (684.0 B)

eric@ubuntu810:~$
```

L'onglet **Ping** vous permet d'envoyer un ping vers un ordinateur cible et ainsi de tester votre réseau et la présence de l'ordinateur cible. Un peu comme avec un sonar :



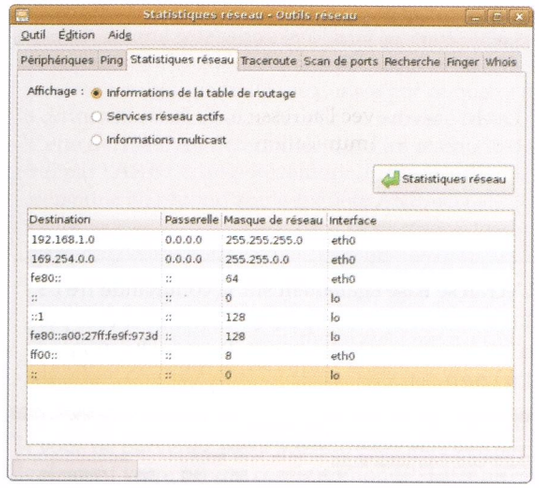
Saisissez en haut l'adresse de destination et cliquez sur **Ping**. Dans la partie inférieure apparaissent des statistiques sur les temps de réponse – la première réponse nécessite toujours plus de temps.

Si le *ping* n'atteint pas le système cible, cela ne signifie pas encore obligatoirement que l'ordinateur n'est pas joignable. Il peut-être derrière un pare-feu. Dans ce cas, le scan de ports (voir plus bas) permet de résoudre le problème.

Cet écran se base naturellement aussi sur une commande de console nommée `ping` : `ping -c 5` enverra cinq requêtes :

```
eric@ubuntu810: ~
Fichier Édition Affichage Terminal Onglets Aide
eric@ubuntu810:~$ ping -c5 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=150 time=2.22 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=150 time=3.38 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=150 time=4.31 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=150 time=3.15 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=150 time=4.53 ms

--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4031ms
rtt min/avg/max/mdev = 2.225/3.522/4.538/0.836 ms
eric@ubuntu810:~$
```

L'onglet **Statistiques réseau** vous permet d'afficher la table de routage :



La commande de console correspondante est `netstat -nr` :

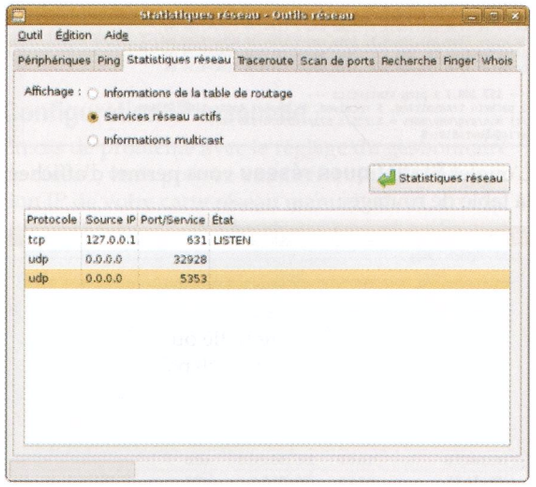
```
eric@ubuntu810: ~
Fichier Édition Affichage Terminal Onglets Aide
eric@ubuntu810:~$ netstat -nr
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic   MSS Fenêtre irtt Iface
192.168.1.0     0.0.0.0         255.255.255.0   U         0 0          0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 eth0
eric@ubuntu810:~$
```

La première ligne indique que les paquets de données pour les adresses IP du réseau cible 192.168.0.0 peuvent être distribués directement via eth0. C'est pour cette raison que 0.0.0.0. est affiché sous *Passerelle*.

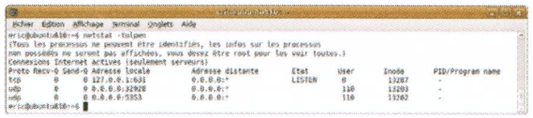
Vous êtes peut-être déconcerté par la deuxième ligne, car vous n'avez naturellement configuré aucune adresse IP sur le réseau 169.254.0.0 ?! Sous Windows, ce réseau est connu sous l'appellation APIPA (Automatic Private IP Adressing) et est conçu pour l'attribution automatique d'adresses IP lorsque aucun serveur DHCP n'est accessible. Il est affiché par défaut dans la table de routage même lorsqu'il n'est pas utilisé.

La dernière ligne indique que toutes les autres adresses cibles (0.0.0.0) sont envoyées à la passerelle par défaut 192.168.0.1. Cette ligne est particulièrement importante pour pouvoir déterminer si l'ordinateur sait quel chemin emprunter pour aller vers l'extérieur.

Les services réseau actifs vous présentent les ports reliés à votre ordinateur.

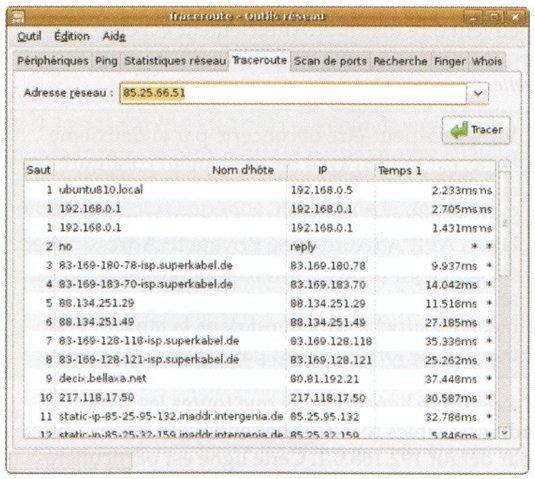


L'écran se base également sur la commande `netstat`. La console vous permet d'obtenir encore plus d'informations : il vous suffit d'ajouter l'option `-tulpen`.

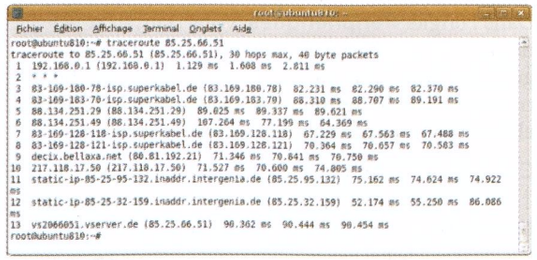


Vous découvrez également ici quel service se trouve derrière chaque port.

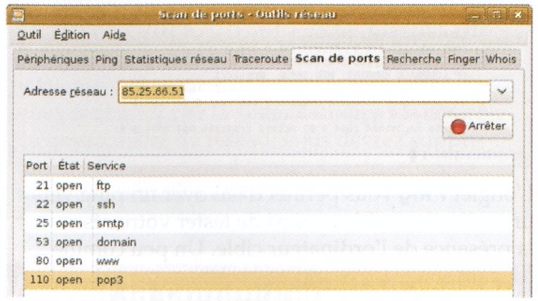
Avec **Traceroute**, vous pouvez afficher la route empruntée par le routeur (Hops) pour arriver à destination – vous voyez donc le chemin que prend un paquet de données pour parvenir à une certaine destination :



Vous pouvez obtenir ici le même résultat à l'aide de la commande de console de même nom :

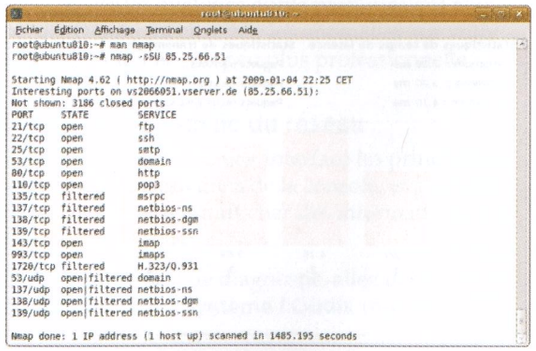


Allons voir du côté des pirates informatiques – avec l'onglet **Scan de ports**, vous pouvez lancer une analyse chargée de la vérification de tous les ports du système cible, afin de connaître les services disponibles :

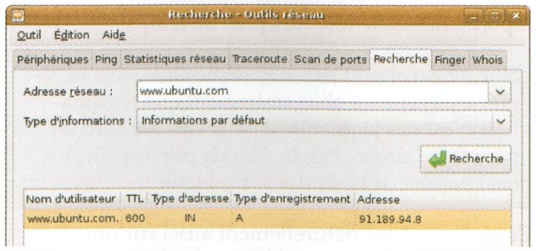


Vous rappelez-vous des ports ? Il s'agit d'une sélection de ports que j'avais mentionnés plus haut.

Vous pouvez utiliser sur la console l'outil `nmap` afin d'afficher les ports ouverts sur un système cible :

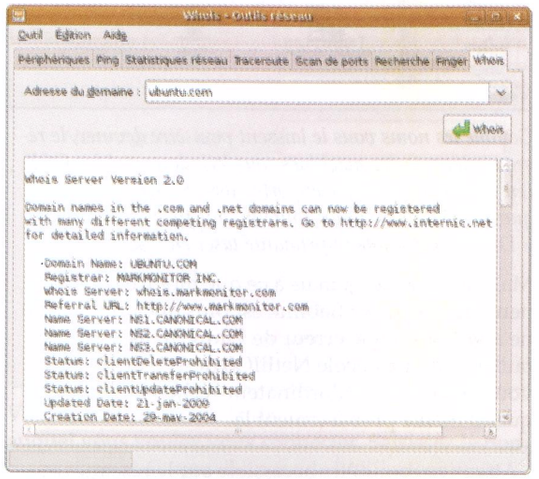


Sous l'onglet, **Recherche**, vous avez la possibilité de tester la résolution du nom :



Je vous fais grâce de l'onglet **Finger** : il s'agit d'un très ancien service UNIX qui ne fonctionne actuellement sur quasi aucun système – ne serait-ce que pour une question de sécurité, Finger permettant l'affichage des utilisateurs connectés à un système cible…

L'onglet **Whois** permet de questionner les bases de données Whois. Il s'agit des bases de données des registres Internet. Elles contiennent des informations de gestion relatives aux domaines enregistrés. Pour ce faire, il vous suffit de saisir le nom du domaine sans le nom de l'hôte et de cliquer sur **Whois** :



Les informations sont nombreuses et déroutantes de prime abord mais parfaitement habilitées à déterminer le propriétaire d'un domaine.

Détection systématique des anomalies

Maintenant que vous connaissez les principaux outils, vous êtes en mesure de les utiliser habilement afin de localiser et de détecter efficacement les problèmes sur le réseau.

Envoie-moi un ping...

En cas de problème de communication avec un autre ordinateur, commencez par vérifier si vous pouvez atteindre le système cible à l'aide d'un `ping`. Si cela s'avère possible, vous savez que la communication est en principe possible au niveau du réseau.

> Dans ce cas, l'erreur est localisée quelque part au niveau de l'application. Contrôlez si possible les deux côtés (client et serveur) afin d'y localiser l'erreur. Un pare-feu est également susceptible d'empêcher l'établissement de la communication.

Si vous n'atteignez pas la cible, soit vous n'obtenez aucun message, soit votre propre ordinateur vous envoie un message `Destination host Unreachable`.

Si vous n'obtenez aucun message en retour, cela signifie tout simplement que vous n'obtenez aucune réponse de la part du système cible. Deux explications possibles : la cible est inatteignable ou bien les paquets ICMP sont bloqués par un pare-feu.

Le message `Destination Host Unreachable` envoyé par son propre système indique que la cible est localisée dans le même sous-réseau mais ne donne aucune réponse. Concrètement, aucun système avec l'adresse cible indiquée n'est présent ici.

> **Qu'est-ce que cela signifie ?** Dans les coulisses, les ordinateurs ne passent pas par l'adresse IP pour communiquer entre eux, mais par ce que l'on appelle l'adresse matérielle ou **adresse MAC**. Cette dernière est déterminée dans un réseau local par l'**ARP** (Adress Resolution Protocol). Si personne ne répond à une requête ARP, cela signifie que le système cible n'est pas présent. Les pare-feu ne bloquent jamais les requêtes ARP, l'origine du problème réside donc ailleurs. En outre, vous pouvez afficher le cache ARP à l'aide de la commande `arp -a`. Si aucun système n'a répondu, il présente la résolution MAC comme étant `<incomplète>`.

Si vous ne parvenez pas à atteindre la cible, vérifiez également votre propre configuration IP. La configuration est correcte et vous ne parvenez même pas à atteindre votre routeur DSL ou votre passerelle par défaut ? Dans ce cas, vérifiez également le matériel, comme par exemple la carte réseau, les câbles, le routeur ADSL ou encore le switch.

La passerelle par défaut est-elle correcte ?

Deux tests très simples vous indiquent si la passerelle par défaut a été correctement définie :

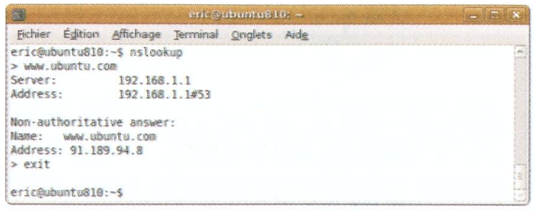
1. Saisissez `netstat -nr` et considérez l'enregistrement `0.0.0.0` dans la première colonne (voir ci-dessus). La colonne **Passerelle** contient-elle la bonne adresse IP ?
2. Effectuez un `Traceroute` afin de voir le chemin emprunté par le paquet pour atteindre la cible. **Attention** : les paquets ICMP sont souvent bloqués au niveau des routeurs, vous n'obtiendrez donc pas systématiquement de réponse – mais vous pouvez au moins voir si le paquet arrive sur la bonne passerelle.

Je ne parviens à invoquer aucune page Internet !

Vous n'arrivez à afficher aucune page Internet (par ex. http://www.ubuntu.com) avec le navigateur ? Commencez par effectuer les tests décrits précé-

demment afin de vous assurer que les fonctions de base du réseau sont opérationnelles.

Si c'est le cas, votre problème peut être dû au DNS. Commencez par tester si vous pouvez résoudre le nom de domaine de la page Web. Pour ce faire, vous avez besoin soit de l'onglet **Recherche** des outils réseau, soit de la commande `nslookup`. Elle contient son propre shell interactif dans lequel vous pouvez saisir vos requêtes :



Saisissez le nom du domaine recherché afin d'obtenir l'adresse IP correspondante. Si cette dernière n'est pas indiquée, cela signifie qu'il y a un problème avec la résolution du nom. Dans ce cas, vérifiez dans `/etc/resolv.conf` si la bonne adresse IP a été saisie pour le serveur DNS.

Pour quitter l'outil, saisissez `exit`.

Détection d'anomalies au niveau de l'application

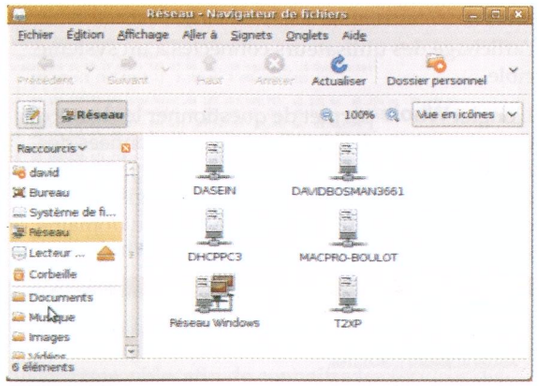
Tout fonctionne au niveau du réseau et la résolution DNS s'effectue correctement ? Le problème est alors provoqué soit par un pare-feu qui bloque la communication souhaitée, soit par l'application elle-même. Il convient de considérer ici aussi bien le client (par ex. sous la forme du navigateur Web) que le serveur (sous la forme du serveur Web). Il arrive souvent que vous ne puissiez contrôler qu'une seule page ; mais le message d'erreur affiché par le client indique au moins l'endroit où réside le problème.

Observons d'un peu plus près certaines applications réseau.

Accès aux partages Windows (et Mac)

L'accès à un partage Windows à partir de Linux ne pose plus de problème.

Dans le menu **Raccourcis**, sélectionnez l'outil *Réseau* afin d'afficher les groupes de travail Windows ou les domaines existants. Le nom par défaut le plus courant pour un groupe est WORKGROUP. C'est le cas ici et voici les ordinateurs qui y sont connectés :



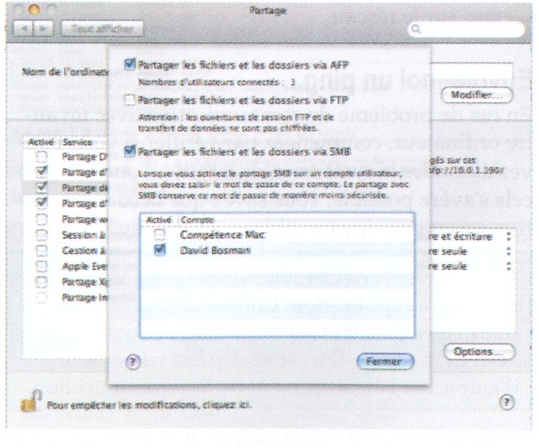
Comme les noms vous le laissent peut-être deviner, le réseau comporte des machines sous Windows et Mac OSX (en plus de Linux). Il comporte aussi d'autres périphériques : « Dasein » est un routeur sans fil et « DHCPPC3 » une imprimante laser en réseau.

Mais ne vous fiez jamais à ce que montre l'environnement réseau : sa fiabilité est plus que douteuse. Il ne s'agit pas d'une erreur de Linux, mais d'une défaillance du protocole NetBIOS de Windows. Si vous ne voyez pas l'ordinateur recherché, il est quand même probablement là.

> Attention, pour être accessible sur le réseau, vous devez activer un partage de dossier sur l'ordinateur Windows. Sinon, vous ne pourrez pas vous y connecter.

Réseau avec des Mac ?

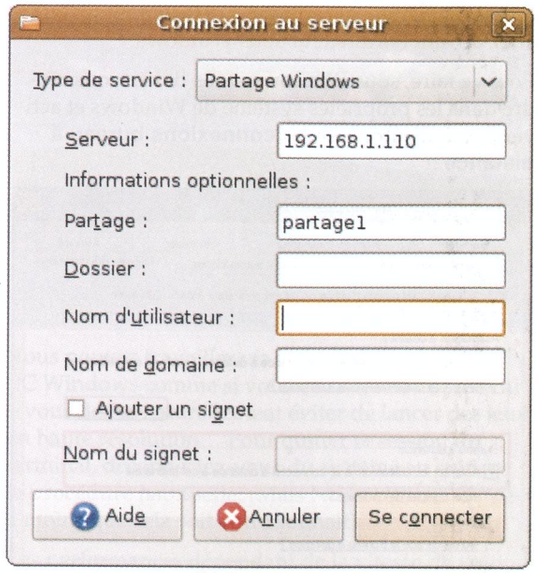
Pour que Ubuntu puisse se connecter à un Mac, il faut activer le partage Samba (SMB) sur ce dernier, dans les préférences système/Partage, cochez Partage de fichier et dans les Options, cochez Partage SMB, vous devez aussi définir les utilisateurs qui seront autorisés à se connecter :



L'ordinateur n'apparaît pas ? Essayez la connexion manuelle

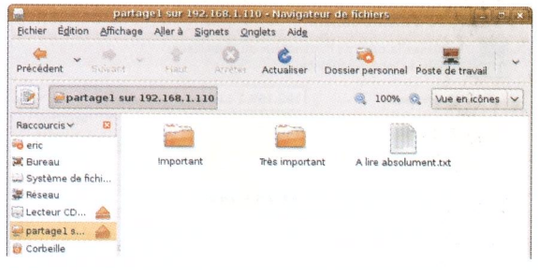
Si vous ne le voyez pas, essayez d'y accéder directement à l'ordinateur Windows.

Allez dans le menu **Raccourcis / Se connecter à un serveur**. Sélectionnez simplement le type de service *Partage Windows*.



Vous devez connaître l'adresse IP de l'ordinateur auquel vous voulez vous connecter. Saisissez le nom du partage et éventuellement de l'utilisateur et du domaine que vous voulez utiliser pour la connexion.

Pour établir la connexion, vous devez encore saisir le mot de passe correspondant qui vous est demandé ; le navigateur de fichiers *Nautilus* s'ouvre immédiatement et vous présente le contenu du partage :

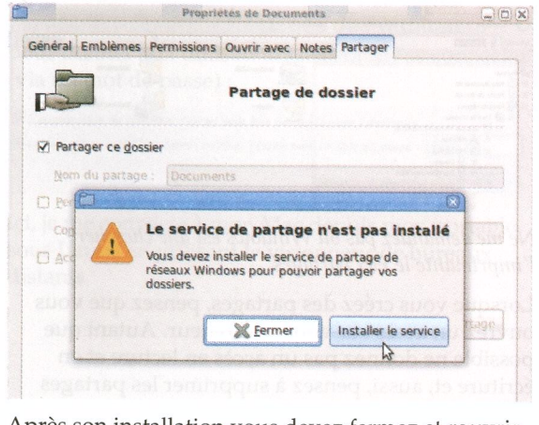


Selon les droits dont vous disposez, vous pouvez à présent lire, créer, modifier ou supprimer des objets du partage – comme vous le feriez sur un ordinateur Windows classique.

> NDLE : **ça ne marche pas ?**
>
> Linux voit l'autre PC sur le réseau, mais refuse de se connecter. Malgré que vous donniez le bon nom d'utilisateur et le mot de passe, la connexion est refusée ?
>
> Les réseaux locaux sont parfois très étonnants, une technologie qui semble parfois remonter à une autre époque, quand il fallait parfois prier les dieux pour que ça marche… pour que la chasse soit bonne et qu'on ramène un steak de mammouth à la caverne. Je caricature, et les choses se sont grandement améliorées, mais n'empêche que quand ça ne marche pas *on se sent bien seul devant notre écran.*
>
> Je n'ai qu'un seul conseil : si vous avez *tout* vérifié (cartes, pilotes, câbles (un câble endommagé peut poser problème), signal radio, activation du partage,…), si même redémarrer le poste sous Windows et celui sous Linux ne change rien : *recommencez*. Patientez éventuellement une petite heure, la magie (des dieux du réseau ?) se produira peut-être.
>
> Et si cela ne suffit pas ? J'ai toujours une clé USB sur moi.

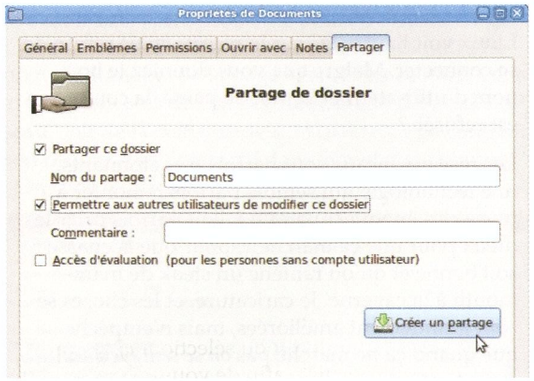
Accéder à Linux depuis un PC ou un Mac ?

C'est également possible, mais il faut installer Samba sur Linux. Le plus simple est de faire un clic droit sur un dossier que vous voulez partager et d'afficher ses Propriétés. Affichez ensuite l'onglet Partager et cochez Partager ce dossier. Une fenêtre vous invite à installer le service de partage (c'est Samba) :



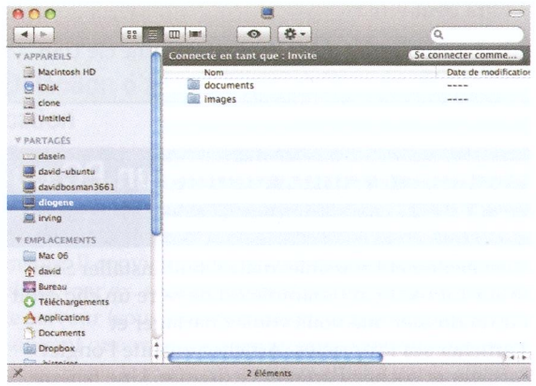
Après son installation vous devez fermez et rouvrir votre session d'utilisateur. Faites ensuite à nouveau

un clic droit sur le dossier à partager et cochez Partager ce dossier :

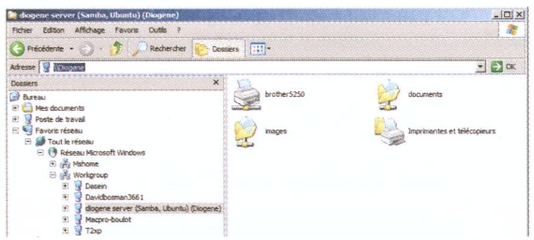


Paer défaut, les utilisateurs peuvent uniquement afficher le contenu du dossier, ils ne peuvent rien changer. Pour leur en donner le droit, cochez l'option **Permettre aux autres utilisateurs de modifier ce dossier**.

Votre ordinateur sous Linux est à présent accessible depuis le réseau, ici depuis un Mac :



Ici, sous Windows :



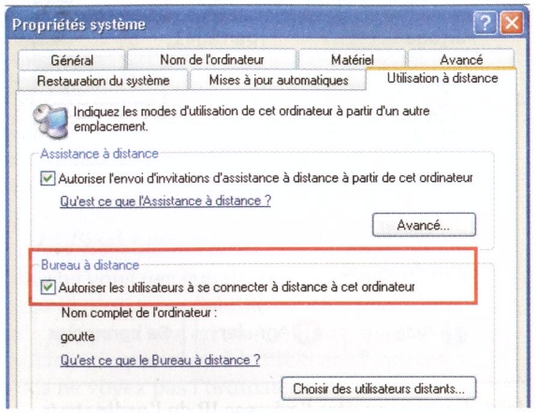
Ne me demandez pas où Windows est allé chercher l'imprimante le Fax partagés…

Lorsque vous créez des partages, pensez que vous ouvrez un accès sur votre ordinateur. Autant que possible ne donnez pas un accès en lecture et en écriture et, aussi, pensez à supprimer les partages dès qu'ils ne servent plus à rien.

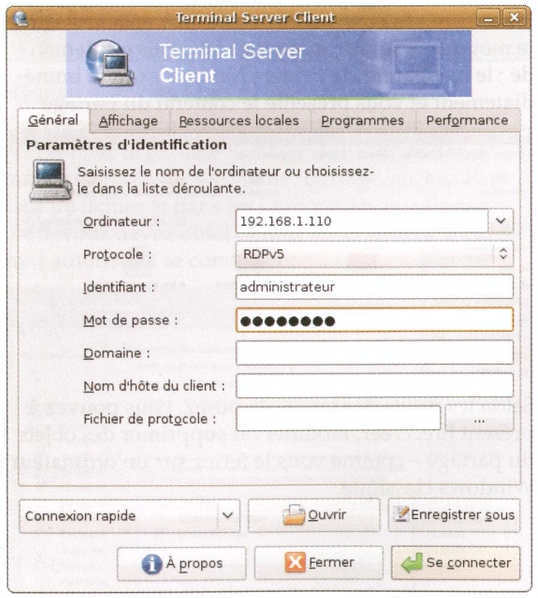
Bureau à distance

Pouvoir utiliser un ordinateur Windows à distance est particulièrement pratique. C'est comme si vous étiez devant l'ordinateur sous Windows.

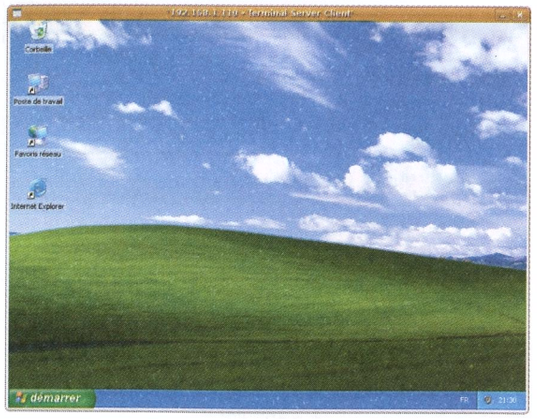
Pour ce faire, sous Windows, vous devez vous rendre dans les propriétés système de Windows et activer l'option autorisant les **connexions bureau à distance** :



Sous Linux, vous pouvez ensuite invoquer le **Terminal Server Client** à partir du menu **Applications**, sous-menu **Internet**.



Vous vous trouvez alors dans la fenêtre du Terminal Server Client sur le PC Windows :



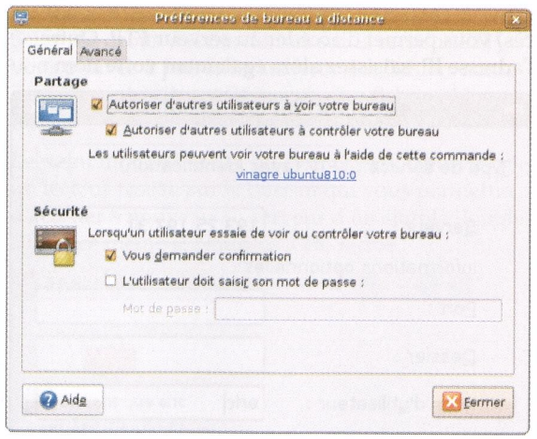
Vous pouvez travailler via le serveur terminal du PC Windows comme si vous étiez connecté en local – vous devez naturellement éviter de lancer des jeux en haute résolution… Pour quitter la session du terminal, déconnectez-vous du système en suivant la procédure habituelle. (Mais NE L'ARRÊTEZ PAS, à moins que cela soit votre souhait).

Les performances dépendant de la vitesse de votre connexion au réseau : il vaut vraiment mieux être connecté en Ethernet et sur un réseau local.

Cela fonctionne avec d'autres PC/Linux ?

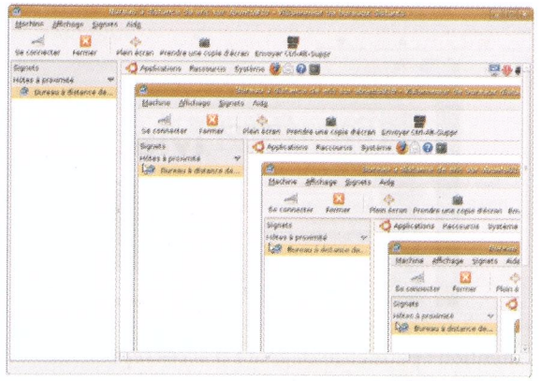
Oui, les PC sous Linux peuvent aussi accéder les uns aux autres par le biais d'un bureau à distance. En utilisant VNC.

Côté serveur, la configuration s'effectue dans **Système/Préférences/Bureau à distance** :



Pour activer la fonctionnalité serveur, vous devez sélectionner l'option **Autoriser d'autres utilisateurs à voir votre bureau**.

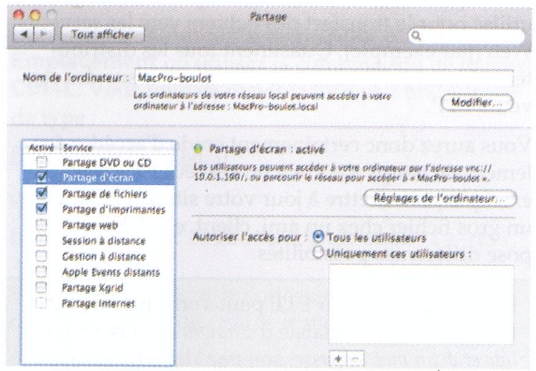
Vous pouvez alors invoquer le **Visionneur de bureaux distants** à partir du client via le menu **Applications**, sous-menu **Internet**.



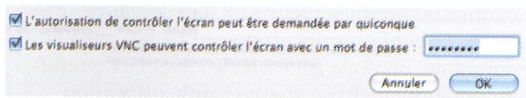
Cliquez sur **Se connecter** ou sélectionnez un ordinateur à partir de la liste afin de vous connecter à ce système par le biais de VNC. Un résultat amusant est obtenu lorsque l'on établit une connexion vers soi-même, comme présenté dans l'illustration. Avez-vous déjà essayé de placer deux miroirs face à face et de vous y regarder ? Le principe est ici le même...

Ca marche avec Mac ?

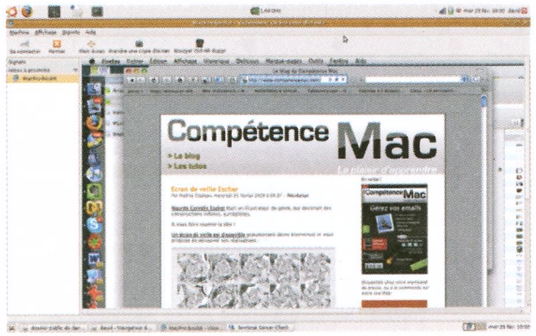
Oui. Là aussi, vous devez d'abord autoriser l'accès à distance sur le Mac, via les **Préférences Système/Partage** :



J'ai ensuite cliqué sur **Réglages de l'ordinateur** pour donner aux utilisateurs le droit de se connecter (via un mot de passe) :



Ici, je me connecte à mon Mac depuis mon portable sous Linux, en utilisant le visionneur de bureaux distants



Attention : permettre ce genre d'accès représente toujours un risque. Particulièrement si votre réseau est accessible depuis Internet et n'est pas protégé par un bon pare-feu. Ne laissez pas ce genre d'accès actif sans raison valable.

Connexion FTP

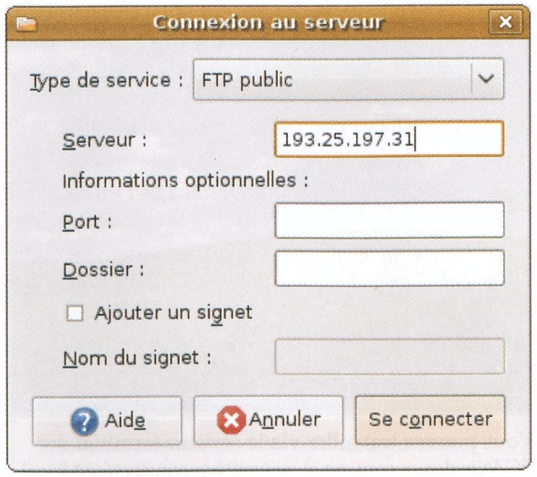
Le protocole FTP permet le transfert de fichiers à partir de ou vers un serveur FTP. Le protocole est déjà très ancien et un véritable cauchemar niveau sécurité (les données des utilisateurs sont transférées sous forme non cryptée !) ; ce qui ne l'empêche pas, aujourd'hui encore, d'être le protocole le plus utilisé pour le transfert des fichiers vers les serveurs Web (par exemple). Quasiment tous les distributeurs Linux proposent leur distribution sur des serveurs FTP.

Vous aurez donc certainement envie d'accéder également de temps à autre à un serveur FTP, par exemple pour mettre à jour votre site, ou récupérer un gros fichier chez un ami, client, etc. Ubuntu propose différentes possibilités.

L'accès à un **serveur FTP** peut s'effectuer de deux façons : soit par la saisie d'un *nom d'utilisateur valide et d'un mot de passe*, soit par l'intermédiaire d'un *accès anonyme*, également nommé **FTP anonyme**.

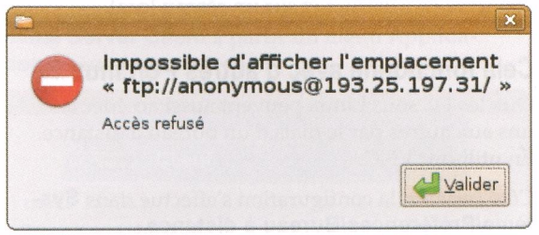
Les possibilités disponibles dépendent de la configuration du serveur FTP. Les serveurs de téléchargement gratuits proposent l'accès par FTP anonyme. En règle générale, le client FTP prend automatiquement en charge l'authentification, indiquant `anonymous` pour l'utilisateur et utilisant souvent une adresse email fictive.

La méthode plus simple consiste à se connecter directement via le menu **Raccourcis/Se connecter à un serveur** :

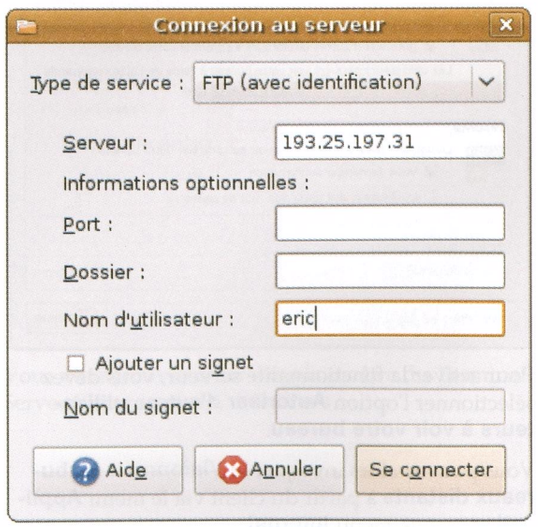


Si vous voulez accéder à un serveur FTP anonyme, choisissez **FTP public**. Il vous suffit de saisir ici l'adresse (nom DNS ou adresse IP) du serveur et de cliquer sur **Se connecter**.

Soit vous verrez apparaître le navigateur de fichiers de Nautilus, soit vous obtiendrez un message d'erreur, si la connexion anonyme n'est pas autorisée :

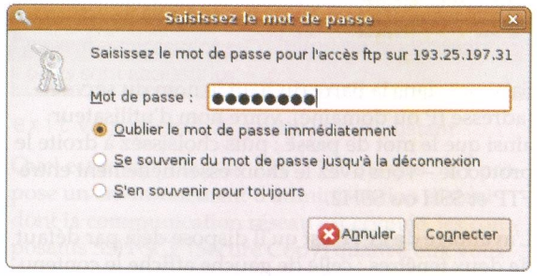


Dans ce cas, seule l'option FTP (avec restriction d'accès) vous permet d'accéder au serveur FTP. Outre l'adresse IP, saisissez alors également votre nom :



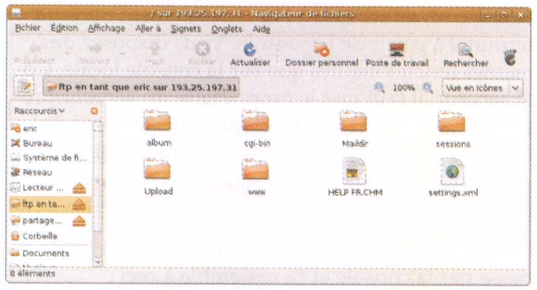
Si votre serveur le permet, l'accès SSH est sécurisé : les données sont cryptées. Il est donc à privilégier. Mais tous les serveurs ne le permettent pas.

Si vous avez pris contact avec le serveur FTP, vous êtes invité à indiquer le mot de passe pour l'utilisateur indiqué :



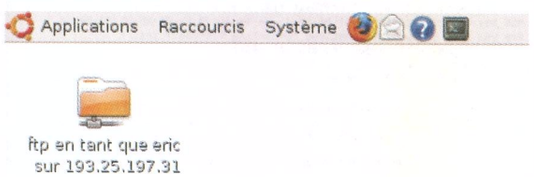
Vous pouvez aussi préciser ici que le client fournisse automatiquement le mot de passe, sur demande. La sécurité laissant de toute façon à désirer sous FTP, la sélection de cette option ne représente donc pas un très gros risque.

Si le mot de passe est correct vous avez accès au serveur FTP directement depuis Nautilus :



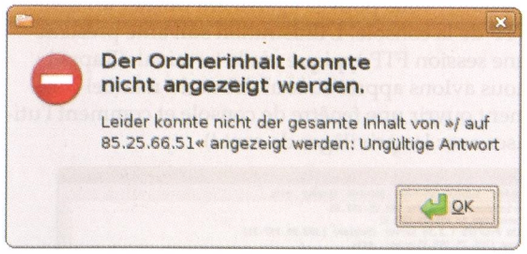
Selon les droits d'accès définis sur le serveur FTP, vous pouvez à présent transférer des fichiers sur le serveur (upload) ou télécharger des fichiers (download).

Le point intéressant est, qu'à ce niveau, Ubuntu crée un lecteur réseau sur le bureau qui vous permettra d'accéder à nouveau au serveur d'un simple double-clic, et ce même si Nautilus a été fermé :



Pour supprimer ce raccourci, cliquez sur l'icône à l'aide du bouton droit de la souris et sélectionnez le point de menu **Démonter le volume**.

Le comportement n'est malheureusement pas très stable – il m'est souvent arrivé lors de tentatives d'accès d'obtenir un message d'erreur qui ne disparaissait plus.



Voici une astuce que j'ai déjà eu l'occasion de vous présenter dans le cadre de la gestion des processus : ouvrez un terminal, recherchez Nautilus à partir de la liste des processus (`ps ax`) et notez l'ID de son processus (PID) figurant dans la première colonne. Saisissez ensuite la commande `kill <PID de Nautilus>`. Ne vous inquiétez pas, le processus Nautilus redémarrera automatiquement.

Vous pouvez d'ailleurs aussi accéder à un serveur FTP directement via le navigateur de fichiers de Nautilus. Pour ce faire, ouvrez Nautilus. (Il vous suffit pour cela de sélectionner n'importe quelle commande du menu **Raccourcis**.)

Dans Nautilus, ouvrez le menu **Aller à**, sous-menu **Emplacement** ou utilisez la combinaison de touches **Ctrl+L**. Vous pouvez maintenant saisir une adresse de type :

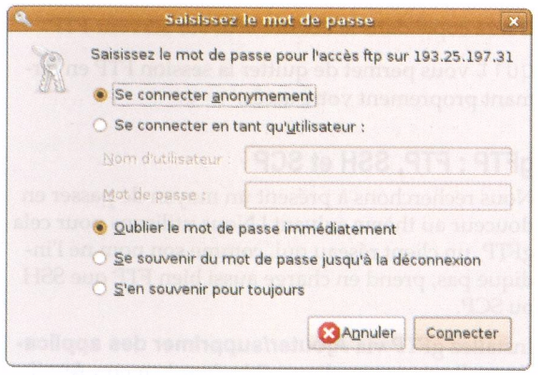
ftp://eric@85.25.66.51

qui se lit :

ftp://nom-d-utilisateur@adresse-IP-du-serveur

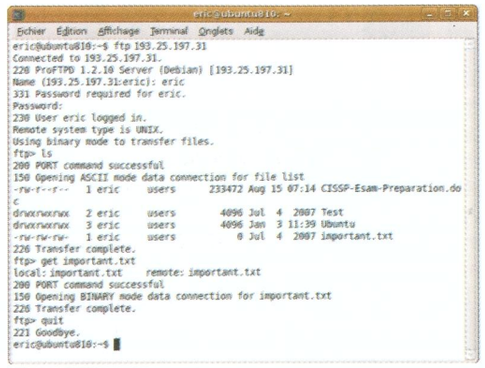
Si vous avez indiqué un utilisateur, vous serez une nouvelle fois invité à saisir le mot de passe.

Si vous tapez uniquement ftp://<Adresse du serveur FTP>, vous pouvez choisir la façon dont vous voulez vous connecter :



Via la console

En outre, vous pouvez également utiliser le client FTP de la console. L'illustration suivante présente une session FTP typique via le terminal. (Rappel : nous avions appris au tout début du manuel comment ouvrir une fenêtre de console et comment l'utiliser avec les privilèges de root !)



Tapez `ftp` suivi d'un espace et ajoutez le domaine ou l'adresse IP su serveur. Appuyez sur **Entrée.**

Saisissez alors le nom d'utilisateur et le mot de passe qui vous sont demandés. Rien ne s'affiche quand vous le tapez, c'est normal, c'est une sécurité. : cela sert à ce que personne ne puisse épier le mot de passe.

Une fois connecté au serveur, vous pouvez utiliser les commandes du Shell.

La commande `ls` vous indique le contenu du dossier FTP.

Avec `get`, vous téléchargez des fichiers dans le dossier local actuel.

Avec `put`, vous transférez des fichiers sur le serveur. Pour ce faire, il vous suffit d'indiquer le nom du fichier à uploader, accompagné éventuellement du chemin du dossier, soit par exemple :

```
put /home/eric/important.txt.
```

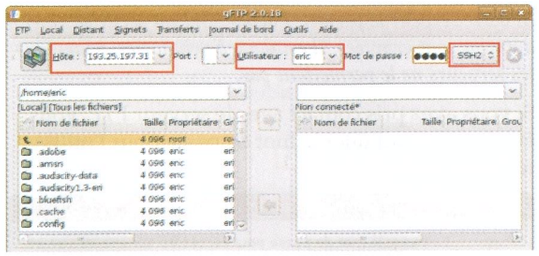
Il sera copié dans le dossier actif sur serveur FTP.

`quit` vous permet de quitter la session FTP en fermant proprement votre accès.

gFTP : FTP, SSH et SCP

Nous recherchons à présent un moyen de passer en douceur au thème suivant ! Nous utilisons pour cela gFTP, un client réseau qui, comme son nom ne l'indique pas, prend en charge aussi bien FTP que SSH ou SCP.

Installez gFTP via **Ajouter/supprimer des applications**. Il est ensuite disponible dans le menu **Applications/Internet**.



Saisissez dans la barre du haut le nom du serveur (adresse IP ou domaine), votre nom d'utilisateur ainsi que le mot de passe ; puis choisissez à droite le protocole – vous avez le choix essentiellement entre FTP et SSH ou SSH2.

L'avantage de gFTP est qu'il dispose déjà par défaut de deux fenêtres : celle de gauche affiche le contenu local (vos dossiers, sur votre ordinateur), celle de droite le contenu distant (le contenu du serveur auquel vous êtes connecté).

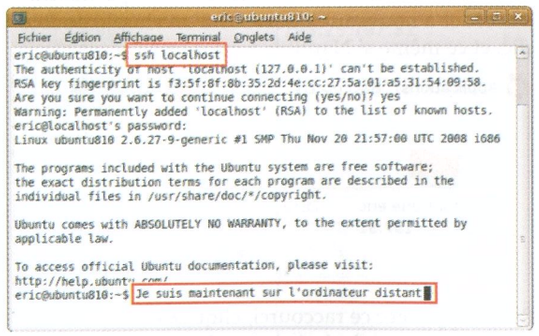
Avant de vous montrer ce que gFTP et SSH vous permettent de faire, permettez-moi un bref aparté :

SSH (Secure Shell) permet une connexion sécurisée à un serveur. Au final, c'est comme si vous étiez en train de travailler tout à fait normalement dans un terminal à partir de la console. Vous disposez des mêmes fonctionnalités que sur la console locale. Côté serveur, il faut que le propriétaire du serveur propose un accès SSH (c'est indiqué dans le détail de l'offre) ou si vous en êtes l'administrateur, il disposer de `openssh-server` que vous pouvez installer séparément via *Synaptic*.

Normalement se connecter à un serveur via SSH se passe par la console et ça ressemble à ceci :

```
ssh <adresse du serveur>
```

Si je souhaite me connecter avec mon propre ordinateur (c'est également possible, mais utile uniquement dans le cadre de tests), j'indique la cible `localhost` :



Une fois connecté, vous disposez d'un shell sur le serveur, vous pouvez travailler comme si vous étiez sur votre propose ordinateur. C'est très pratique (si vous comparez avec un accès bureau à distance que l'on a vu plus haut, vous verrez que c'est bien plus

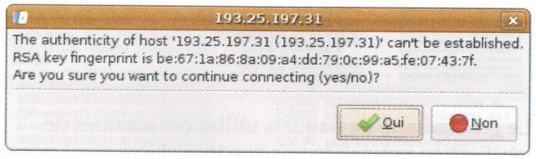
rapide, pas besoin du haut débit pour espérer pouvoir travailler correctement : une connexion téléphonique de base suffira, même si sa lenteur risque de se faire sentir lors des gros transferts) et c'est très puissant.

> C'est d'ailleurs ainsi que sont administrés la plupart des serveurs Linux sur Internet, car la bande passante nécessaire est faible et toutes les fonctions sont accessibles.

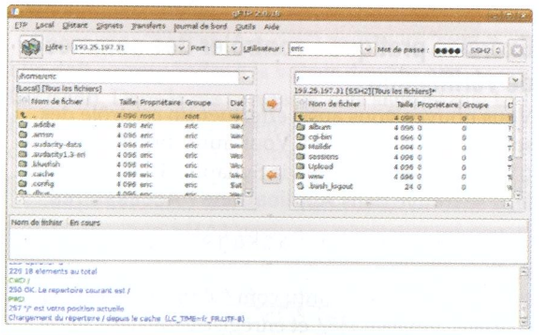
`exit` vous permet de quitter le shell distant.

Quel est le rapport avec FTP ? Tandis que SSH propose un environnement d'administration distant dont la communication réseau est cryptée, les composants SSH SCP (Secure Copy) quant à eux vous permettent un transfert sûr et crypté des fichiers, comme vous le feriez avec FTP. C'est là qu'intervient à nouveau gFTP, car c'est un client SCP fort pratique.

Lors de la première connexion, vous devez confirmer l'authenticité du serveur (comme via la console) :



Vous avez ensuite accès aux fichiers du serveur :



> Si vous avez déjà travaillé avec des clients SSH sous Windows, vous remarquerez les similitudes visuelles entre **gFTP** et **WSFTP LE** ou **WinSCP**.
>
> Sous Windows, vous avez besoin d'un client SSH spécifique, comme par ex. **PuTTY** ; alors que sous Linux, vous pouvez travailler directement via la console et la commande ssh.

Vous voilà maintenant fin prêt pour la gestion réseau, que ce soit sur le réseau local ou sur Internet ! Bon amusement et bonne découverte !

Sécurité

Linux réduit considérablement les problèmes auxquels les utilisateurs de Windows sont confrontés. Les risques de virus, vers et autres chevaux de Troie sont par ex. bien plus faibles : vous n'avez pas besoin d'un logiciel antivirus sur Linux.

Il existe néanmoins toute une série de mesures de sécurité à prendre pour protéger vos données.

Dans ce chapitre, il sera question de cryptage, de signature numérique et d'autres mesures de protection conçues pour donner du fil à retordre aux pirates. Je vous réserve ici quelques trucs et astuces que vous devez absolument connaître – alors plus une seconde à perdre, c'est parti !

Intégrité des téléchargements

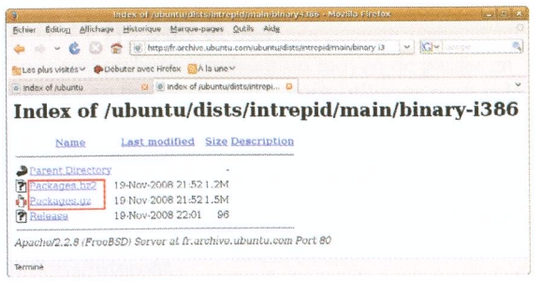
Une mesure importante consiste à vérifier l'intégrité des logiciels téléchargés à partir d'Internet. Les gestionnaires de paquets effectuent ce contrôle automatiquement. Mais s'il s'agit de téléchargements provenant d'autres sources, c'est à vous qu'incombe la vérification de l'intégrité.

Cela permet de vérifier, entre autres choses, qu'un logiciel n'a pas été modifié par un pirate qui y aurait, par exemple introduit un code malveillant.

> Le dépôt principal de la distribution Debian a été récemment piraté. Il s'agit pour ainsi dire du pire scénario imaginable ! Tous les serveurs miroirs prélèvent les paquets à partir de ce dépôt. Un pirate qui modifie des paquets à cet endroit peut infecter *tous* les ordinateurs téléchargeant et installant les paquets modifiés !

Comment diable se protéger de ce fléau ?

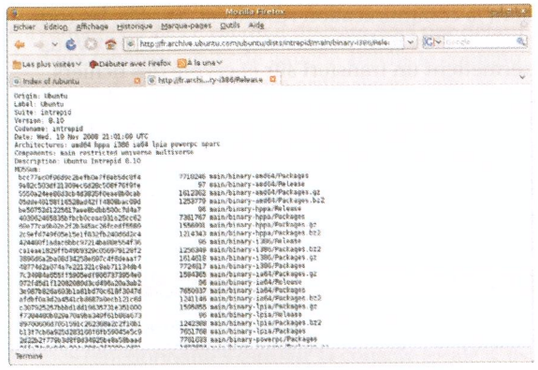
Comme vous le savez déjà, la gestion des paquets Ubuntu est basée sur APT. Cette dernière prélève dans les différents dépôts de composants (par ex. *intrepid main*) un fichier nommé `Packages` destiné à la liste des paquets disponibles :



Une somme de vérification est utilisée pour confirmer l'authenticité de ce fichier.

> Les sommes de vérification sont générées par des algorithmes de hachage, tels que MD5, SHA-1 ou SHA-256. Il s'agit d'un cryptage à sens unique qui produit une chaîne de caractères (la somme de vérification) de longueur définie. Ce cryptage n'est pas réversible.

Le dossier principal d'un dépôt de distribution contient un fichier `Release`. Il renferme les sommes de vérification de tous les fichiers `Packages` existants.



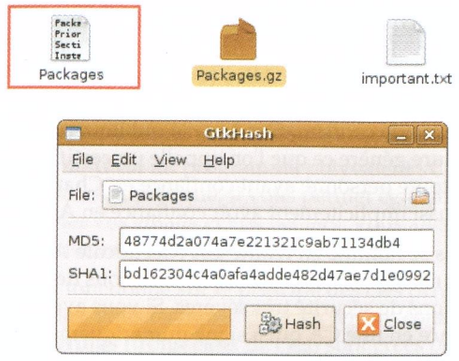
Le gestionnaire de paquets utilise ces sommes de vérification pour s'assurer que les fichiers n'ont pas été modifiés subrepticement.

Exercice pratique : contrôler des sommes de vérification

Vous pouvez vous aussi effectuer un contrôle des sommes de vérification. Vous aurez besoin de `gtkhash`, disponible via Synaptic. Une fois installé, il se trouve dans **Applications / Accessoires**.

Téléchargez un fichier `Packages.gz`, par ex. à l'adresse http://fr.archive.ubuntu.com/ubuntu/dists/intrepid/main/binary-i386/ et enregistrez-le dans votre dossier personnel.

Faites un clic droit sur le fichier téléchargé et sélectionnez la commande **Extraire ici** du menu contextuel. Vous disposez maintenant d'une version décompressée du fichier `Packages` que vous pouvez vérifier via **GtkHash**. Pour ce faire, sélectionnez le fichier et cliquez sur **Hash**.



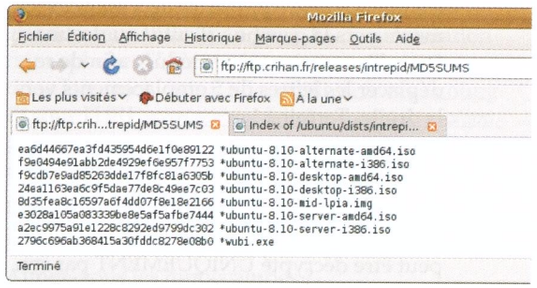
Vous pouvez à présent comparer : la valeur de hachage actuelle doit correspondre à celle du fichier `Release`. Regardez dans le dossier principal de la distribution (par ex. http://fr.archive.ubuntu.com/ubuntu/dists/intrepid/) et ouvrez le fichier `Release`. Comparez la somme de vérification MD5 :

```
Origin: Ubuntu
Label: Ubuntu
Suite: intrepid
Version: 8.10
Codename: intrepid
Date: Wed, 19 Nov 2008 21:01:09 UTC
Architectures: amd64 hppa i386 ia64 lpia powerpc sparc
Components: main restricted universe multiverse
Description: Ubuntu Intrepid 8.10
MD5Sum:
 bcc77ac0f96d9c2befb0e7f6eb5dc8f4          7718246 main/binary-amd64/Packages
 9a82c503df21309ec6d28c508f76f9fe               97 main/binary-amd64/Release
 5550a24ee86d3cb4d3835f0eae8b0cab          1612362 main/binary-amd64/Packages.gz
 05dde40158f16528ad42ff4808bac09d          1253779 main/binary-amd64/Packages.bz2
 be50752d1225617aee8bdbb500c7d4a7               96 main/binary-hppa/Release
 403962465835bfbcb0ceae931e25cc62          7361767 main/binary-hppa/Packages
 69e77ca9b02e2f2b345ac26fcedf5589          1556991 main/binary-hppa/Packages.gz
 2c9efd749f05e15e1f832fb240d6d2c4          1214343 main/binary-hppa/Packages.bz2
 424480f1adac6bbc97214ba88e554f36               96 main/binary-i386/Release
 caeae1829ffb49b9329c056979128f2          1256349 main/binary-i386/Packages.bz2
 3896d6a2ba08d34258e697c4f8deaaf7          1614618 main/binary-i386/Packages.gz
 48774d2a074a7e221321c9ab71134db4          7724617 main/binary-i386/Packages
 7c34984a655ff5905edf9067373954e0          1594365 main/binary-ia64/Packages.gz
```

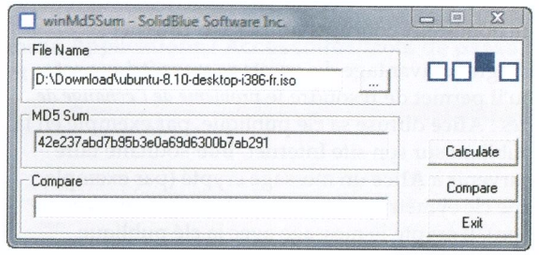
Elle doit correspondre à la valeur donnée par **GtkHash**. Dans le cas contraire, cela signifie que le fichier a été modifié subrepticement et que vous ne devez plus lui accorder votre confiance !

> Vous pouvez bien sûr aussi comparer les sommes de vérification **SHA1** que vous trouverez un peu plus loin dans le fichier `Release`. Comme vous pouvez le constater, la somme de vérification est ici plus longue qu'avec **MD5**, **MD5** utilisant un cryptage 128 bits et **SHA1** un cryptage 160 bits. De ce fait, **SHA1** est en principe plus sûr – il a malheureusement été prouvé récemment qu'ici non plus l'algorithme n'est pas totalement sûr. D'où l'introduction de **SHA2**, également nommé **SHA256** (en référence à la longueur de clé 256 bits). Il n'est pas encore pris en charge par *GtkHash*, mais figure déjà dans le fichier `Release`.

Vous pouvez utiliser ce type de vérification partout où des valeurs de hachage analogues sont mises à disposition pour les téléchargements. Prenons l'exemple des images ISO d'Ubuntu : vous trouverez dans le dossier de téléchargement un fichier `MD5SUMS` contenant les sommes de vérification MD5 de tous les fichiers ISO :



Si vous avez téléchargé une image ISO sous Windows, vous pouvez vérifier le fichier Image à l'aide du programme gratuit **winMd5Sum** :



Quiz : parvenez-vous à retrouver la somme de vérification calculée en bas dans l'illustration ? Astuce : recherchez les deux premiers et les deux derniers chiffres.

Quis custodiet ipsos custodes ?

Qui surveille le gardien ? Cette locution latine contient une question fondamentale : qui me garantit que le fichier de vérification `Release` n'a pas été tout simplement modifié ?

C'est ici qu'intervient un autre mécanisme, pas toujours évident à comprendre, appelé **signature numérique**. Je vais essayer de vous le présenter en quelques mots :

Il existe trois types de cryptage :

1. **Hachage** : un cryptage à sens unique qui vous permet de générer un nombre défini de caractères en tant que somme de vérification pour un texte en clair de longueur quelconque. Cette procédure n'est pas réversible – il est (quasiment) impossible de découvrir le texte d'origine à partir d'une valeur de hachage.
2. **Cryptage symétrique** : les deux parties obtiennent la même clé, utilisée aussi bien pour le cryptage que pour le décryptage. En outre, ces deux parties connaissent l'algorithme défini et, en fonction de la clé, savent ce qu'il y a à faire. Un exemple simple, l'*algorithme* est le suivant : déplace des lettres de X emplacements vers la gauche. La

clé est X. Si sa valeur est par ex. 5, les deux parties savent alors que le texte d'origine a été déplacé lettre par lettre de 5 emplacements vers la gauche – il suffit au destinataire d'inverser cette procédure afin de redéplacer les lettres de 5 emplacements vers la droite. Terminé !

3. **Cryptage asymétrique** : c'est ici une paire de clés qui est générée, une clé privée (Private Key) et une clé publique (Public Key). Le texte en clair, crypté par une des clés, peut être décrypté UNIQUEMENT par son pendant.

Observons d'un peu plus près encore le cryptage asymétrique : il y a ici de nombreuses informations de base à connaître et qui vous aident à comprendre de nombreux concepts utilisés sur Internet !

Un grand avantage du cryptage asymétrique est qu'il permet de résoudre le *problème de l'échange de clés* : **Alice** diffuse sa clé publique, par exemple en la publiant sur son site Internet. **Bob** souhaite faire parvenir à **Alice** un message crypté (par exemple une clé symétrique pour la suite de la communication). Il crypte le message avec la clé publique d'**Alice**. **Alice** est la seule à pouvoir décrypter ce message à l'aide de sa clé privée (conservée en un lieu secret et sûr).

Bob et Alice peuvent ensuite communiquer facilement entre eux via la clé symétrique, cette dernière nécessitant considérablement moins de calculs qu'un cryptage asymétrique (la longueur des clés symétriques est comprise entre 128 et 512 bits, tandis que celle des clés asymétriques s'étend de 1 024 à 4 096 bits !).

Génial ! Mais en quoi cela va-t-il nous être utile ? C'est très simple, si nous considérons la chose sous un autre angle : si **Alice** crypte un message avec sa clé privée, tous ceux en possession de la clé publique d'**Alice** (donc théoriquement tout le monde) peuvent décrypter ce message. Qu'est-ce qui nous indique cela ?

> La clé privée d'**Alice** étant (ou devant normalement être) secrète et connue uniquement d'**Alice**, il est donc prouvé qu'un message crypté avec la clé privée d'**Alice** doit provenir d'**Alice** ! Ce que tout le monde peut vérifier ! Voilà, la **signature numérique** est terminée !

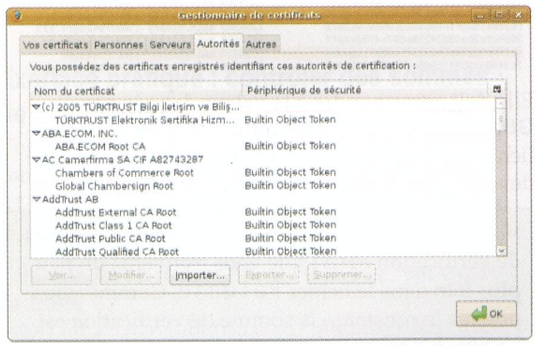
Ah bon ? Et si je ne connais pas **Alice** et que de ce fait je n'ai pas confiance en sa clé publique ? Dans ce cas, nous faisons appel à la PKI – Public Key Infrastructure. C'est là qu'intervient **Charles** : **Alice** connaît **Charles** et a confiance en lui. **Bob** connaît aussi **Charles** et a également confiance en lui – mais il ne connaît pas **Alice**.

Mais voilà qu'**Alice** s'adresse à **Charles** et lui demande une lettre de confirmation, ce que **Charles** délivre naturellement à **Alice** (vu qu'il s'agit d'une personne digne de confiance). Elle joint à présent cette lettre de confirmation à sa clé publique. Cette procédure génère ce que l'on appelle un **certificat**. Ainsi, toute personne ayant confiance en **Charles** peut aussi implicitement avoir confiance en **Alice**.

Charles se nomme en réalité **CA** (Certificate Authority) et crée le certificat correspondant sous des conditions de sécurité très sévères. Si vous avez confiance en une CA qui a signé numériquement la clé publique d'un étranger, vous pouvez alors aussi avoir confiance en cet étranger.

Ouf, quel casse-tête ! Le prix à payer pour assurer une bonne maîtrise des bases du **cryptage et des signatures numériques**. Mais revenons-en à la pratique !

Le principe cité plus haut est utilisé dans le cadre des téléchargements signés (par ex. celui des plug-ins pour les navigateurs) ou des connexions HTTPS (par ex. pour les services bancaires en ligne). Votre navigateur contient une série de certificats CA dignes de confiance. Dans *Mozilla Firefox*, allez dans le menu **Edition**, sous-menu **Préférences**, onglet **Avancé > Chiffrement**. Cliquez ici sur le bouton **Afficher les certificats** pour faire apparaître les certificats CA en lesquels *Firefox* a confiance.



Nous ne sommes malheureusement pas encore arrivés au bout de nos peines. Vous avez déjà appris beaucoup de choses concernant les signatures, mais il existe une autre structure en plus de la PKI classique. Elle se base sur ce que l'on appelle un **Web of Trust**. Ici, les utilisateurs se font mutuellement confiance et forment une Chain of Trust, une chaîne de confiance. Le programme concerné se nomme *GnuPG* (Gnu Privacy Guard).

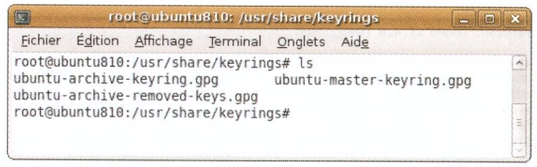
GnuPG

Pourquoi diable devons-nous maintenant encore nous occuper de ce *GnuPG* ? C'est très simple : le dossier de distribution dans lequel vous avez trouvé le fichier `Release` contient certainement aussi un fichier `Release.gpg`. Ce fichier renferme la signature *GnuPG* du fichier `Release`, créée à partir de la clé privée du dépôt correspondant. Elle prouve en dernière instance que le fichier `Release` est valide.

Mais *GnuPG* est un programme polyvalent ! Outre la signature numérique, il vous permet également de crypter vos emails et vos fichiers. Vous allez l'apprécier, croyez-moi ;-).

GnuPG fonctionne sur le principe des clés privées et publiques, vous naviguez donc en terrain connu. Des **keyrings** (trousseaux de clés) sont créés pour la gestion de ces clés.

Il existe un paquet nommé `ubuntu-keyring` qui contient les **clés publiques** de tous les dépôts Ubuntu. Il est installé automatiquement à chaque installation Ubuntu. Vous trouverez ces keyrings sous `/usr/share/keyrings`. Il s'agit de trois fichiers ou trousseaux de clés.



C'est là qu'intervient la *Chain of Trust* : si un paquet issu d'un dépôt est maintenant installé, le gestionnaire de paquets vérifie à l'aide de la clé publique adaptée du dépôt respectif si la signature du fichier `Release` (dans le fichier `Release.gpg`) est correcte. Si tel est le cas, il peut comparer la somme de vérification MD5 ou SHA du fichier `Packages` respectif à celle contenue dans le fichier `Release`, et ainsi vérifier son authenticité.

Le fichier `Packages` englobe maintenant tous les paquets d'un composant de dépôt (comme par ex. *main* ou *security*) – accompagnés une nouvelle fois de sommes de vérification, comme ici avec le paquet `seahorse` :

```
Package: seahorse
Priority: optional
Section: gnome
Installed-Size: 10104
Maintainer: Ubuntu Core Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Jose Carlos Garcia Sogo <jsogo@debian.org>
Architecture: i386
Version: 2.24.1-0ubuntu1
Depends: libatk1.0-0 (>= 1.20.0), libavahi-client3 (>= 0.6.16), libavahi-common3 (>= 0.6.16), libava
= 0.71), libgconf2-4 (>= 2.13.5), libglade2-0 (>= 1:2.6.1), libglib2.0-0 (>= 2.18.0), libgnome-keyring0
libldap-2.4-2 (>= 2.4.7), libnotify1 (>= 0.4.4), libnotify1-gtk2.10, libpango1.0-0 (>= 1.21.6), libsoup2.
Recommends: openssh-client, seahorse-plugins
Filename: pool/main/s/seahorse/seahorse_2.24.1-0ubuntu1_i386.deb
Size: 1197210
MD5sum: 6500c500b32e4aa26cad6703f49db6dd
SHA1: 3cd22fd6086bfef0da9eab09049c24b8fe1c417f
SHA256: 3d1a3dc508cfdb47a0073f63d01a2f84f07a81e142ae174c1e510afa5567c906
Description: A Gnome front end for GnuPG
 Seahorse is a Gnome front end for GnuPG - the Gnu Privacy Guard program. It
 is a tool for secure communications and data storage. Data encryption and
 digital signature creation can easily be performed through a GUI and Key
 Management operations can easily be carried out through an intuitive
 interface.
```

Ces sommes de vérification peuvent être ensuite comparées au paquet téléchargé afin de vérifier une dernière fois son authenticité – et c'est terminé !

GPG ou PGP ?

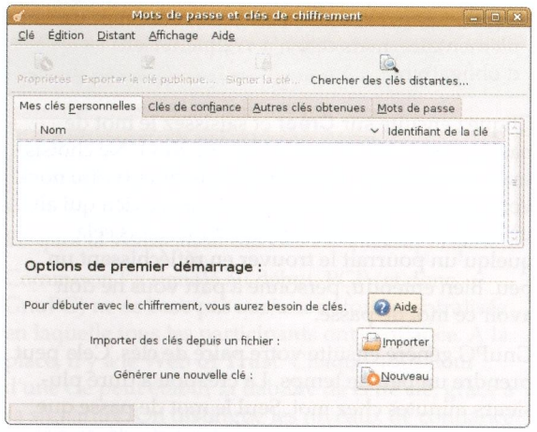
GnuPG est une implémentation libre de PGP (Pretty Good Privacy). En principe, GnuPG est utilisé en ligne de commande (man gpg). La documentation francophone d'Ubuntu fournit toutes les informations à ce sujet : http://doc.ubuntu-fr.org/gnupg. Et sous l'URL www.spywarewarrior.com/uiuc/gpg/gpg-com-0.htm, vous trouverez également une « Command Reference », cette fois-ci en anglais.

Je préfère toutefois vous présenter une très bonne interface graphique pour GnuPG nommée *Seahorse*.

Aimez-vous les hippocampes ?

Seahorse

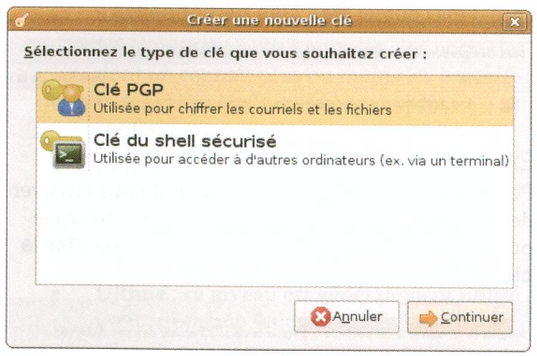
Par défaut, il est installé avec Ubuntu depuis Ubuntu 8.04 (Hardy Heron). Vous la trouverez dans le menu **Applications / Accessoires/Mots de passe et clés de chiffrement**.



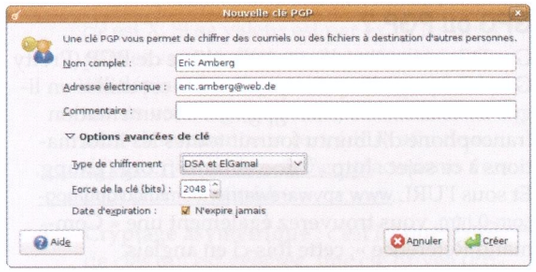
Pour utiliser GnuPG, vous devez disposer de vos deux clés (publiques et privées) et de la clé publiques des personnes à qui vous souhaitez écrire.

Créer vos clés

Cliquez sur le bouton **Nouveau** :



Sélectionnez **Clé PGP**, nous n'avons pas besoin de clé SSH pour le moment.



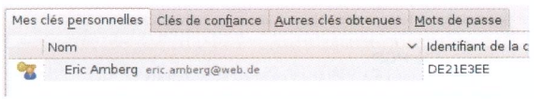
Indiquez votre nom et l'adresse email pour laquelle vous souhaitez utiliser la clé (une paire de clé est associée à une adresse email).

> Les **Options avancées de clé** vous permettent de définir les algorithmes de cryptage ainsi que la longueur des clés. Vous pouvez aussi fixer ici la date d'expiration. Les valeurs par défaut conviennent généralement.
>
> Fixer une date d'expiration permet de renouveler régulièrement votre clé. C'est une sécurité supplémentaire, mais cela n'a vraiment rien d'obligatoire.

Cliquez ensuite sur **Créer** et saisissez le mot de passe pour votre *clé privée* (Private Key). Ne choisissez pas un mot de passe facile à deviner (votre nom, votre date de naissance, etc.), n'utilisez rien qui ait un sens. Rien qui puisse être deviné. Sans cela, quelqu'un pourrait le trouver en réfléchissant un peu. Bien entendu, personne à part vous ne doit avoir ce mot de passe.

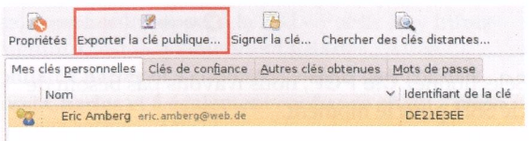
GnuPG génère ensuite votre paire de clés. Cela peut prendre un peu de temps. La création a duré plusieurs minutes chez moi. Seul le mot de passe que vous avez choisi pourra décrypter cette clé, si vous le perdez, vous ne pourrez plus décrypter vos emails et vos fichiers et personne (à part peut-être la NSA, avec beaucoup de temps) ne pourra le faire.

Les clés sont stockées, avec d'autres fichiers, dans le dossier `~/.gnupg`. Elle est bien entendu aussi affichée dans l'onglet **Mes clés personnelles** :

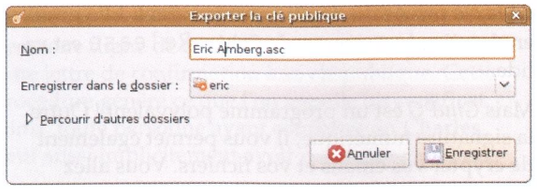


Exporter une clé

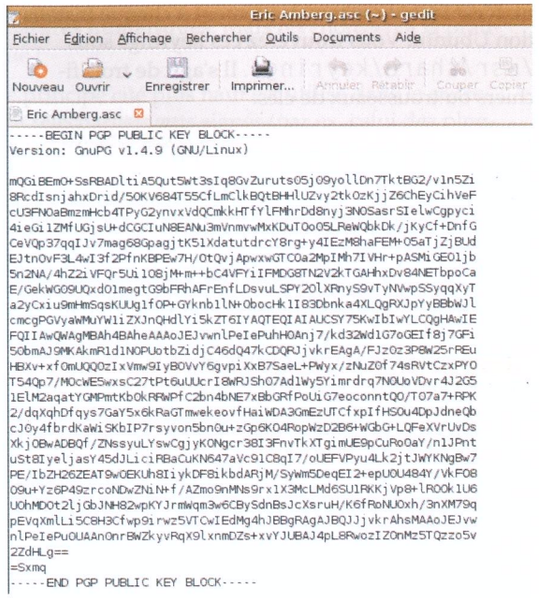
Pour que nos correspondants puissent nous envoyer des emails cryptés, nous devons exporter notre clé publique. Sélectionnez-la et cliquez sur **Exporter la clé publique** dans la barre d'outils :



Vous pouvez changer le nom du fichier si vous en avez envie :



Enregistrez le fichier `.asc` dans le dossier souhaité. C'est un simple fichier… ou presque : ne double-cliquez pas dessus pour l'ouvrir, car cela importerait la clé dans votre trousseau (c'est pratique quand on reçoit des clés publiques, mais pas avec la sienne). A la place, faites un clic droit et choisissez **Ouvrir avec « Éditeur de texte »** :



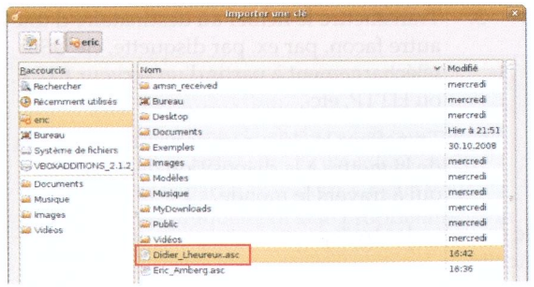
Woaw. Quelle clé. Avec elle vos correspondants vont pouvoir vous envoyer des messages que vous seul serez capable d'ouvrir.

Vous n'avez plus qu'à envoyer votre clé publique, ce fichier, à vos correspondants. Pour pouvoir l'utiliser, il doit l'importer dans son trousseau, comme vous devrez importer leurs clés publiques de votre côté si vous voulez leur envoyer des emails cryptés.

Sinon, vous devez récupérer vous-même les fichiers de clé publique correspondants de votre interlocuteur et importer la clé.

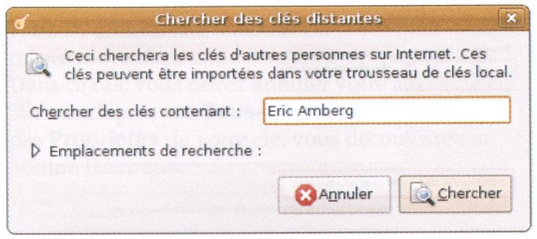
Importer une clé

Le menu **Clé** vous permet d'**Importer** une clé :

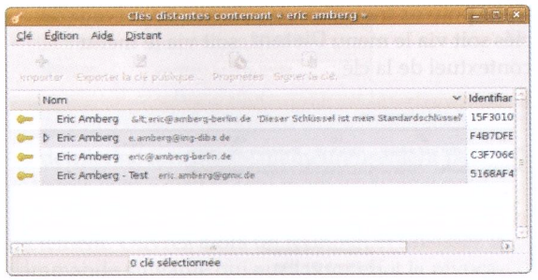


La clé publique est localisée à présent dans l'onglet **Autres clés obtenues**.

Si votre interlocuteur a déposé sa clé publique sur un serveur de clés, vous pouvez aussi récupérer la clé via le menu **Distant**. Cliquez sur **Chercher des clés distantes** et indiquez un critère de recherche, par exemple le nom du propriétaire de la clé :



Un clic sur **Chercher** et tous les éléments trouvés sont affichés dans la liste de résultats :



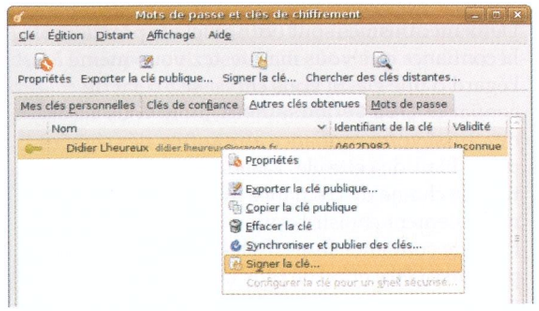
Il reste à trouver la bonne personne s'il y en a plusieurs. En cas de doute, il vaut mieux demander au correspondant sa clé…

Cliquez sur la clé puis sur **Importer**.

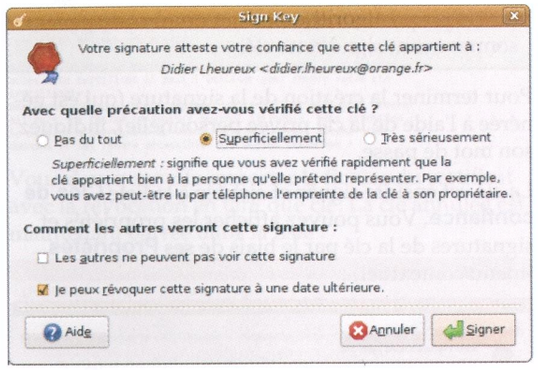
> Je vous expliquerai un peu plus loin comment transférer vos propres clés sur un serveur de clés.

Signer une clé

Pour le moment, vous n'avez pas encore accordé votre confiance à cette clé. Pour qu'une clé devienne digne de confiance, vous devez la signer. Pour ce faire, faites un clic droit sur la clé et choisissez la commande **Signer la clé** :



C'est là que commence quelque chose de typique à PGP, mais qui peut paraître un peu curieux de prime abord :



En quoi la vérification des clés est-elle importante ?

Comme on l'a signalé au début, PGP (et donc GnuPG) ne se base pas sur une autorité centralisée en laquelle tous les participants ont confiance. A la place, il y a le **Web of Trust**. Chaque utilisateur d'une clé peut établir la fiabilité de cette clé, grâce à sa signature. On distingue les niveaux de confiance suivants :

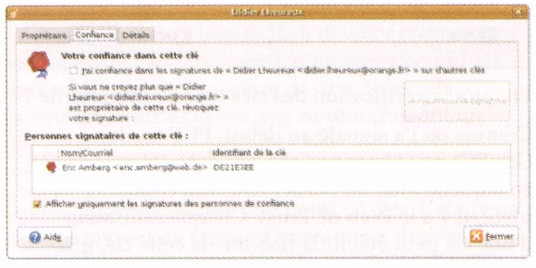
- **Inconnue** : une clé existante pour laquelle aucune précision n'a encore été apportée.
- **Jamais** : les signatures de ce contact ne sont jamais reconnues comme valides.
- **Marginale** : le niveau de confiance le plus fréquemment utilisé pour les clés normales dont on connaît plus ou moins les propriétaires et auxquels on fait à peu près confiance.
- **Complète** : ce niveau doit être accordé uniquement aux clés dont les propriétaires sont très bien connus et auxquels ont fait pleinement confiance.
- **Ultime** : ce niveau est généralement utilisé pour soi-même uniquement.

Deux mécanismes sont utilisés ici : le premier est la confiance que vous manifestez vous-même à l'égard d'une clé. Si vous choisissez l'état de confiance **Très sérieusement**, alors vous accordez aussi votre confiance aux autres clés signées par le biais de cette clé. Pour signer la clé sans activer la chaîne de confiance implicite, vous devez normalement choisir l'option du milieu, à savoir l'état de confiance **Superficiellement**. Ce réglage peut aussi être effectué plus tard et à tout moment dans l'onglet **Confiance** des propriétés de la clé.

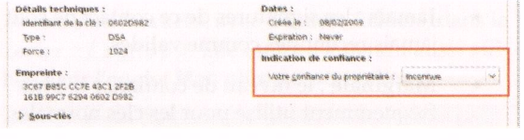
Le deuxième mécanisme est le nombre de signatures propre à une clé. La fiabilité de la clé augmente proportionnellement au nombre de personnes ayant signé cette clé.

Pour terminer la création de la signature (qui est générée à l'aide de la clé privée personnelle), indiquez son mot de passe.

La clé est ensuite signée et ajoutée à la liste **Clés de confiance**. Vous pouvez afficher les propriétés et signatures de la clé par le biais de ses **Propriétés** (menu contextuel) :



Comme vous pouvez le constater, la clé a été signée par nous-mêmes. Dans l'onglet **Détails**, vous pouvez modifier à tout moment l'état de confiance par l'intermédiaire de **Indication de confiance**.



Mettre une clé à disposition

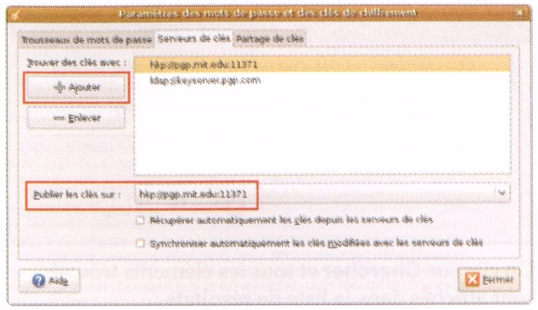
Si vous avez créé une clé, vous devez faire parvenir à votre interlocuteur la partie publique, à savoir la clé publique. Différentes possibilités s'offrent à vous :

1. Copier l'intégralité de la clé (y compris les lignes de début et de fin BEGIN PGP et END PGP) dans un email sous la forme de texte. Le destinataire peut alors enregistrer cette clé dans un fichier et y accéder quand bon lui semble.
2. Envoyer le fichier de la clé au destinataire sous la forme de pièce jointe d'un email.
3. Transmettre le fichier au destinataire d'une autre façon, par ex. par disquette, clé USB, téléchargement à partir d'un serveur FTP ou HTTP, etc.
4. Transférer la clé sur un serveur de clés afin de la mettre à la disposition de n'importe qui à travers le monde.

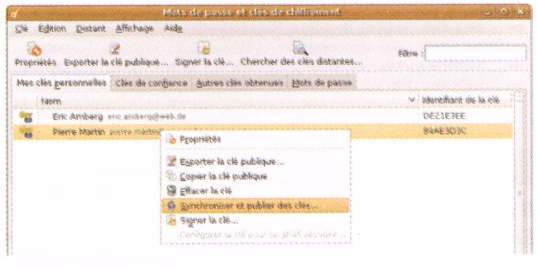
Je crois que les trois premiers ne poseront aucun difficulté.

Serveur de clé

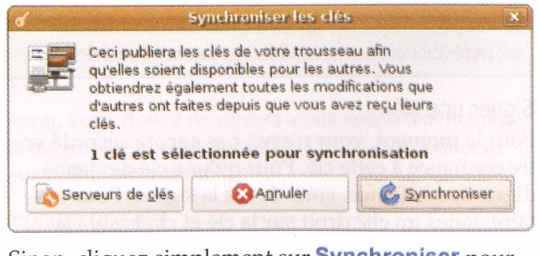
Sur Internet, il existe toute une série de serveurs de clés PGP se synchronisant en permanence. *Seahorse* contient déjà deux serveurs de clés, mais aucun serveur n'est sélectionné par défaut pour le transfert de clés, dans le menu **Édition | Préférences**. Affichez l'onglet **Serveurs de clés** :



Vous pouvez transférer une clé sur un serveur de clés soit via le menu **Distant**, soit via le menu contextuel de la clé :



Si nécessaire, vous pouvez aussi accéder à l'onglet **Serveurs de clés** à partir d'ici.



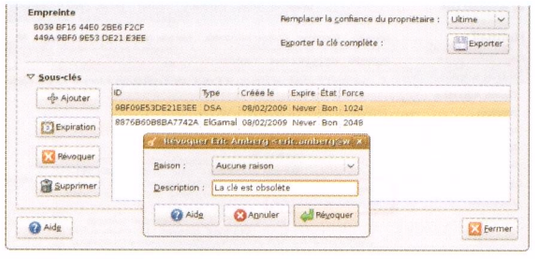
Sinon, cliquez simplement sur **Synchroniser** pour actualiser toutes les clés importées (de plus amples informations un peu plus loin) et transférer la clé personnelle choisie.

En quoi l'**actualisation des clés importées** est-elle importante ? D'autres utilisateurs auraient pu entre-temps signer une clé et, de ce fait, augmenter sa fiabilité. D'un autre côté, la clé aurait pu être annulée (voir la partie suivante). Ce que vous pouvez savoir uniquement si le propriétaire de la clé vous contacte directement ou fournit une mise à jour au serveur de clés.

Votre clé une fois transférée, vous pouvez la rechercher via le menu **Distant**, comme expliqué précédemment – accordez au serveur un peu de temps, il ne s'affiche pas toujours immédiatement.

Annuler une clé

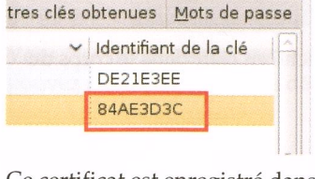
Il peut arriver que, pour une raison ou une autre, votre clé ne soit plus valide (par exemple parce que vous en utilisez une nouvelle ou parce que votre clé privée est tombée entre les mains de quelqu'un). Dans ce cas, vous devez annuler votre ancienne clé. Si vous cliquez sur **Sous-clé** dans l'onglet **Détails** des **Propriétés** de votre clé, vous découvrirez un bouton **Révoquer** :



Ce dernier ne fonctionne malheureusement pas encore très bien – c'est le seul moment auquel nous sommes confrontés à la console et utilisons `gpg` directement. Ouvrez alors un terminal et saisissez la commande suivante qui aura pour effet la révocation d'une clé :

```
# gpg --output Revocation_<Nom>.asc --gen-revoke <identifiant de la clé>
```

Cette commande crée un certificat de révocation pour l'identifiant de clé indiqué. Cet identifiant est indiqué complètement à droite dans la liste des clés :

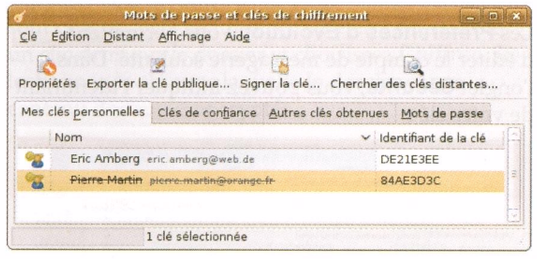


Ce certificat est enregistré dans le fichier indiqué après `--output`. Vous êtes libre de choisir le nom du fichier ; il doit toutefois contenir l'identifiant de clé ou le nom et porter l'extension `.asc`.

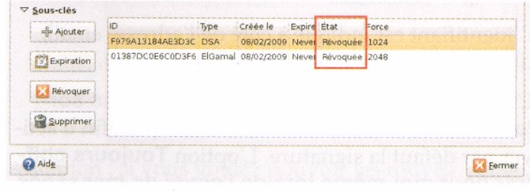
Un assistant vous guide tout au long de la procédure. Pour pouvoir verrouiller la clé, vous devez indiquer le mot de passe de la clé privée.

```
eric@ubuntu810:~$ gpg --output Revocation_Didier_Lheureux.asc --gen-revoke 84AE3D3C

sec  1024D/84AE3D3C 2009-02-08 Pierre Martin <pierre.martin@orange.fr>

Générer un certificat de révocation pour cette clé ? (o/N) o
choisissez la cause de la révocation:
  0 = Aucune raison spécifiée
  1 = La clé a été compromise
  2 = La clé a été remplacée
  3 = La clé n'est plus utilisée
  Q = Annuler
(Vous devriez sûrement sélectionner 1 ici)
Votre décision ? 3
Entrez une description optionnelle ; terminez-là par une ligne vide:
> Ce n'était qu'une clé de test
>
Cause de révocation: La clé n'est plus utilisée
Ce n'était qu'une clé de test
Est-ce d'accord ? (o/N) o

Vous avez besoin d'une phrase de passe pour déverrouiller la
clé secrète pour l'utilisateur: « Pierre Martin <pierre.martin@orange.fr> »
clé de 1024 bits DSA, ID 84AE3D3C, créée le 2009-02-08

sortie avec armure ASCII forcée.
Certificat de révocation créé.

Déplacez-le dans un support que vous pouvez cacher ; si Mallory a
accès à ce certificat il peut l'utiliser pour rendre votre clé
inutilisable.
Une bonne idée consiste à imprimer ce certificat puis à le stocker
ailleurs, au cas où le support devient illisible. Mais attention :
le système d'impression de votre machine pourrait stocker ces
données et les rendre accessibles à d'autres personnes !
```

Vous devez ensuite importer le fichier du certificat avec la révocation en tant que clé. La clé annulée est maintenant barrée dans la liste :



Comme vous pouvez à présent le constater dans les propriétés sous **Détails**, l'état de la clé a été modifié :



Envoyez la clé verrouillée à votre interlocuteur ou bien transférez-la sur un serveur de clés pour actualiser l'état de la clé.

Notez que votre interlocuteur doit d'abord procéder à une synchronisation des clés avant que la révocation n'arrive chez lui. En cas de doute, il est donc préférable de passer par l'information directe via email, etc.

GnuPG et Evolution

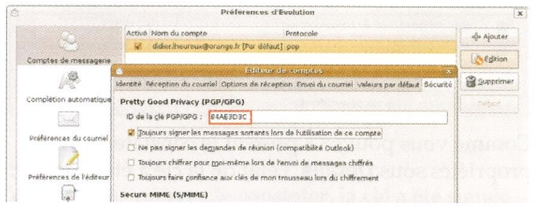
Jouer avec des clés, c'est très amusant, mais il ne s'agit finalement que d'un moyen d'atteindre un objectif. Quel objectif ? Il y en a plusieurs en réalité mais tous se ramènent à la protection de votre vie privée et de vos données.

Considérons d'abord le scénario le plus fréquent : la signature et le cryptage d'emails.

> Pourquoi devrais-je crypter mes emails ? Les emails sont comparables à des cartes postales – n'importe qui peut les lire, y compris sur les différents serveurs par lesquels le courrier transite. Si vous voulez transmettre quelque chose de plus confidentiel, vous n'échapperez pas au cryptage.
>
> La signature numérique assure qu'un email n'a pas été falsifié. Cela évite, par exemple de cliquer sur un lien en toute confiance en pensant que c'est un ami qui nous l'envoie, quand un réalité c'est un email frauduleux.
>
> Falsifier l'adresse d'un expéditeur constitue l'un des exercices les plus simples et une évidence pour les spams. Une signature numérique permet de lever tous les doutes concernant la provenance d'un email.

Evolution est le client de messagerie utilisé par défaut sous Ubuntu. Il est compatible avec GnuPG et peut ainsi créer et ouvrir des courriers cryptés.

Les **Préférences d'Evolution** vous permettent d'éditer le compte de messagerie souhaité. Dans l'onglet **Sécurité**, vous pouvez indiquer l'identifiant de votre clé :

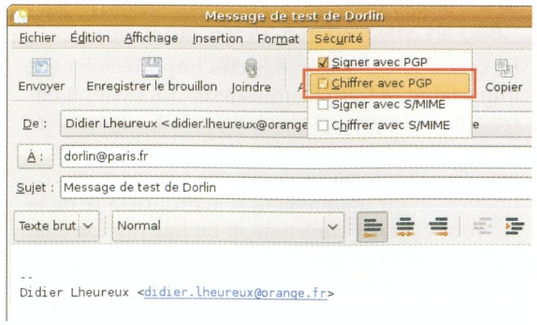


L'identifiant est indiqué à côté de l'adresse email, dans Seashore (et aussi dans les propriétés).

Cliquez sur **Toujours signer les messages sortants lors de l'utilisation de ce compte** afin d'activer par défaut la signature. L'option **Toujours chiffrer pour moi-même lors de l'envoi de messages chiffrés** est particulièrement utile : dans ce cas, la clé publique personnelle est également utilisée pour le cryptage – vous pouvez ainsi continuer à afficher en clair les emails cryptés que vous avez envoyés.

Sinon, vous pouvez procéder au décryptage uniquement à l'aide de la clé publique du destinataire, de telle sorte qu'aucun préréglage des options du compte n'est ici possible.

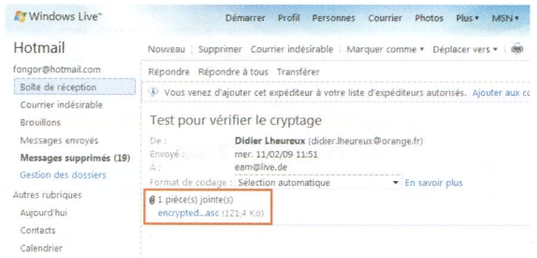
Si vous souhaitez à présent créer un nouvel email, vous pouvez vous rendre dans le menu **Sécurité** et activer l'option **Chiffrer avec PGP** :



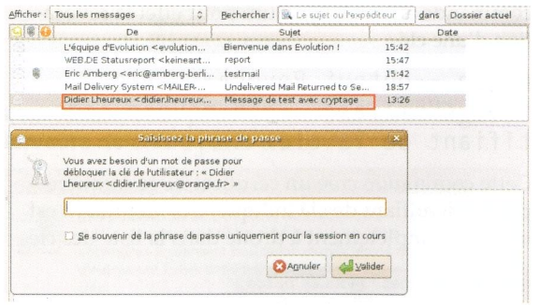
Le message est crypté automatiquement lorsque vous cliquez sur **Envoyer**.

> Pour que le cryptage ait lieu, l'adresse email du destinataire doit concorder avec celle indiquée dans la clé publique.

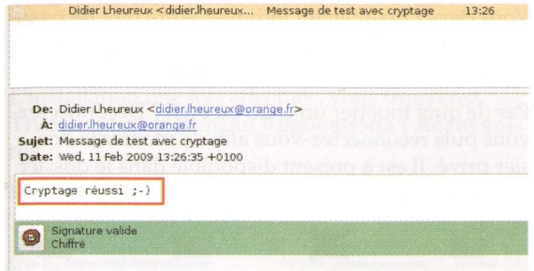
Le texte est ajouté en pièce jointe à l'email (vide) sous forme cryptée et est nommé `encrypted.asc`, comme présenté ici, avec l'exemple d'un fournisseur de messagerie gratuit originaire de Redmond :



Evolution rangera automatiquement au bon endroit chaque email obtenu sous cette forme et demandera la saisie du mot de passe de la clé privée dès que l'email sera cliqué à partir de la liste:



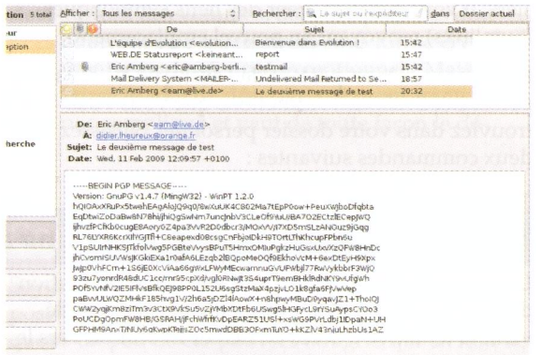
Le texte est ensuite affiché en clair dans le corps de l'email :



Mais si vous obtenez un email crypté de la part d'autres clients de messagerie, les choses peuvent se présenter différemment.

Evolution s'en tient aux standards Internet et indique dans l'en-tête de l'email qu'il s'agit d'un email crypté PGP. Dans d'autres programmes par contre (par exemple les anciennes versions d'Outlook), le cryptage est effectué « inline », c'est-à-dire dans le corps de l'email – dans ce cas de figure, le type de l'email ne permet pas au destinataire d'en deviner le contenu, le décryptage automatique est donc impossible.

Dans ce cas, vous voyez bien le contenu crypté, mais Evolution ne vous propose aucune option concernant le décryptage :



La procédure suivante vous permet de vous sortir de ce mauvais pas : sélectionnez le contenu de l'email et choisissez la commande **Copier** du menu contextuel (et non pas **Enregistrer sous** !). Ouvrez un éditeur de texte et insérez-y à nouveau le contenu. Enregistrez le fichier par ex. sous `crypté.asc` et ouvrez une console. Saisissez maintenant la commande suivante :

```
# gpg --decrypt crypté.asc
```

Vous êtes invité à saisir votre mot de passe. Vous pouvez ensuite voir le texte décrypté :

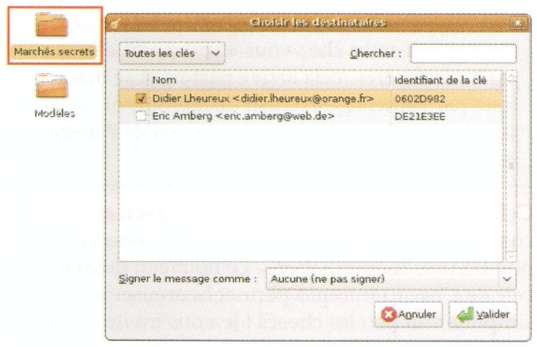
```
eric@ubuntu810:~$ gpg --decrypt crypté.asc

Vous avez besoin d'une phrase de passe pour déverrouiller la
clé secrète pour l'utilisateur: « Didier Lheureux <didier.lheureux@orange.fr> »
clé de 2048 bits ELG-E, ID 19EE5D0F, créée le 2009-02-11 (ID clé principale 435A8A98)

gpg: chiffré avec une clé de 2048 bits ELG-E, ID 19EE5D0F, créée le 2009-02-11
      « Didier Lheureux <didier.lheureux_orange@viereck.fr> »
Content-Type: multipart/signed; micalg="pgp-sha1"; protocol="application/pgp-signature";
        boundary="=-NstLiki0p5yxf1uMTVI4"


--=-NstLiki0p5yxf1uMTVI4
Content-Type: text/plain; charset="UTF-8"
Content-Transfer-Encoding: quoted-printable

Cryptage r=C3=A9ussi ;-)
```

Cette procédure n'est pas particulièrement élégante, mais au moins elle fonctionne.

Crypter des fichiers et des dossiers

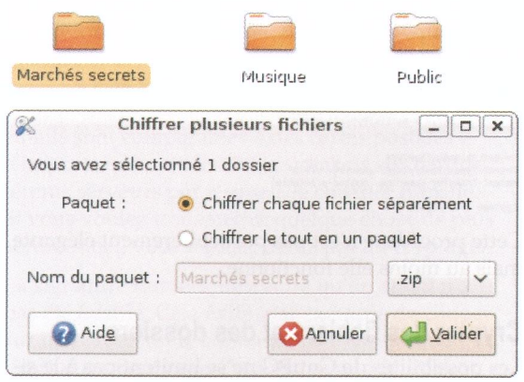
Les possibilités de GnuPG ne se limitent pas à la signature et au cryptage des emails. Supposons que vous ayez un dossier au contenu strictement confidentiel – nommons-le `Marchés secrets` (son nom importe peu).

Vous ne voulez évidemment pas que n'importe qui s'en approche et puisse lire ces données. Vous pouvez empêcher cela d'une part par le biais des droits de l'utilisateur. Mais il peut aussi arriver qu'une personne accède malgré tout à vos fichiers sous l'apparence d'un autre utilisateur – si ces derniers sont cryptés, cela ne lui servira à rien ! Alors, faites un clic droit sur le dossier et choisissez **Chiffrer**. Sélectionnez la clé avec laquelle le dossier et son contenu doivent être cryptés :



Attention : si vous voulez pouvoir les rouvrir vous-même, choisissez *votre clé*. Si vous utilisez la clé d'un correspondant, lui seul pourra ouvrir ces fichiers ou dossiers.

Si dans la boîte de dialogue suivante vous choisissez l'option *Chiffrer chaque fichier séparément*, l'apparence de votre dossier sera alors un peu moins « ordonnée ». La raison : des fichiers temporaires sont également créés (et malheureusement pas supprimés...) en plus des fichiers d'origine. Le dossier contient également les fichiers cryptés, portant l'extension `.pgp`.



Un fichier d'archive non crypté est généré même si vous créez une archive et que vous la cryptez (choix de la deuxième option). C'est à partir de ce fichier d'archive qu'est ensuite créé – en qualité de deuxième fichier – un fichier d'archive crypté, portant lui aussi l'extension `.pgp`. Si vous souhaitez conserver uniquement la version cryptée, vous devez dans tous les cas supprimer le fichier d'origine.

Mais assurez-vous au préalable d'être en mesure de restaurer l'état non crypté : pour ce faire, copiez le fichier PGP dans un autre dossier et essayez de le restaurer à partir de là. Sur la console, vous pouvez à nouveau utiliser la commande `gpg --decrypt <nom du fichier d'archive>`.

> Si vous constatez chez vous « l'évaporation » du jeu de caractères (cela arrive parfois et se manifeste par l'affichage exclusif de hiéroglyphes), saisissez simplement `reset` au niveau de l'invite de saisie.

Cela peut vous sembler un peu inélégant, et je vous comprends ! Cette procédure nécessite encore de nombreuses manipulations. Le nouveau dossier *Private* de Ubuntu Intrepid permet heureusement de simplifier un peu les choses ! Je vous invite à lire la suite...

Le dossier Private

Depuis Intrpid, Ubuntu permet d'utiliser un dossier Private un peu spécifique. Il s'agit en réalité d'un fichier crypté (monté comme un dossier, dans lequel vous pouvez ajouter, supprimer, modifier vos fichiers) par **ecryptfs**. *Ah, et alors ?* Apportons quelques éclaircissements à ces propos abscons :

Commençons par installer les composants nécessaires via le paquet ecrypt-utils :

```
# sudo apt-get install ecryptfs-utils
```

La commande suivante nous permet de créer rapidement et facilement le dossier *Private* :

```
# ecryptfs-setup-private
```

Vous devez maintenant indiquer votre mot de passe et définir le mot de passe pour le montage du dossier – ne l'oubliez surtout pas !

Pas de quoi fouetter un chat, ou bien ? Déconnectez-vous puis reconnectez-vous afin de monter le dossier privé. Il est à présent disponible dans le dossier personnel sous le nom `Private`.

À partir de maintenant, tout ce que vous enregistrerez dans le dossier `Private` sera crypté et plus précisément dans le dossier caché `~/.Private`.

Que devez-vous placer dans ce dossier ? Il est judicieux de transférer les informations sensibles dans le dossier Private, par ex. le dossier `.evolution` contenant les données de vos emails ou `.ssh` qui contient vos clés SSH. Pour que les programmes concernés puissent continuer à accéder à ces dossiers, il vous suffit de créer des liens symboliques (Symlinks) renvoyant au nouvel emplacement.

Examinons cela en partant de l'exemple du dossier personnel *Evolution*. Admettons que vous vous trouviez dans votre dossier personnel. Saisissez les deux commandes suivantes :

```
mv .evolution Private/evolution
ln -s Private/evolution .evolution
```

La première commande déplace le fichier Evolution dans le dossier crypté ; la seconde crée un lien déviant la tentative d'accès au fichier Evolution vers le nouvel emplacement. Les données de vos emails sont désormais en sécurité !

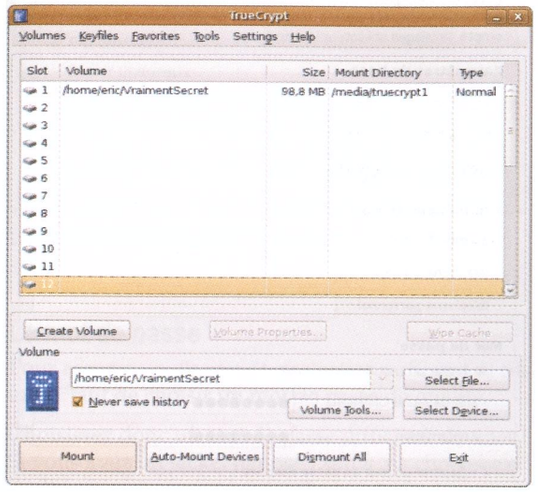
TrueCrypt

Cette technique de cryptage avec **ecryptfs** présente un inconvénient : le contenu des fichiers est bel et bien crypté, mais le nom des fichiers et la structure des dossiers restent visibles dans le dossier Private. Pour obtenir un enregistrement vraiment sûr des données, considérez **TrueCrypt**. Le programme est Open Source et peut être installé facilement sous Ubuntu par le biais d'un fichier d'installation.

Rendez-vous à l'adresse http://www.truecrypt.org/downloads.php et sélectionnez la plateforme appropriée.

Vous devez d'abord décompresser le fichier `.tar.gz` puis démarrer le programme d'installation (le seul présent).

L'installation une fois effectuée, vous trouverez **TrueCrypt** dans le menu **Applications | Accessoires** :



TrueCrypt vous permet vraiment d'effectuer une foule de choses, entre autres créer des fichiers qui deviendront des volumes, dans lesquels des données pourront être enregistrées sous forme cryptée avec un niveau de sécurité exceptionnel ! Mais je vous laisse le soin de découvrir ce programme. Il n'est pas compliqué et vaut de toute façon le détour !

Conseils de sécurité

Soyons honnêtes : nous connaissons tous les risques possibles concernant la sécurité et nous nous moquons un peu de ces pauvres bougres qui en font les frais, faute d'avoir pris certaines mesures de précaution. Mais faisons-nous vraiment mieux ?

> Créer une protection de base n'est guère compliqué – il suffit souvent de quelques petites mesures pour augmenter considérablement le niveau de protection !

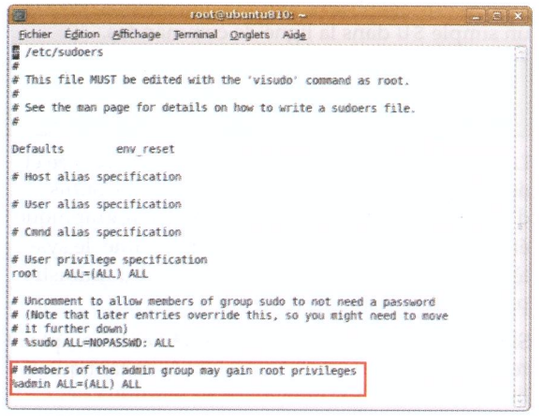
Sudo et les droits de l'utilisateur

Au fil des pages, nous avons été plusieurs fois confrontés à `sudo`. Grâce à ce concept, Ubuntu dispose d'un argument clé encore négligé par les autres grands distributeurs tels qu'openSUSE ou Fedora.

L'utilisateur créé lors de l'installation est votre utilisateur par défaut, avec lequel vous vous connectez au système et travaillez. Cet utilisateur dispose de droits limités, il ne peut par exemple rien installer, ni modifier aucune configuration du système. Seul **root** y est autorisé, l'administrateur système.

Pourtant, vous avez malgré tout continué à installer, configurer, etc ? C'est grâce à `sudo`. C'est très simple : après avoir saisi la commande `sudo`, tous les utilisateurs apparaissant sous une forme ou une autre dans le fichier `/etc/sudoers` disposent de droits supplémentaires et peuvent exécuter des tâches qui leur sont inaccessibles en tant qu'utilisateur normal.

Par défaut, un utilisateur nouvellement enregistré n'a pas le droit d'exécuter `sudo` et donc d'obtenir des privilèges d'administration. La ligne du fichier `/etc/sudoers` chargée de l'extension des privilèges est mise en relief dans l'illustration suivante :



La ligne commençant par % indique les groupes d'utilisateurs, ici le groupe `admin`. Grâce à l'indication `ALL=(ALL) ALL`, les membres de ce groupe sont autorisés à tout lorsqu'un `sudo` est effectué. Ils ne sont alors soumis à aucune restriction.

> Par défaut, les membres de ce groupe sont constitués uniquement des utilisateurs normaux que vous avez créés lors de l'installation. Les utilisateurs créés ultérieurement ne font pas partie du groupe `admin`. Dans la partie suivante, je vous expliquerai comment modifier cela.

Vous savez déjà comment utiliser `sudo` :

```
$ sudo <commande>
```

Devoir commencer systématiquement par saisir cette commande, c'est relativement usant. Utilisez ici la commande suivante, que vous n'avez à taper qu'une seule fois. (Je l'avais déjà mentionné au début du manuel.)

```
$ sudo -s
```

Elle permet de disposer des privilèges pour la durée de la session du terminal. L'invite devient une invite

root, reconnaissable au signe dièse (#). Pour quitter le mode root, il vous suffit de saisir `exit`.

Activer le compte root

root, l'administrateur système à proprement parler, est désactivé par défaut : vous ne pouvez pas vous connecter en tant que root. Cela s'explique par le fait qu'aucun mot de passe n'a été attribué à *root*. Vous pouvez modifier cela à l'aide de la ligne de commande suivante :

```
$ sudo passwd
```

Si aucun autre paramètre n'est ajouté à la commande `passwd`, c'est le mot de passe de l'utilisateur root qui sera défini. Vous pouvez désormais vous connecter en tant que *root*, ou devenir *root* via un simple `su` dans la ligne de commandes.

> Demandez-vous néanmoins si vous avez vraiment besoin d'activer root avant de le faire.

La fonctionnalité sudo est malgré tout conservée et ne doit pas être désactivée, la majorité des outils d'administration de l'interface utilisateur graphique étant basés `sudo`. Sans cette fonctionnalité, le système peut dans certains cas ne plus être administré par l'interface graphique !

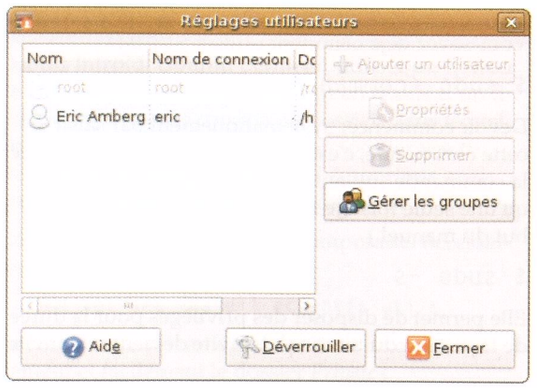
Si vous souhaitez par la suite désactiver le compte root, saisissez la commande suivante en tant qu'utilisateur normal :

```
$ sudo passwd -l root
```

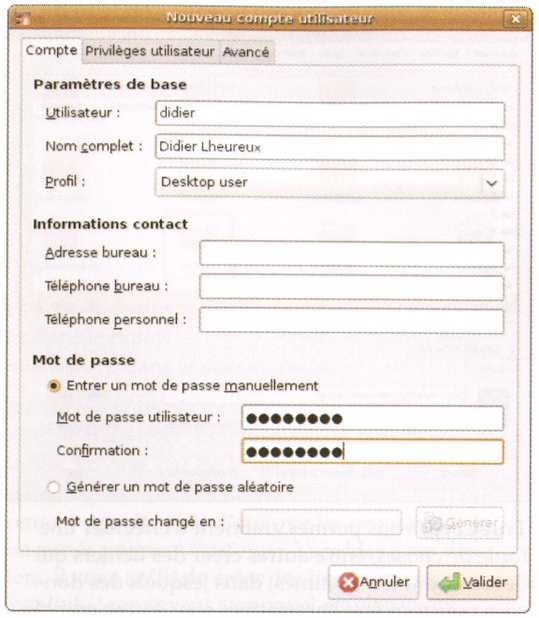
Gestion des utilisateurs et des groupes

Linux est un système multi-utilisateurs. Vous disposez de votre compte, mais si d'autres personnes veulent utiliser le même ordinateur vous ne devez pas les laisser accéder à votre compte, elles doivent disposer du leur. Linux est prévu pour ça, depuis toujours.

Rendez-vous pour cela dans le menu **Système | Administration | Utilisateur et groupes** :



Comme vous n'avez le droit de rien faire pour le moment, commençons par activer le mode d'administration système en cliquant sur le bouton **Déverrouiller**. Comme d'habitude, il vous faut saisir le mot de passe. Vous pouvez ensuite cliquer sur le bouton **Ajouter un utilisateur** et faire ceci :



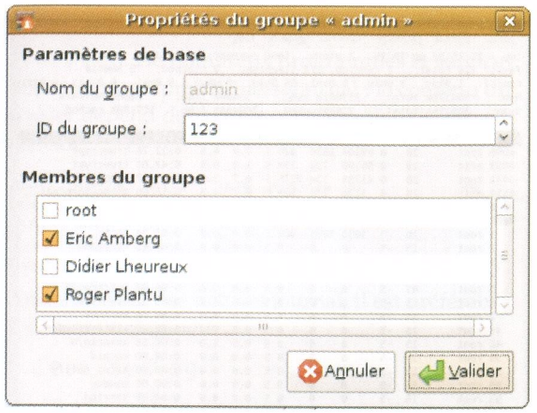
Indiquez les données correspondantes ; les **Informations de contact** ne sont là qu'à titre informatif, elles ne sont pas utilisées par le système.

Le profil permet de déterminer les droits de l'utilisateur. **Desktop user** est un simple utilisateur.

> Notez qu'un tel utilisateur n'a pas la possibilité de travailler avec `sudo` car, à l'origine, il n'est pas membre du groupe `admin`.

Avec le profil **Administrator**, par contre, l'utilisateur est automatiquement membre du groupe `admin`, il peut donc utiliser `sudo`. Ne vous inquiétez pas, tout cela pourra être encore adapté ultérieurement.

Si vous avez créé un utilisateur non administrateur, vous pouvez modifier son affiliation à un groupe en cliquant sur **Gérer les groupes**. Recherchez le groupe `admin`, cliquez sur **Propriétés** et ajoutez l'utilisateur souhaité en plaçant une encoche :



Le nouvel utilisateur peut désormais aussi utiliser `sudo`.

Le mot de passe

Pour assurer un niveau de protection suffisant, il est important de choisir un mot de passe de qualité.

Choisissez donc un mot de passe difficile à deviner ! Si vous avez un teckel baptisé *Dalton*, ne choisissez en aucun cas le mot de passe `Dalton`, `Dalton123` ou `DaltonDalton` etc...

Un bon mot de passe doit répondre aux critères suivants :

1. Une longueur de 8 caractères, de préférence 12.
2. Il ne doit pas avoir de sens. Sinon il y a le risque qu'un petit curieux le devine ou qu'il figure déjà dans des listes de cracks de mots de passe pouvant comporter des millions d'enregistrements.
3. Il ne doit pas s'agir de permutations (simples) d'une notion connue.
4. Le mot de passe doit contenir des majuscules et des minuscules, des chiffres et des caractères spéciaux.
5. Le mot de passe doit être une chaîne de caractères d'apparence aléatoire.
6. Le mot de passe doit être changé régulièrement (par exemple tous les trois mois).

Comment trouver un bon mot de passe ? Vous manquez d'inspiration ? Moi aussi ! J'ai recours à l'astuce suivante : j'imagine une phrase tout ce qu'il y a de plus classique contenant l'un ou l'autre chiffre, je garde les initiales, j'y insère un ou deux caractères spéciaux, et le mot de passe est terminé. Un exemple :

Ma tante Sophie a son anniversaire le 8.4. et a des cheveux rouges.

On obtient le mot de passe suivant :

MtS@sal84eadcr.

J'ai ici tout simplement remplacé le **a** par un **@**. Voici quelques suggestions de remplacement judicieux :

a, A - @

1, l, i - !

S, s - $ ou §

E, e - €

Espace – par un point (.), un tiret (-) ou un signe de soulignement (_)

D'autres astuces relatives à l'utilisation d'un mot de passe :

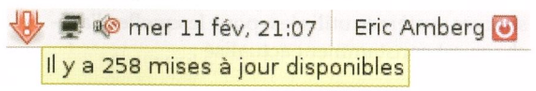
- Pensez à changer régulièrement votre mot de passe, cela permet d'augmenter considérablement le niveau de sécurité.
- Ne transmettez jamais votre mot de passe à une tierce personne !
- N'écrivez pas votre mot de passe !
- N'utilisez pas partout le même mot de passe – utilisez un mot de passe individuel, au moins pour les applications critiques.

En outre : la fonction d'enregistrement du mot de passe proposée par les navigateurs doit être utilisée uniquement pour les applications non critiques, car elles ne sont pas protégées à 100%. En outre, on oublie d'autant plus vite des mots de passe lorsque l'on n'est pas contraint de les mémoriser.

Mises à jour

Il est impératif que votre système soit continuellement actualisé et que toutes les mises à jour de sécurité soient effectuées : si Linux est un système très sécurisé, c'est justement grâce aux mises à jour ! Rassurez-vous, contrairement à un autre système d'exploitation bien connu, il est rarissime qu'une mise à jour demande de redémarrer l'ordinateur. Il n'y a donc aucune gêne, vous pouvez continuer à utiliser l'ordinateur.

La vérification automatique des mises à jour proposée par Ubuntu simplifie considérablement la tâche, mais c'est quand même à vous de passer à l'action ! Dès que vous voyez la flèche rouge pointant vers le bas et ornée d'un point d'exclamation :



Ne soyez pas impressionné par le nombre des mises à jours. C'est pas toujours comme ça. Et, comme je disais, on doit rarement redémarrer et elles ne vous empêcheront pas de travailler.

Les mises à jour de sécurité permettent justement de remédier aux failles de sécurité qui, sans cela, pour-

raient être utilisées à votre insu. Il m'est arrivé une fois d'oublier de charger à temps les correctifs pour mon serveur, résultat : mon serveur root a été soudainement utilisé en tant que serveur de téléchargement pour des logiciels douteux ! Alors, ne vous moquez pas des autres, et agissez...

Le bon sens : les fraudes

Quiz : que feriez-vous si un jour un homme sonnait à votre porte et se présentait en ces termes : « Bonjour, je suis envoyé par votre banque et je dois vérifier votre compte. Veuillez me remettre votre carte de crédit et votre code secret, je règlerai rapidement le problème et vous la rendrai immédiatement après. Faute de quoi, votre compte sera malheureusement bloqué ! »

Vous souriez ? Bien, mais ne soyez pas si sûr de vous ! Les escrocs déploient des astuces de plus en plus subtiles pour nous leurrer et nous convaincre de leur donner des informations confidentielles : faux email de la banque, des sites de vente en ligne, de sites de paiement, etc. Moi-même, je dois parfois y regarder à deux fois avant de reconnaître un email frauduleux.

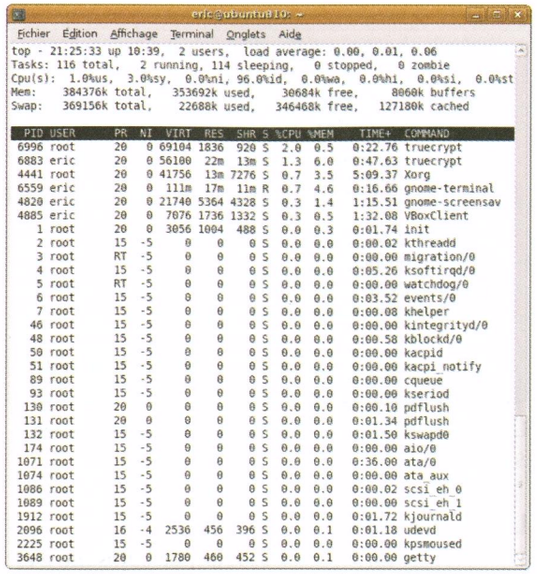
En résumé : restez toujours attentif aussi bien sur Internet que dans la vie réelle ! Ne cliquez pas sans réfléchir sur n'importe quel lien contenu dans un email. *Personne, aucun site Web, ne devrait vous demander par email un mot de passe ou un code PIN, et certainement pas votre banque !*

En cas de doute, contactez directement la banque, le site et pas en cliquant sur le bouton répondre : vous risquez d'écrire au fraudeur.

Savoir ce qui fonctionne sur son système

Vous rappelez-vous du passage dans lequel je vous présentais les principaux services et processus ? Consultez fréquemment votre liste de processus et répertoriez régulièrement les services et processus qui y figurent. Si par la suite votre système commence à se comporter de façon étrange, vous retrouverez plus facilement le processus ou service susceptible d'être à l'origine du trouble.

`top` est un outil de ligne de commandes particulièrement pratique dans ce cas de figure. Il vous indique les processus actuels tout en précisant les ressources systèmes utilisées, pour lesquelles l'affichage est constamment actualisé :



```
eric@ubuntu810: ~
Fichier  Edition  Affichage  Terminal  Onglets  Aide
top - 21:25:33 up 10:39,  2 users,  load average: 0.00, 0.01, 0.06
Tasks: 116 total,   2 running, 114 sleeping,   0 stopped,   0 zombie
Cpu(s):  1.0%us,  3.0%sy,  0.0%ni, 96.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:    384376k total,   353692k used,    30684k free,     8060k buffers
Swap:   369156k total,    22688k used,   346468k free,   127180k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 6996 root      20   0 69104 1836  920 S  2.0  0.5   0:22.76 truecrypt
 6883 eric      20   0 56100  22m  13m S  1.3  6.0   0:47.63 truecrypt
 4441 root      20   0 41756  13m 7276 S  0.7  3.5   5:09.37 Xorg
 6559 eric      20   0  111m  17m  11m R  0.7  4.6   0:16.66 gnome-terminal
 4820 eric      20   0 21740 5364 4328 S  0.3  1.4   1:15.51 gnome-screensav
 4885 eric      20   0  7076 1736 1332 S  0.3  0.5   1:32.08 VBoxClient
    1 root      20   0  3056 1004  488 S  0.0  0.3   0:01.74 init
    2 root      15  -5     0    0    0 S  0.0  0.0   0:00.02 kthreadd
    3 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 migration/0
    4 root      15  -5     0    0    0 S  0.0  0.0   0:05.26 ksoftirqd/0
    5 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 watchdog/0
    6 root      15  -5     0    0    0 S  0.0  0.0   0:03.52 events/0
    7 root      15  -5     0    0    0 S  0.0  0.0   0:00.08 khelper
   46 root      15  -5     0    0    0 S  0.0  0.0   0:00.00 kintegrityd/0
   48 root      15  -5     0    0    0 S  0.0  0.0   0:00.58 kblockd/0
   50 root      15  -5     0    0    0 S  0.0  0.0   0:00.00 kacpid
   51 root      15  -5     0    0    0 S  0.0  0.0   0:00.00 kacpi_notify
   89 root      15  -5     0    0    0 S  0.0  0.0   0:00.00 cqueue
   93 root      15  -5     0    0    0 S  0.0  0.0   0:00.00 kseriod
  130 root      20   0     0    0    0 S  0.0  0.0   0:00.10 pdflush
  131 root      20   0     0    0    0 S  0.0  0.0   0:01.34 pdflush
  132 root      15  -5     0    0    0 S  0.0  0.0   0:01.50 kswapd0
  174 root      15  -5     0    0    0 S  0.0  0.0   0:00.00 aio/0
 1071 root      15  -5     0    0    0 S  0.0  0.0   0:36.00 ata/0
 1074 root      15  -5     0    0    0 S  0.0  0.0   0:00.00 ata_aux
 1086 root      15  -5     0    0    0 S  0.0  0.0   0:00.02 scsi_eh_0
 1089 root      15  -5     0    0    0 S  0.0  0.0   0:00.00 scsi_eh_1
 1912 root      15  -5     0    0    0 S  0.0  0.0   0:01.72 kjournald
 2096 root      16  -4  2536  456  396 S  0.0  0.1   0:01.18 udevd
 2225 root      15  -5     0    0    0 S  0.0  0.0   0:00.00 kpsmoused
 3648 root      20   0  1780  460  452 S  0.0  0.1   0:00.00 getty
```

Par défaut, le processus accaparant le plus le processeur est affiché tout en haut. **q** vous permet de quitter le programme.

> Si un processus vous semble suspect, lancez une *recherche dans Google*. Vous trouverez des renseignements pour chaque processus.

Du côté des anti-virus et des pare-feu

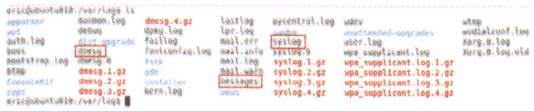
Ces mesures de sécurité obligatoires sur les systèmes Windows sont généralement inutiles sur Linux. Vous trouverez bien des programmes anti-virus pour Linux, mais leur rôle consiste généralement à rechercher des virus pour… Windows. Ils sont en principe installés sur des serveurs de fichiers.

Sous Windows, un pare-feu est chargé essentiellement de deux choses :

1. Bloquer les tentatives d'intrusions provenant de l'extérieur qui tentent d'utiliser les services actifs. Par défaut, Ubuntu ne propose pratiquement aucun service accessible, ce dont témoigne la commande `netstat -tulpen`.
2. Bloquer l'accès à des serveurs sur Internet à partir de programmes ou processus locaux, cherchant à communiquer des informations espionnées. Ce type de programme n'existe pas sous Linux – c'est le rôle des responsables des dépôts d'en vérifier l'intégrité.

Parle-moi ! Les fichiers journal

Les systèmes Linux disposent généralement d'un très bon suivi des événements, ce que l'on appelle un système de logging ou de journalisation. Les fichiers journal se trouvent en général sous `/var/log`. De nombreux programmes et services créent leurs propres fichiers journal ; le contenu du dossier dépend donc de la configuration de votre système.



```
eric@ubuntu810:/var/log$ ls
apparmor   dmesg.4.gz   lastlog   syscentral.log  udev          wtmp
apt        dpkg.log     lpr.log   syslog          ubiquity      wvdialconf.log
auth.log   faillog      mail.err  syslog.1        ubuntu-upgrade-xorg.0.log
boot       fontconfig.log mail.info syslog.2.gz   wpa_supplicant.log  Xorg.0.log.old
bootstrap.log  fsck     mail.log  syslog.3.gz     wpa_supplicant.log.1.gz
btmp       gdm          mail.warn syslog.4.gz     wpa_supplicant.log.2.gz
ConsoleKit messages     messages  ubuntu-upgrade  wpa_supplicant.log.3.gz
cups       mysql.log    news      vmware          wpa_supplicant.log.4.gz
eric@ubuntu810:/var/log$
```

Pour invoquer des fichiers journal, il est préférable d'utiliser un pagineur :

```
$ less syslog
```

Si le fichier est trop gros, vous pouvez limiter l'affichage aux derniers enregistrements, en indiquant le nombre de lignes à partir du bas :

```
$ tail -n40 syslog
```

Mais il y a mieux encore ! Si vous voulez « vivre » en direct l'apparition des événements, vous pouvez aussi ouvrir le fichier et suivre chaque enregistrement en temps réel :

```
$ tail -f syslog
```

Pour quitter, appuyez sur **CTRL+C**

Si vous recherchez des erreurs, soyez particulièrement attentif aux fichiers journal suivants :

`dmesg` – contient la liste des événements qui se sont produits au démarrage. Vous pouvez donc voir si, par ex., un composant matériel bien précis a été trouvé et initialisé.

`syslog` – le fichier journal système le plus important d'Ubuntu. Vous trouverez ici la plupart des événements du système. Inconvénient : il est souvent très long et peut éventuellement nécessiter une recherche via grep. Je vais vous montrer comment simplifier la recherche avec grep.

`messages` – le fichier journal traditionnel sous Linux est aussi utilisé intensivement sous Ubuntu ; mais il contient souvent les mêmes événements que syslog.

> En cas de doute, vérifiez également si un fichier journal ou un dossier porte le même nom que les composants que vous souhaitez examiner – comme dit : les services s'occupant de vos fichiers journal sous Linux sont nombreux.

Bon, comment fonctionne la commande `grep` ? Admettons que vous recherchiez l'expression *bluetooth*, votre recherche pourrait ressembler à ceci :

```
$ grep -B5 -i bluetooth syslog
```

L'option `-B5` permet d'afficher les 5 lignes précédant l'occurrence trouvée. Cela vous aide à juger de la pertinence de l'enregistrement. Avec `-i`, la commande `grep` ignore les majuscules et les minuscules. Si vous voulez afficher également quelques lignes après l'occurrence trouvée, ajoutez `-A<nombre de lignes>` de la même façon qu'avec `-B`, soit par ex. :

```
$ grep -A2 -B5 -i bluetooth syslog
```

Vous voilà déjà avec une bonne quantité de « matière » entre les mains.

> **Astuce** : si vous ouvrez un fichier avec `vim`, vous pouvez taper **/** pour ensuite rechercher une expression. **n** vous permet de passer à l'occurrence suivante.

Sauvegardes

C'est l'histoire d'un homme qui saute du 50ème étage ! Et qui dit, à chaque étage : « Jusqu'ici, tout va bien... »

Personne – ni vous, ni moi – n'est à l'abri de la perte de données ! Une personne qui a enregistré des données importantes sur son ordinateur et qui ne procède pas à des sauvegardes régulières fait preuve d'une grave négligence.

Dans les pages suivantes, nous allons parler de sauvegardes :

- quelles données devez-vous sauvegarder,
- quelles stratégies adopter,
- sur quels médias effectuer les sauvegardes,
- comment se déroule la sauvegarde et
- comment automatiser des sauvegardes ?

Quels éléments sauvegarder ?

L'objectif consiste en principe à pouvoir restaurer le plus rapidement possible le système et les principales données en cas de catastrophe.

Il est facile de réinstaller Linux depuis le CD d'installation, mais comment récupérer vos données personnelles et tous vos réglages sans devoir tout réinstaller?

La première question à se poser c'est de savoir où ces données peuvent bien être enregistrées ?

Voici les dossiers les plus importants pour la sauvegarde des données :

`/home` – c'est ici que se trouvent les dossiers personnels de tous les utilisateurs. Chaque utilisateur y dispose d'un dossier à son nom. Il contient tous ses documents.

`/root` – si vous l'avez activé et si vous l'utilisez, le dossier de root n'est pas dans /home. Normal, ce n'est pas un utilisateur au sens strict.

`/etc` – c'est ici que sont stockés quasiment tous les fichiers de configuration du système. Si vous souhaitez les conserver, vous devez sauvegarder l'intégralité du dossier.

`/var/backups` – c'est ici que sont localisées les sauvegardes (internes) créées par le système.

`/var/mail` – il contient traditionnellement les emails des utilisateurs. Il est utile sous Ubuntu uniquement si vous avez installé un serveur de messagerie. Si vous vous demandez de quoi je parle, c'est sans doute que vous n'en avez pas installé un ;-)

`/var/log` – c'est ici que sont enregistrés les fichiers journal. Nous avons déjà mentionné l'importance des fichiers journal.

`/usr/local` – c'est dans ce dossier que sont parfois installés des programmes n'appartenant pas à la distribution. Conformément aux conventions du système de fichiers Linux, en qualité d'utilisateur, vous êtes également tenu d'installer ici vos propres programmes. Ces programmes contiennent souvent des fichiers de configuration enregistrés ici et non pas sous `/etc`. D'autres fichiers variables peuvent aussi être enregistrés sous le dossier d'installation du programme, sous `/usr/local`. Si vous trouvez ici des données (non pas uniquement des dossiers vides !), vous devez donc également les inclure à la sauvegarde.

`/boot/grub/menu.lst` – il s'agit du fichier de configuration de Grub, le chargeur d'amorçage (Bootloader). Si vous avez procédé à des modifications, vous souhaiterez peut-être aussi le sauvegarder ce fichier, mais prudence : la restauration fonctionne uniquement sur un système utilisant *exactement* le même noyau !

> **Un autre conseil** : si des fichiers peuvent facilement être récupérés sur Internet, il n'est pas forcément utile de les sauvegarder. Vous gagnerez du temps et de la place.

Stratégies de sauvegarde

Automatisation et fréquence

Une bonne sauvegarde est automatisée. Une sauvegarde manuelle n'est pas fiable. Si vous devez penser à la faire, vous oublierez et c'est (évidemment) à ce moment-là que vous en aurez besoin. Croyez-moi.

> NDLE : fréquence et nombre de sauvegardes
>
> La fréquence des sauvegardes dépend de votre utilisation de l'ordinateur. 1 fois par heure, par jour, par semaine, par mois, etc.
>
> Une seule sauvegarde ne suffit pas. Du moins ça ne suffit pas si vous tenez vraiment à vos données. Ici, à Compétence Micro, les données sont sauvegardées sur 4 supports différents, locaux et distants.
>
> *Qui a dit que l'informatique était une chose simple et qui ne coûtait pas cher ?* Demandez-vous toujours ce que valent vos données. Si elles valent moins que le prix d'un disque dur ou deux…

Types de sauvegardes

Il existe différents types de sauvegardes que je souhaite vous présenter brièvement :

Sauvegarde complète : toutes les données sont sauvegardées. Elle utilise naturellement plus de place.

Sauvegarde incrémentale : seules les données modifiées depuis la dernière sauvegarde sont sauvegardées.

Vous pouvez par exemple effectuer une sauvegarde complète le dimanche, et une sauvegarde incrémentale les autres jours de la semaine. Vous avez ainsi une sauvegarde pour chaque jour, mais vous devrez éventuellement restaurer plusieurs sauvegardes (celle du dimanche, puis celles du lundi et du mardi, par exemple, pour retrouver votre système tel qu'il était mardi. C'est plus long à restaurer.

Sauvegarde différentielle : toutes les données modifiées depuis la dernière sauvegarde complète sont sauvegardées. La sauvegarde différentielle ne nécessite pas autant d'espace mémoire que la sauvegarde complète et permet en outre une restauration vraiment rapide des données.

Si vous souhaitez économiser de la place, vous utiliserez davantage la sauvegarde incrémentale ; pour une restauration rapide des données, une sauvegarde complète récente est naturellement plus judicieuse.

Supports de sauvegarde

Pour être fiable une sauvegarde doit idéalement être facile d'accès et… située à autre endroit que l'ordinateur.

- Facile d'accès : vous n'avez qu'à tendre le bras, insérer un DVD ou brancher un disque dur pour récupérer une sauvegarde en cas de besoin.
- Ailleurs : en cas d'incendie, de vol, etc. la seule bonne sauvegarde est celle qui se trouvait ailleurs…

Différentes possibilités s'offrent à vous :

Deuxième disque dur : si vous disposez d'un deuxième disque dur interne que vous n'utilisez pas pour le moment, vous pouvez le monter par exemple sous `/backup` et y placer les sauvegardes.

Mais si l'ordinateur est détruit ou volé, vous perdez aussi votre sauvegarde.

Disque dur externe : l'avantage des médias externes est qu'ils peuvent être à tout moment déplacés et stockés ailleurs. L'inconvénient est qu'ils ne sont pas toujours disponibles. Un disque dur externe se révèle malgré tout un média de sauvegarde pratique dans de nombreux cas.

Serveur FTP/SSH/… : FTP est un moyen bien connu de sauvegarde, mais une mauvaise solution pour ce qui est de la sécurité (pour cela, optez pour SSH). Nécessite en plus une connexion réseau. Il s'agit toutefois d'une variante valide pour ce qui est des sauvegardes internes.

CD/DVD : très pratique, mais leur capacité est limitée et vous devez l'insérer et le retirer manuellement. L'avantage est la simplicité de l'archivage. Des CD ou des DVD sur l'étagère et c'est fini.

> Attention au vieillissement des disques cependant. Vérifiez régulièrement l'état de vos CD et DVD. Là aussi, deux copies (sur des disques de marques différentes) valent mieux qu'une.

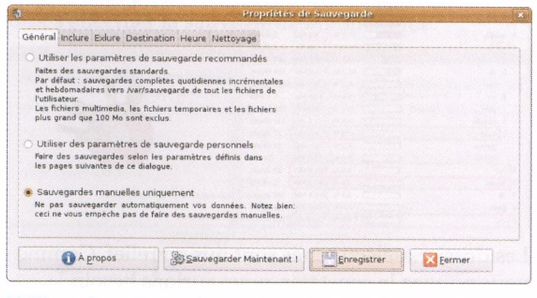
Sauvegarder

Comment ? En copiant simplement tous les fichiers à sauvegarder ? Il doit certainement exister une manière plus élégante, non ?

Effectivement ! Je vais vous présenter un très bon programme de sauvegarde pour PC de bureau. Il se nomme **sbackup** et peut être installé facilement via Synaptic. Vous le trouverez ensuite dans **Système/Administration**, sous le nom **Simple Backup Config** ou **Simple Backup Restore**.

Préparer la sauvegarde

Ouvrez **Simple Backup Config**. Le programme de sauvegarde devant être utilisé avec des privilèges d'administration, il convient une fois de plus de commencer par saisir le mot de passe.



Utiliser les paramètres de sauvegarde recommandés vous permet de conserver les paramètres conseillés par défaut. Une sauvegarde complète sera alors effectuée chaque semaine et des sauvegardes incrémentales chaque jour. Tous les dossiers importants sont inclus. Notez que les fichiers multimédia, les fichiers temporaires et les gros fichiers de plus de 100 Mo sont exclus.

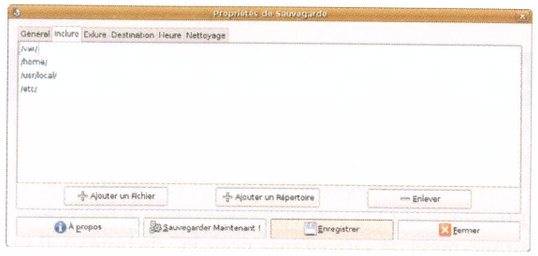
Mis à part la sauvegarde de l'intégralité du dossier `/var` et du stockage des sauvegardes dans `/var/backup`, ce préréglage est relativement judicieux. Mais si vous avez monté sous ce dossier

une autre ressource physique (par ex. un deuxième disque dur) disposant de suffisamment d'espace disque, vous pouvez effectuer la sauvegarde de cette façon.

Utiliser des paramètres de sauvegarde personnels. Vous pouvez définir vous-même tous les paramètres par le biais des onglets suivants. C'est celle que nous utiliserons.

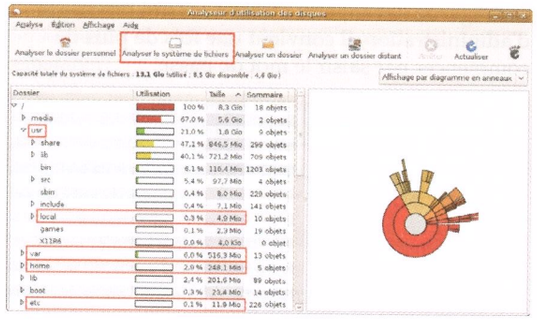
La dernière option est **Sauvegardes manuelles uniquement**. Elle permet d'effectuer une sauvegarde manuelle, sans avoir à se soucier d'aucune planification.

Choisissez la seconde option et cliquez sur l'onglet **Inclure** :



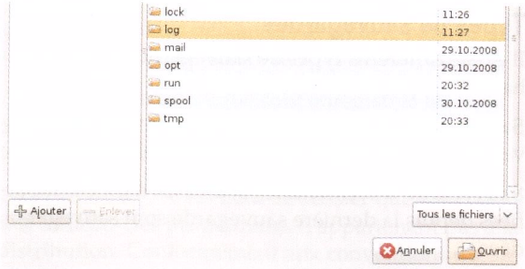
Vous voyez ici les dossiers dont le contenu sera sauvegardé.

Pour voir ce que nous obtiendrons avec ces valeurs, allez dans le menu **Applications / Accessoires** et ouvrez l'outil **Analyseur d'utilisation des disques**. Cliquez sur **Analyser le système de fichiers** et attendez l'affichage des statistiques :

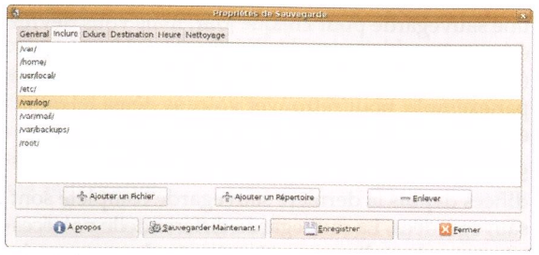


Les dossiers sélectionnés sont mis en relief. Comme vous pouvez le constater, nous aurions besoin d'environ 800 Mo (ce n'est pas mon vrai compte d'utilisateur. En réalité, vous taquinerez très rapidement les Go en utilisant vraiment votre ordinateur...).

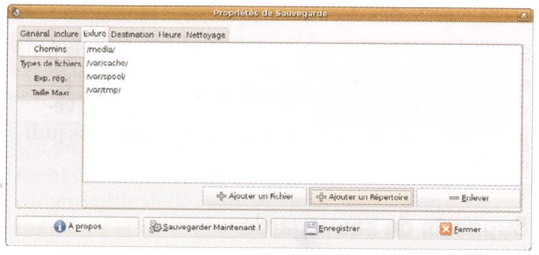
Pour économiser de la place, revenez dans le programme de configuration et cliquez soit sur **Ajouter un fichier, Ajouter un Répertoire**, soit sur **Enlever**. Sélectionnez ensuite le fichier ou le dossier à ajouter, ou à supprimer de la sauvegarde et cliquez sur **Ouvrir** :



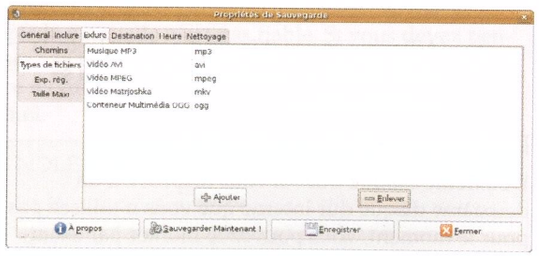
Au final, votre liste doit ressembler plus ou moins à ceci :



Passons à l'onglet suivant, à savoir **Exclure** :

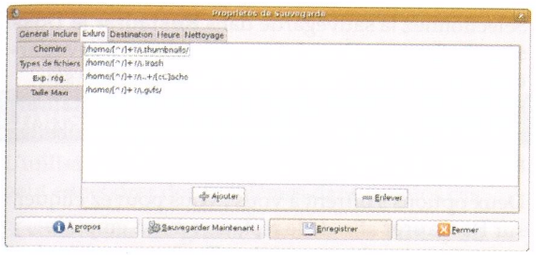


Indiquez ici les dossiers et fichiers que vous voulez explicitement exclure de la sauvegarde. Le dossier `/media` contient tous les médias amovibles connectés à l'ordinateur, de ce fait ils sont exclus. Le sous-élément **Types de fichiers** est particulièrement intéressant :



Vous trouvez ici tous les types de fichiers qui ne sont **pas** sauvegardés, *même s'ils figurent dans des dossiers à sauvegarder*. Comme vous le constatez, il s'agit de fichiers multimédia et d'images ISO. Vous pouvez bien sûr aussi adapter cette liste. Pour sauvegarder les MP3, sélectionnez-le dans la liste et cliquez sur **Enlever**.

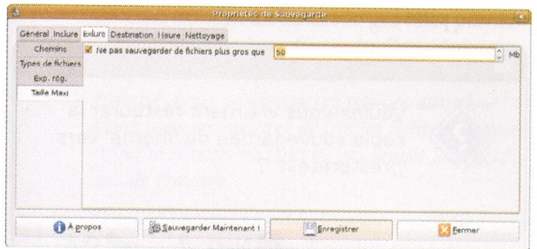
Exp. rég. vous permet d'utiliser des expressions régulières :



Les fichiers ou dossiers auxquels correspond le motif de recherche sont exclus de la sauvegarde. Cette fonctionnalité s'adresse uniquement aux utilisateurs expérimentés,

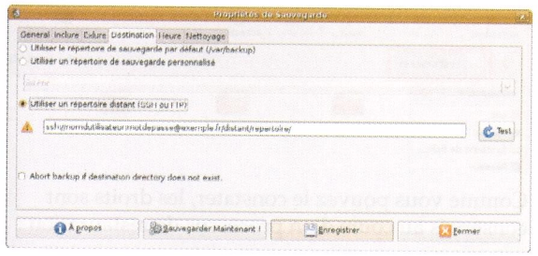
> Les types de fichiers prédéfinis sont des fichiers temporaires qui n'ont pas besoin d'être sauvegardés ;-).

Sous **Taille Maxi**, vous indiquez la taille maximale des fichiers à sauvegarder :

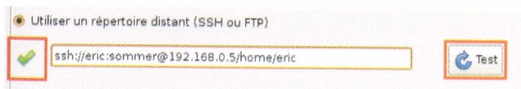


Le préréglage est de 10 Mo, j'ai pour ma part défini une valeur de 50 Mo, ce qui me convient parfaitement. Mais pour être tout à fait sûr, désactivez cette option en la décochant.

Poursuivons avec l'onglet **Destination** :



Si vous ne vous en tenez pas précisément à l'emplacement de sauvegarde par défaut (`/var/backup`), vous pouvez soit choisir un emplacement local (y compris des CD/DVD, mais cela a toujours généré des erreurs chez moi !), soit enregistrer la sauvegarde sur un serveur SSH ou FTP. Indiquez les paramètres de connexion et cliquez sur le bouton **Test** pour voir si ça marche :



Il est maintenant temps d'effectuer une sauvegarde de test. Mais attention : avant de cliquer sur **Sauvegarder maintenant !**, n'oubliez pas de cliquer sur **Enregistrer** afin de mémoriser votre nouvelle configuration.

> *Tester une sauvegarde ?* Ca en fait des manipulations, vous me direz. Encore une fois, il faut se demander pour quoi on fait des sauvegardes. Si c'est pour récupérer nos données en cas de souci, il vaut mieux être sûr que ça marche : que les données sont bien sauvegardées et qu'on pourra y accéder le cas échéant.

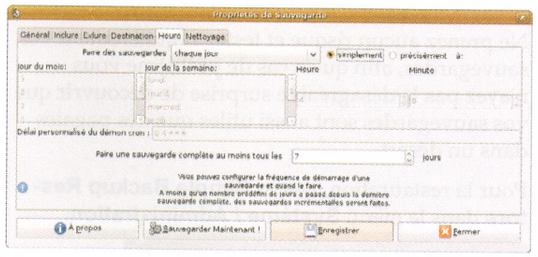
Le processus de sauvegarde lancé, sbackup vous indique l'ID du processus sous lequel s'effectue la sauvegarde en arrière-plan. La sauvegarde une fois terminée, vous la trouverez dans le dossier de sauvegarde que vous aurez indiqué.

```
eric@ubuntu810:~$ ls
2009-01-10_16.14.12.692639.ubuntu810.ful
2009-01-10_16.20.28.729905.ubuntu810.inc
2009-01-10_19.33.15.009990.ubuntu810.ful
```

Un dossier est créé pour chaque sauvegarde, nommé selon la date et l'heure, le nom d'hôte et le type de sauvegarde (ful ou inc). Le dossier contient une archive compressée (.tgz) et quelques fichiers comprenant des informations supplémentaires relatives à la restauration :

```
eric@ubuntu810:~/2009-01-10_19.33.15.009990.ubuntu810.ful$ ls -l
total 235016
-rw-r--r-- 1 eric eric       206 2009-01-10 19:33 excludes
-rw-r--r-- 1 eric eric 240171518 2009-01-10 19:33 files.tgz
-rw-r--r-- 1 eric eric    106058 2009-01-10 19:33 flist
-rw-r--r-- 1 eric eric     79342 2009-01-10 19:33 fprops
-rw-r--r-- 1 eric eric     32047 2009-01-10 19:33 packages
-rw-r--r-- 1 eric eric       313 2009-01-10 19:33 partitions
-rw-r--r-- 1 eric eric         4 2009-01-10 19:33 ver
```

Revenons-en à la configuration de sbackup. L'onglet **Heure** vous permet de définir les moments auxquels doit s'effectuer la sauvegarde automatique.



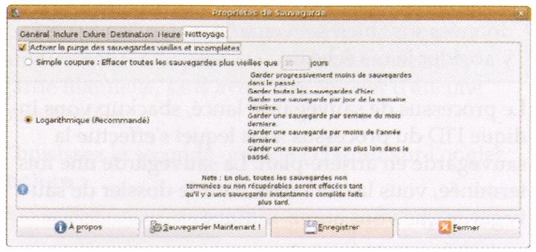
Le réglage par défaut correspond à une sauvegarde incrémentale chaque jour et une sauvegarde complète au minimum tous les sept jours. Ce réglage par défaut conviendra à la majorité d'entre-nous, mais il ne faut pas hésiter à la modifier, comme on a modifié le contenu de ce qui est sauvegardé.

Pour cela, cliquez sur **précisément à** et choisissez l'heure exacte de la sauvegarde. Ou bien, dans **Faire des sauvegardes**, choisissez un autre intervalle de sauvegarde.

Faites un test. Si vous êtes satisfait, n'oubliez pas d'enregistrer ;-).

Votre système de sauvegarde est immédiatement productif.

Le dernier onglet **Nettoyage** permet d'éviter que vos sauvegardes ne finissent par occuper tout l'espace disque.



Vous pouvez choisir de supprimer une sauvegarde tout simplement après un certain laps de temps, ou bien en fonction d'un algorithme sophistiqué, par le biais duquel les anciennes sauvegardes ne sont supprimées que partiellement. Le réglage par défaut convient généralement.

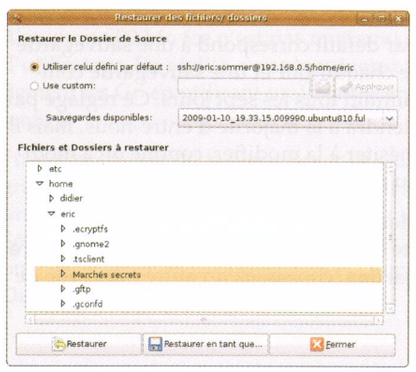
> J'ai lu ici et là que des difficultés pouvaient survenir au cours de ce processus de nettoyage – vérifiez donc que *sbackup* efface réellement les données et au besoin intervenez manuellement.

Restaurer

Les sauvegardes, une formidable invention qui permet aux utilisateurs de dormir sur leurs deux oreilles. Il ne s'agit toutefois que d'un moyen permettant d'atteindre un objectif : restaurer des données en cas de nécessité.

Ne prenez aucun risque et testez régulièrement vos sauvegardes, afin qu'en cas de problème vous n'ayez pas la désagréable surprise de découvrir que vos sauvegardes sont aussi utiles que des pagaies dans un désert.

Pour la restauration, lancez **Simple Backup Restore** dans le menu **Système / Administration**.

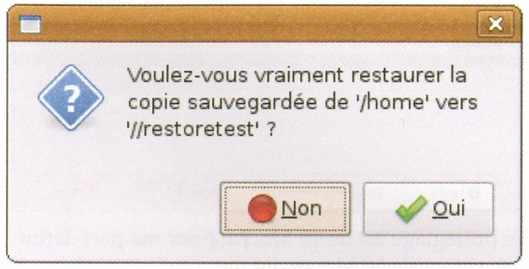


Sélectionnez la sauvegarde dans **Restaurer le Dossier de Source**. Dans la liste déroulante **Sauvegardes disponibles**, sélectionnez la sauvegarde souhaitée puis, dans **Fichiers et Dossiers à restaurer**, indiquez le dossier ou le fichier dont vous souhaitez retrouver le contenu.

Deux options s'offrent à vous : vous pouvez cliquer sur **Restaurer** afin de le restaurer à son emplacement d'origine. L'autre option **Restaurer en tant que** vous permet de restaurer l'objet à l'emplacement de votre choix.

> N'hésitez pas à créer un dossier de test pour essayer la restauration, histoire d'éviter de supprimer accidentellement des fichiers importants. Ce serait le comble.

Il ne vous reste plus qu'à confirmer que vous souhaitez réellement que la restauration se fasse à cet endroit – les fichiers existants seront supprimés sans aucune pitié !



En cliquant sur **Oui**, vous retrouvez ensuite les données restaurées à l'endroit choisi :

Comme vous pouvez le constater, les droits sont conservés au cours d'un processus de restauration :

```
eric@ubuntu810:~$ cd /restoretest/
eric@ubuntu810:/restoretest$ ls
didier  eric  roger
eric@ubuntu810:/restoretest$ ls -la
total 28
drwxr-xr-x  7 root     root     4096 2009-01-10 20:41 .
drwxr-xr-x 21 root     root     4096 2009-01-10 20:37 ..
drwxr-xr-x  3 didier   didier   4096 2009-01-10 20:41 didier
drwxr-xr-x 43 eric     eric     4096 2009-01-10 20:41 eric
drwxr-xr-x  3 roger    roger    4096 2009-01-10 20:41 roger
eric@ubuntu810:/restoretest$
```


Utiliser des CD et des DVD

L'utilisation de CD ou de DVD vierges avec sbackup peut engendrer des problèmes dans certains cas.

Si vous souhaitez graver vos données sur CD ou DVD et ainsi les archiver, vous pouvez utiliser la stratégie très simple que voici :

1. Créez une sauvegarde automatisée vers un dossier `/backup` ou `/agraver`.
2. Dans votre calendrier, créez une alarme qui vous rappelle de graver le dossier (et d'en supprimer le contenu après coup). Sans cela, il y a de fortes chances pour que vous oubliiez la sauvegarde !
3. Ouvrez un terminal et tapez `gksudo brasero` afin de lancer le graveur par défaut d'Ubuntu dans le mode d'administration du système. Sans les droits Administrateur, *vous ne disposerez pas des autorisations vous permettant de lire les fichiers de sauvegarde et vous ne pouvez pas lancer le processus de gravage.*
4. Gravez le contenu du dossier, n'oubliez pas ensuite d'étiqueter et d'archiver proprement les CD ou DVD ;-).

Le shell –Linux à l'état pur !

Ubuntu fait son maximum pour dissimuler ses origines, et il y parvient parfaitement. C'est l'une des principales raisons de son succès auprès des débutants...

Mais derrière la façade étincelante des interfaces graphiques (Gnome, KDE, ...) les outils traditionnels sont toujours là : la console, aussi nommé le *shell*. Au cours de notre périple, vous avez déjà été confronté à plusieurs reprises à la ligne de commandes. Dans ce chapitre, je vous propose un aperçu du monde d'UNIX tel qu'il se présentait à l'origine – avant l'existence des interfaces graphiques, tel que de nombreux utilisateurs expérimentés continuent à l'utiliser, par souci de facilité et de rapidité.

Lorsqu'une personne vous demande si vous connaissez Linux, elle souhaite généralement savoir si vous savez utiliser la console ! Aussi incroyable que cela puisse paraître quand on débute, de nombreuses tâches peuvent être effectuées bien plus rapidement et plus efficacement via la console !

Vous découvrirez les formidables astuces proposées par le shell pour vous permettre de travailler de manière extrêmement efficace. Je vous proposerai également un petit cours accéléré portant sur le thème de la programmation de scripts shell. Vous apprendrez à écrire vos propres scripts shell et donc à tirer le meilleur du shell.

À l'issue de ce chapitre, vous pourrez vraiment vous considérer comme un **utilisateur expérimenté**. Un effet secondaire positif est que vous apprenez en même temps à vous débrouiller avec d'autres distributions Linux, car elles ont toutes un point commun : le shell !

Trucs et astuces

Le **shell** est l'interface par l'intermédiaire de laquelle l'utilisateur peut communiquer avec le système. Sur les systèmes Linux, **bash** (Bourne Again SHell) correspond au shell par défaut. Le programme concerné est `/bin/bash`. Sous Linux, vous êtes toutefois libre de choisir parmi un large éventail de shells, tels que par ex. C-Shell, Korn-Shell, etc.

Dans les pages suivantes, je vous propose un aperçu du fonctionnement de bash. Des bases qui nous permettront à nous aussi de pratiquer quelque peu la « magie des consoles ».

L'environnement utilisateur

Le shell lancé lorsque vous vous connectez est défini dans le fichier `/etc/passwd`. Vous trouvez ici un enregistrement pour chaque utilisateur du système.

```
pulse:x:106:115:PulseAudio daemon,,,:/var/run/pulse:/bin/false
saned:x:107:118::/home/saned:/bin/false
messagebus:x:108:119::/var/run/dbus:/bin/false
polkituser:x:109:120:PolicyKit,,,:/var/run/PolicyKit:/bin/false
avahi:x:110:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin
haldaemon:x:111:122:Hardware abstraction layer,,,:/var/run/hald
eric:x:1000:1000:Eric Amberg,,,:/home/eric:/bin/bash
```

Les colonnes sont séparées par un double-point (:). Voici leur signification !

- **Nom d'utilisateur** – c'est l'identifiant que l'utilisateur doit utiliser pour se connecter au système.
- **Mot de passe** – le mot de passe était jadis enregistré dans `/etc/passwd` ; on y trouve actuellement un X, ce qui signifie que le mot de passe est stocké dans `/etc/shadow`.
- **UID** – l'ID de l'utilisateur, root dispose toujours de la valeur 0, les utilisateurs de services obtiennent des ID compris entre 1 et 999, les utilisateurs normaux commencent à partir de 1000.
- **GID** – l'ID du groupe, définit le groupe principal de l'utilisateur, tel qu'il se présente dans `/etc/group`.
- **Commentaire** – champ de commentaires facultatif.
- **Répertoire personnel** – sur l'interface utilisateur graphique du répertoire personnel.
- **Shell** – le shell démarré lorsque l'utilisateur se rend sur la console ou lorsqu'il s'y connecte.

Vous pouvez d'ailleurs éditer et adapter ce fichier de manière tout à fait classique par le biais d'un éditeur de texte, par exemple lorsque vous voulez passer à un autre shell. Mais avant cela, vous devez d'abord l'installer séparément.

Logiquement, vous pouvez ensuite aussi adapter le fichier `/etc/group`, par ex. pour définir d'autres utilisateurs Admin autorisés à travailler avec sudo. Pour ce faire, il vous suffit d'ajouter le nom de l'utilisateur de manière appropriée !


```
polkituser:x:120:
avahi:x:121:
haldaemon:x:122:
admin:x:123:eric,user1,ststift
eric:x:1000:
sambashare:x:124:eric,ststift
user1:x:1001:
didier:x:1002:
```

Vous constatez d'ailleurs aussi ici que, par défaut, chaque utilisateur se voit assigner un groupe principal de même nom – l'utilisateur **eric** dispose du groupe principal **eric** (cf. `/etc/passwd`).

Si vous pensez qu'il vous suffit de jeter un oeil dans `/etc/shadow` pour connaître le mot de passe de l'utilisateur, vous vous leurrez. Faites un essai : il est enregistré sous la forme d'une valeur de hachage (vous en rappelez-vous ?) qui le rend illisible. Lors de l'ouverture d'une session, le mot de passe saisi est *haché* de la même manière et la valeur de hachage trouvée est tout simplement comparée à celle du fichier.

Pour créer un utilisateur, utilisez la commande `useradd`. La commande suivante crée un utilisateur nommé testuser avec les valeurs par défaut !

```
$ sudo useradd -m testuser
```

L'option `-m` se charge de fournir à l'utilisateur un répertoire personnel sous `/home/<username>`. Ce dernier est d'ailleurs créé à partir de `/etc/skel` qui sert de dossier modèle ; jugez-en par vous-même !

```
root@ubuntu810:~# ls -la /etc/skel
total 28
drwxr-xr-x   2 root root  4096 2008-10-30 21:57 .
drwxr-xr-x 135 root root 12288 2009-02-12 20:43 ..
-rw-r--r--   1 root root   220 2008-05-12 20:48 .bash_logout
-rw-r--r--   1 root root  3115 2008-05-12 20:48 .bashrc
lrwxrwxrwx   1 root root    26 2009-02-04 12:07 Examples -> /usr/share/example-content
-rw-r--r--   1 root root   675 2008-05-12 20:48 .profile
root@ubuntu810:~# ls -la /home/testuser
total 20
drwxr-xr-x 2 testuser testuser 4096 2009-02-12 20:43 .
drwxr-xr-x 6 root     root     4096 2009-02-12 20:43 ..
-rw-r--r-- 1 testuser testuser  220 2009-02-12 20:43 .bash_logout
-rw-r--r-- 1 testuser testuser 3115 2009-02-12 20:43 .bashrc
lrwxrwxrwx 1 testuser testuser   26 2009-02-12 20:43 Examples -> /usr/share/example-content
-rw-r--r-- 1 testuser testuser  675 2009-02-12 20:43 .profile
```

Ce que vous ajoutez dans `/etc/skel` apparaîtra dans tous les répertoires personnels de tous les utilisateurs qui seront créés à l'avenir (!). Vous pouvez donc adapter à votre guise le modèle d'environnement utilisateur, par exemple en créant un nouveau sous-dossier, un fichier de bienvenue, etc.

Avez-vous remarqué le point figurant devant certains fichiers ? Il s'agit de fichiers cachés. Lorsque `ls` est invoqué, ils sont affichés uniquement si vous avez spécifié l'option `-a`.

Nous passons déjà au thème suivant ! Comme vous le voyez, chaque utilisateur possède ces fichiers. Si vous le souhaitez, vous pouvez immédiatement les ouvrir pour y jeter un oeil. Utilisez `nano`, par exemple

```
nano
/home/testuser/<nom_du_fichier>
```

ou également vim. Remplacez simplement `nano` par `vim`.

.profile

Ce fichier est lu lorsqu'un utilisateur ouvre une session. Ce fichier est le siège des événements suivants !

- `.bashrc` est invoqué, à condition qu'il existe (ce qui est le cas par défaut).
- S'il existe, le sous-dossier `bin` est intégré dans la variable du chemin, ce qui permet d'invoquer sans indication de chemin les scripts personnels qui y sont enregistrés.

.bashrc

Normalement, ce fichier est invoqué uniquement s'il s'agit d'un shell interactif sans connexion, c'est-à-dire un shell que vous invoquez vous-même. Quant à .profile, il est lu dans tous les cas de figure. Y compris lors du processus d'ouverture de session.

Certains réglages propres à l'environnement sont effectués dans ce fichier, entre autres !

- La configuration de l'historique des commandes
- La configuration de l'invite
- L'établissement de variables d'environnement
- La définition des couleurs dans les listes de dossiers affichées à l'aide de `ls`
- La définition d'alias

Le dernier point est particulièrement intéressant ! Jetez un oeil dans le fichier – vous trouverez tout à la fin quelques lignes commençant par alias. Les alias permettent de raccourcir encore les commandes à taper. Il est par exemple possible de saisir simplement `ll` à la place de `ls -l`, ce qui facilite encore votre travail.

```
# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    eval "`dircolors -b`"
    alias ls='ls --color=auto'
    #alias dir='dir --color=auto'
    #alias vdir='vdir --color=auto'

    #alias grep='grep --color=auto'
    #alias fgrep='fgrep --color=auto'
    #alias egrep='egrep --color=auto'
fi

# some more ls aliases
#alias ll='ls -l'
#alias la='ls -A'
#alias l='ls -CF'
```

Personnellement, je préfère que `l` invoque la commande `ls -la`. Procédons à la modification.

N'oubliez pas de supprimer le signe de commentaire (#) pour activer la ligne !

```
# some more ls aliases
#alias ll='ls -l'
#alias la='ls -A'
alias l='ls -la'
```

Enregistrez le fichier, fermez la session et rouvrez-la. Vous pouvez dès maintenant invoquer la commande `ls -la` à l'aide de `l`.

```
root@ubuntu810:/# l
total 84
drwxr-xr-x  20 root root  4096 2009-02-04 12:16 .
drwxr-xr-x  20 root root  4096 2009-02-04 12:16 ..
drwxr-xr-x   2 root root  4096 2009-02-04 12:17 bin
drwxr-xr-x   3 root root  4096 2009-02-04 17:52 boot
lrwxrwxrwx   1 root root    11 2009-02-04 12:06 cdrom -> media/cdrom
drwxr-xr-x  12 root root 13420 2009-02-12 20:31 dev
drwxr-xr-x 135 root root 12288 2009-02-12 20:43 etc
drwxr-xr-x   6 root root  4096 2009-02-12 20:43 home
```

.bash_logout

Lorsque vous quittez le shell, ce fichier est invoqué et efface le contenu de la console afin de vider les saisies visibles et la mémoire tampon de la console.

Les fichiers de configuration à l'échelle du système

`/etc/` contient les fichiers `profile` et `bash.bashrc`. Leur fonction est la même que celle des fichiers correspondants du répertoire personnel, mais ils agissent à l'échelle du système, ils sont donc activés par chaque utilisateur. En cas d'ambiguïté, les réglages personnels remplacent ceux effectués à l'échelle du système.

Travailler avec le shell

Bash est très *puissant* – un terme bien singulier pour désigner un petit morceau de logiciel, ne trouvez-vous pas ! L'adjectif *puissant* est utilisé pour exprimer la richesse des fonctions. Autrement dit, un shell permet de faire une foule de choses. Voici quelques trucs et astuces !

Compléter des lignes

Sous Linux, vous n'avez pas besoin de tout écrire entièrement dans la console. Si une commande ou un chemin est explicite, vous pouvez utiliser la fonction de complétion automatique à l'aide de la touche de **Tabulation**. Si vous saisissez par ex. `apt-g`**+Tabulation**, le système complète automatiquement l'expression et affiche `apt-get`.

Tabulation tapé deux fois affiche les autres possibilités lorsque la saisie n'est pas encore explicite. Voici par ex. ce qui est affiché pour les commandes commençant par `apt` !

```
eric@ubuntu810:~$ apt
apt-cache                  apt-ftparchive                 aptitude-run-state-bundle     apturl
apt-cdrom                  apt-get                        apt-key
apt-config                 aptitude                       apt-mark
apt-extracttemplates       aptitude-create-state-bundle   apt-sortpkgs
eric@ubuntu810:~$ apt
```

Je trouve également cette fonction très intéressante avec les chemins d'une certaine longueur. Utilisez la **Tabulation** à chaque endroit à partir duquel vous pouvez compléter le sous-dossier suivant. À votre avis, en combien de temps cette méthode nous permettra-t-elle de passer dans le dossier suivant avec `cd` ?

`/lib/modules/2.6.27-7-generic/kernel/drivers/bluetooth`

7 secondes. Il suffit de taper les deux premières lettres du sous-dossier, de cliquer sur **Tabulation**, et ainsi de suite. Habituez-vous à cette technique et vous deviendrez si rapide avec la ligne de commandes que plus personne n'arrivera à vous suivre. Vous deviendrez ce fameux *gourou* Linux dont votre cercle de connaissances n'aura de cesse de vanter les mérites ;-).

Transmettre des résultats

Vous pouvez rediriger l'affichage d'une commande vers une autre commande afin de la lui transmettre, ce que vous avez déjà appris dans le cadre de l'affichage des longues listes. Vous pouvez ainsi afficher facilement page par page les longues listes de dossiers par exemple à l'aide de

`ls -la | less.`

La redirection dans un fichier est une autre possibilité. Vous pouvez aussi rediriger le résultat de `ls -la` dans le fichier `listing.txt` par le biais de la commande suivante :

```
$ ls -la >listing.txt
```

Si `listing.txt` n'existe pas encore dans le dossier actuel, le fichier est créé à cet endroit. Si le fichier dispose déjà d'un contenu, il est *remplacé si* le signe « Plus grand que » (>) est utilisé. Si vous voulez *ajouter* à la suite d'un fichier existant, utilisez le double signe « Plus grand que » (>>).

Si vous voulez d'une part faire apparaître le résultat à l'écran et d'autre part le rediriger vers un fichier, vous pouvez utiliser `tee` qui fera office de… « T » !

```
$ ls -la | tee listing.txt
```

Si vous insérez `tee` dans l'option `-a`, l'affichage est ajouté au fichier au lieu d'écraser le contenu.

Vous pouvez aussi associer différentes commandes. Si vous voulez par exemple en une seule ligne basculer dans votre répertoire personnel puis l'afficher, vous pouvez alors utiliser un *branchement inconditionnel* à l'aide du point-virgule :

```
$ cd /home/eric ; ls
```

La commande `ls` est exécutée dans tous les cas, même si `cd /home/eric` a échoué. `ls` liste le contenu du dossier actif, quel qu'il soit, /home/eric si tout a fonctionné comme prévu.

« && » permet d'exécuter la deuxième commande uniquement si la première a réussi :


```
$ cd /home/asdedgasdz && ls
```

Renvoie un message d'erreur de la première commande, car l'utilisateur `asd...` n'existe pas. `ls` n'est pas exécuté.

Mais cela peut bien sûr aussi se faire dans l'autre sens. Vous voulez tester si un dossier existe déjà, et dans le cas contraire le créer ? Utilisez alors le double trait vertical (Pipe) `||`, comme montré dans l'exemple suivant :

```
$ cd /home/eric/bin || mkdir
/home/eric/bin
```

Si le dossier indiqué existe, nous nous y rendons. Dans le cas contraire, il est créé. Mais cette procédure génère encore un message d'erreur, ce qui n'est pas forcément souhaitable dans un script.

Nous allons donc désactiver les messages d'erreur. Nous utiliserons pour cela l'aventureuse syntaxe `2>/dev/null`. Elle se charge de guider le canal des erreurs (indiqué avec un 2) vers la poubelle virtuelle `/dev/null`. Dans ce cas, la ligne de commandes ressemblerait à ceci !

```
$ cd /home/eric/bin 2>/dev/null ||
mkdir /home/eric/bin
```

Du vrai travail de professionnel ;-).

Invoquer des commandes à plusieurs reprises

Un fichier `.bash_history` existe dans le répertoire personnel de chaque utilisateur, il stocke les commandes déjà saisies. Les touches de clavier **Haut** et **Bas** permettent de parcourir la liste de ses commandes. Pour afficher une nouvelle fois la commande que vous venez de saisir, il vous suffit donc d'appuyer sur **Haut**.

CTRL+R vous permet d'activer la recherche de commandes.

```
(reverse-i-search)`apt': sudo apt-get install ecryptfs-utils
```

Saisissez simplement la chaîne de caractères recherchée jusqu'à l'affichage de la commande souhaitée. Les touches **Haut** et **Bas** vous permettent de parcourir l'historique des commandes vers le haut ou le bas en partant de l'élément trouvé.

Variables shell et affectation de variables

Une variable est un « espace » mémoire qui contient une valeur (c'est très résumé).

`$ variable=valeur` permet d'en définir la valeur.

`$echo $Variable` permet d'afficher le contenu de la variable.

Notez que pour définir son contenu, vous tapez le nom de la variable *sans* le signe $ (ne confondez pas avec l'invite de commande), mais pour l'invoquer vous devez lui ajouter le $. Aucun espace ne doit suivre ou précéder le signe égal.

S'il ne s'agit pas d'un nombre, la valeur doit être placée entre guillemets simples ou doubles.

```
$ variable="toto"
```

`PATH` est une variable essentielle. Elle contient tous les chemins examinés lors de la saisie d'une commande, de telle sorte que vous n'avez pas besoin de saisir l'intégralité du chemin de la commande :

```
eric@ubuntu810:~$ echo $PATH
/home/eric/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:
/sbin:/bin:/usr/games
```

Comme vous pouvez le voir, les dossiers sont séparés par un double-point (`:`). Voici comment ajouter un autre dossier au PATH.

Admettons que vous disposiez encore d'un dossier de commandes nommé `/usr/local/sbin`. Vous pouvez alors le compléter par la commande suivante !

```
$ PATH=$PATH':/usr/local/sbin'
```

Nous affectons ici à la variable `PATH` son ancienne valeur (`$PATH`) et la complétons par la chaîne de caractères qui apparaît ensuite entre apostrophes.

> Les guillemets (") servent à conserver les espaces dans une chaîne de caractères, afin que le Bash ne pense pas qu'une nouvelle commande, option ou autre commence ici. Mais les guillemets autorisent l'exploitation d'autres caractères spéciaux (par ex. \ ou *) par bash. Pour vous assurer que la chaîne de caractères est reprise conformément à sa version originale, utilisez des apostrophes simples (').

Vous pouvez saisir ces modifications non seulement dans la ligne de commandes mais également dans `.profile`. De cette façon, le nouveau dossier sera ajouté à la variable PATH à chaque fois que vous ouvrirez votre session.

Les principales variables shell :

`PS1` – contient l'invite. Vous pouvez personnaliser intégralement votre invite, libre à vous de déterminer vos propres besoins. Considérez la page de manuel (page man) de bash et recherchez l'expression PROMPTING.

`HOME` – contient le chemin vers votre répertoire personnel, par exemple `/home/eric`.

`LOGNAME` – contient le nom sous lequel se connecte l'utilisateur, par exemple eric.

HOSTNAME – contient le nom d'hôte de l'ordinateur (qui permet par exemple de l'identifier sur un réseau), par exemple `ubuntu810`.

Commandes utiles

Vous découvrirez dans ce chapitre des commandes très utiles et pratiques dans le cadre d'une utilisation quotidienne de Linux. Je vous présenterai également l'une ou l'autre astuce qui vous simplifieront la vie.

> Toutes les commandes Unix disposent d'un mode d'emploi complet accessible depuis la console : tapez man suivi du nom de la commande et validez. Par exemple :
>
> `$ man find`
>
> Ces pages de man sont également accessibles sur Internet, il suffit de lancer une recherche sur Google.

C'est parti !

Find : Sherlock Homes

La commande `find` vous permet de retrouver n'importe quel fichier ou dossier du système de fichiers. La commande est un peu compliquée, il vous faudra un certain temps pour vous y habituer. En contrepartie, elle est très puissante et vous rendra de grands services. Voici les bases de son fonctionnement.

```
find <dossier de départ> <options
de recherche> -[Option]
```

Avant d'en venir aux exemples, commençons par un aperçu des *options de recherche*.

`-atime [+-]<jours>`	dernier accès exprimé en jours : plus ancien du nombre de jours indiqué (+), ou moins ancien (-)
`-mtime [+-]<jours>`	dernière modification au niveau du contenu exprimée en jours (le reste fonctionne de la même manière que `atime`)
`-user <nom>`	appartient à l'utilisateur
`-group <nom>`	appartient au groupe
`-type <type de fichier>`	`f` = fichier `d` = dossier `l` = lien
`-name <fichier/dossier>`	recherche du nom de fichier ou de dossier indiqué
`-size [+-]<taille>k`	taille supérieure (+) ou inférieure (-) au nombre de kilooctets indiqué
`-maxdepth <niveau>`	profondeur de dossier maximale à partir du dossier de départ jusqu'à celui recherché, le niveau est exprimé en chiffres

Voici maintenant les exemples promis :

```
find / -name gulugulu
```

recherche à partir du dossier (/) tous les dossiers ou fichiers nommés précisément `gulugulu`.

```
find /home -group otto -type f
```

recherche à partir de `/home` tous les fichiers auxquels le groupe *otto* est assigné.

```
find /home -user otto -mtime -3
```

recherche sous `/home` tous les fichiers de l'utilisateur *otto* dont le contenu a été modifié au cours des 3 derniers jours. Vous pouvez ainsi procéder à des sauvegardes incrémentales et différentielles.

```
find / -user otto -size +3000k
```

recherche tous les fichiers de l'utilisateur *otto* dont la taille est supérieure à 300 Ko. Cela vous permet de mettre la main sur les éléments occupant un large espace disque.

Restons-en là concernant ces exemples. Bien, mais nous ne sommes toujours pas arrivés au bout de cette commande particulièrement efficace.

Les *options* derrière les *options de recherche* déterminent le sort réservé au résultat de la recherche. Tant que rien d'autre n'est stipulé, `find` affiche sur l'écran le résultat de sa recherche. Cette option correspond à `-print` qui est définie par défaut. Il n'existe à vrai dire qu'une seule option importante dans votre cas de figure :

```
-exec <commande> {} \;
```

Quelques explications seraient les bienvenues : comme vous pouvez l'imaginer, une autre commande est mise ici à exécution. Elle reprend ligne par ligne le résultat de la commande `find`. Cela est exprimé par les accolades `'{}'`. `'\;'` il s'agit d'un branchement inconditionnel de la commande indiquée. Les commandes d'une même ligne doivent être séparées par un point-virgule qui doit être en outre *échappé* (`\`), afin qu'il ne soit pas interprété comme faisant partie intégrante de la commande.

Pfff, un véritable casse-tête. Voici un exemple :


```
find / -user julie -type f -exec
rm -f {} \;
```

recherche tous les fichiers de l'utilisatrice *julie* et les efface (`rm`) sans demande de confirmation (option `-f` de `rm`). Cela peut être utile, lorsqu'un utilisateur (par ex. un collaborateur dans une entreprise) ne travaille plus sur un système et que ses fichiers deviennent inutiles. **Mais attention à ne pas détruire des fichiers dont on a encore besoin**. RM est une commande très puissante qui peut même endommager le système d'exploitation lorsqu'elle est utilisée n'importe comment !

Il est impératif que vous procédiez à une sauvegarde des fichiers.

Find peut être relativement lente car elle va vérifier chaque fichier sur le disque dur.

Locate : Dr. Watson

Cette commande comptera certainement parmi vos commandes préférées. `locate gulugulu` vous permet de rechercher un fichier ou un dossier nommé *gulugulu* ou contenant cette expression dans son nom. La recherche est beaucoup plus rapide qu'avec `find` car elle est effectuée dans une base de données interne précédemment créée. En saisissant `updatedb`, vous actualisez la base de données. Vous devez réaliser cela après chaque installation et chaque grosse modification au niveau de votre système.

grep

La commande `grep` vous permet de rechercher des chaînes de caractères dans des fichiers texte.

```
grep <options>
<chaîne_de_caractères> <fichier>
```

Si `grep` trouve l'élément recherché, la ligne correspondante est alors affichée. Il est intéressant de noter que la chaîne de caractères peut aussi être indiquée sous la forme d'une *expression régulière* (Regular Expression). Ce qui rend la commande particulièrement puissante.

> Mais nous nous contenterons ici d'aborder les bases des expressions régulières. *Regular Expressions* est un thème très vaste auquel l'on pourrait consacrer des livres entiers.

Considérez tout d'abord les principales options :

`-i`	ignore les majuscules/minuscules
`-l`	indique uniquement les noms de fichiers (lorsque la recherche est effectuée dans plusieurs fichiers)
`-m`	indique le numéro de ligne
`-w`	l'expression doit être un mot (c'est-à-dire ne pas faire uniquement partie intégrante d'une chaîne de caractères plus longue)
`-x`	l'expression doit correspondre à la ligne entière
`-e`	permet l'utilisation d'expressions régulières évoluées

L'option `-i` vous sera particulièrement utile. Si vous travaillez avec des expressions régulières, il vous faudra éventuellement associer aussi l'option `-e` activant les expressions complexes. Sans cela, le motif de recherche de l'expression régulière risquera de ne pas être interprété ou mal interprété. Au lieu de `-e`, vous pouvez aussi utiliser `egrep` en tant que variante de `grep`.

> Un dernier mot concernant l'option `-l`
>
> Plutôt que d'effectuer la recherche dans un seul fichier, vous pouvez en indiquer plusieurs. Ainsi, `*.txt` par ex. recherchera dans le dossier actuel tous les fichiers portant l'extension `txt`. Ce n'est qu'à partir de là que l'option `-l` prend réellement tout son sens !

Quelques exemples pour commencer :

```
grep gulugulu fi-
chier_de_recherche.txt
```

grep recherche ici la chaîne de caractères *gulugulu* dans le fichier `fichier_de_recherche.txt`. Il s'agit de l'utilisation la plus simple. Il se peut aussi que la chaîne de caractères recherchée ne constitue qu'une partie d'une chaîne de caractères plus longue (à moins que vous utilisiez l'option `-w`).

> **Attention** ! si la chaîne de recherche contient des caractères spéciaux (ou des espaces), elle doit être entourée d'apostrophes simples.

Un autre exemple !

```
grep -i Gulugulu fi-
chier_de_recherche.txt
```

Cette instruction recherche aussi bien les lignes contenant *gulugulu* que *Gulugulu* ou *gUlUGulu* – les majuscules et minuscules sont simplement ignorées.

```
grep -i ^gulugulu fi-
chier_de_recherche.txt
```

recherche toutes les lignes commençant par *gulugulu*. Nous utilisons ici une expression RegEx (^).

```
grep gulugulu$ fi-
chier_de_recherche.txt
```

recherche toutes les lignes se terminant par *gulugulu*, c'est le rôle de l'expression RegEx `$`.

```
grep e.*ch fi-
chier_de_recherche.txt
```

recherche toutes les chaînes de caractères contenant un 'e', quel que soit l'endroit, suivi d'un nombre quelconque de caractères puis de 'ch'. Cette instruction permettrait par ex. de retrouver les mots *ecchymose*, *mercurochrome* ou *perruche*. C'est l'expression RegEx `.*` qui est utilisée ici ; elle signifie *aucun, un ou un nombre quelconque de caractères*.

Comme vous le savez déjà, la commande `grep` est souvent utilisée en association avec d'autres commandes. Vous pouvez ainsi rechercher une certaine chaîne de caractères dans le résultat d'une commande.

```
ps ax | grep -i nautilus
```

vous présente par exemple tous les processus nautilus.

`grep` simplifie surtout la recherche dans les fichiers journal, et ce de façon remarquable. Ces derniers sont souvent très gros et donc confus. Si vous recherchez par ex. une adresse IP ou un service, vous pouvez les indiquer via une chaîne de caractères de recherche.

cat : *chat* alors...

La commande `cat` sert à l'affichage et à la fusion de fichiers textes. Précisons que le nom anglais de la commande, cat (*chat* en français), n'a rien à voir avec nos amis à quatre pattes : il est tiré du mot *concatenate* (enchaîner). `cat` vous permet d'afficher à l'écran l'intégralité d'un fichier (texte), du début à la fin. Pour l'affichage des fichiers plus longs, il est toutefois préférable d'utiliser la commande `less`.

Si vous souhaitez ajouter le contenu d'un fichier texte (fichier2.txt) à la fin d'un autre fichier déjà existant (fichier1.txt), écrivez ceci !

```
cat fichier2.txt >> complet.txt
```

> Attention ! si vous affichez des *fichiers binaires* (en très gros, autre chose que des fichiers de texte brut) avec `cat`, il peut arriver que votre jeu de caractères « glisse » et que vous voyiez affichés uniquement des hiéroglyphes. Pour vous sortir de ce mauvais pas, il ne vous reste plus qu'à taper un `reset` invisible.

sort : le respect de l'ordre

Cette commande trie le contenu d'un fichier texte. Elle connaît entre autres les options suivantes :

`-c`	vérifie si le fichier est déjà trié
`-f`	traite minuscules et majuscules de façon analogue
`-m`	réunit deux ou plusieurs fichiers pré-triés en un gros fichier trié (plus rapide que de commencer par les assembler puis les trier)
`-o <fichier>`	écrit le résultat dans le fichier indiqué, qui peut aussi être identique à l'ancien
`-r`	trie dans l'ordre inverse
`+n1 [-n2]`	considère pour le tri uniquement le texte à partir de la colonne *n1* (et éventuellement jusqu'à la colonne *n2* – non incluse)

Exemples !

```
sort -m trie1 trie2 -o trie
```

réunit les deux fichiers (pré-triés !) `trie1` et `trie2` et les écrit dans `trie`.

```
ls -l | sort -r
```

affiche le contenu du dossier dans l'ordre inverse.

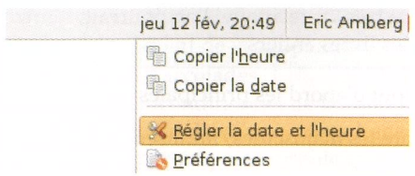
date – Quelle heure est-il !

Vous connaissez déjà cette commande. Elle affiche la date et l'heure :

```
eric@ubuntu810:~$ date
jeudi 12 février 2009, 20:48:01 (UTC+0100)
```

Mais vous pouvez également régler la date et l'heure par le biais de cette commande. Cette procédure étant beaucoup plus simple en passant par l'interface graphique, je recommande aux curieux la lecture de la page man de `date`.

Sous GNOME, il suffit de cliquer-droit sur l'affichage de la date, au niveau du tableau de bord :



Vous y trouverez l'option **Régler la date et l'heure** ...

whoami : qui suis-je ?

Pour les distraits ! la commande `whoami` affiche l'utilisateur avec lequel vous êtes en train de travailler. Cela peut paraître idiot, mais ça peut être très utile lorsque vous manipulez `su` ou `sudo` et travaillez avec différents utilisateurs. C'est parfois suffisant pour s'emmêler les pinceaux...

id : à quoi suis-je affilié ?

La commande `id` vous présente toutes les affiliations de votre utilisateur actuel !

```
eric@ubuntu810:~$ id
uid=1000(eric) gid=1000(eric) groupes=4(adm),20(dialout),24(cdrom),46(plugdev))
,108(lpadmin),123(admin),124(sambashare),1000(eric)
```

Sont affichés en particulier les ID de l'utilisateur et des groupes auxquels il est affilié.

Tar : emballer, déballer, ranger

Tar correspond aux initiales de *tape archiver*. Ce programme est surtout utilisé dans l'environnement Linux lorsqu'il est question de créer des copies de secours (par ex. sur des bandes ou sur CD/DVD) ou tout simplement de compresser des fichiers afin de gagner de l'espace disque :

```
tar [options] <nom_archive>.tar
<dossier/fichier>
```

Il y a de nombreuses options, certaines vont définir ce que fera tar...

`-c`	(create) crée une nouvelle archive
`-t`	(table) présente la table des matières de l'archive
`-r`	(recreate) ajoute un fichier dans une archive
`-d`	(diff) compare des fichiers et définit les éventuelles différences
`-x`	(extract) extrait les fichiers de l'archive indiquée et les copie dans le dossier actuel, sans effacer les fichiers de l'archive

... d'autres sont les *options* à proprement parler !

`-f <fichier archive.tar>`	utilise le fichier indiqué en tant qu'archive, ce que vous indiquez dans tous les cas de figure
`-v`	présente toutes les actions (au cours de l'archivage), –vv vous permet d'afficher encore plus d'éléments (very verbose – je ne plaisante pas !)
`-z`	compresse le dossier indiqué par l'intermédiaire du programme de compression standard de Linux `gzip`

Créer une archive Tar

Tar est parfaitement adapté à la création de sauvegardes.

> Pour pouvoir utiliser `tar` afin d'effectuer une sauvegarde des fichiers autres que vos propres fichiers, vous devez utiliser `sudo` ou travailler en tant que root.

La commande

```
tar -cvf didier-arch.tar
/home/didier
```

vous permet de sauvegarder et d'archiver l'intégralité du contenu du répertoire personnel de *didier*. Le fichier d'archive créé est localisé dans le dossier actuel (si vous n'avez pas indiqué de chemin) et se nomme `didier-arch.tar`.

> **Attention** ! si vous utilisez des chemins *absolus* pour les fichiers ou dossiers à sauvegarder (avec un `/` au début), `tar` supprime la barre oblique placée au début et vous en informe.
>
> Pourquoi `tar` fait-il cela ? C'est très simple : si vous décompressez une archive en tant qu'Admin (et *root* est l'Admin) et que cette archive contient le chemin absolu, par exemple `/etc/passwd`, vous pouvez écraser le fichier de mots de passe du système et le rendre éventuellement inutilisable.
>
> Il s'agit donc d'un mécanisme de sécurité qui permet en outre de rendre les choses plus flexibles. Il conserve en son sein des structures de dossiers complètes mais en permettant de les décompresser où l'on veut sur le disque dur, sauf là où ça risque de créer des problèmes graves.

Compresser une archive

Vous voulez gagner de la place en archivant des fichiers ? `tar`. Ne fait pas gagner de place, elle se contente de regrouper les fichiers dans une archive. Mais vous pouvez demander à Linux de compresser l'archive. L'option `-z` permet de compresser l'archive au cours de la création.

L'option `-z` utilise le programme de compression `gzip`. Nous en reparlerons plus longuement dans la prochaine partie. Dans ce cas de figure, la commande `tar` ressemble à ceci !

```
tar -cvzf eric-arch.tar.gz
/home/eric
```

Notez que nous avons nommé la nouvelle archive `*.tar.gz`. L'archive compressée peut être considérablement plus petite que sa version non compressée.

```
-rw-r--r--   1 root root   7884800 2009-01-13 12:16 eric-arch.tar
-rw-r--r--   1 root root     76090 2009-01-13 12:16 eric-arch.tar.gz
```

Le degré de compression dépend bien entendu du type des fichiers d'origine et de leur niveau de compression au départ.

Afficher une archive tar

Vous avez créé et compressé une archive `tar`. Voyons comment en afficher le contenu.

Si vous avez une archive non compressée, utilisez directement l'option `-tf <nom_archive>`. Le contenu vous est présenté :

```
root@ubuntu810:/# tar -tf user1-arch.tar
home/user1/
home/user1/.bash_logout
home/user1/.gnome2/
home/user1/.gnome2/keyrings/
```

Vous avez aussi la possibilité d'afficher le contenu d'un fichier compressé (`tar.gz`). Vous devez pour cela utiliser l'option `-z` afin d'indiquer à la commande qu'il s'agit d'un fichier compressé :

```
root@ubuntu810:/# tar -tzf user1-arch.tar.gz
home/user1/
home/user1/.bash_logout
home/user1/.gnome2/
home/user1/.gnome2/keyrings/
home/user1/.gnome2/keyrings/login.keyring
home/user1/.gnome2/panel2.d/
```

Ça c'est du service !

Décompresser une archive

L'option `tar` de décompression d'une archive est `-x`.

```
root@ubuntu810:/# tar -xzvf user1-arch.tar.gz
home/user1/
home/user1/.bash_logout
home/user1/.gnome2/
home/user1/.gnome2/keyrings/
home/user1/.gnome2/keyrings/login.keyring
home/user1/.gnome2/panel2.d/
home/user1/.gnome2/panel2.d/default/
```

Elle permet de décompresser dans le dossier actuel les fichiers archivés, c'est-à-dire la *structure de fichiers* archivée (tout ce qui se trouvait au-dessous de `home/user1` a été sauvegardé dans l'exemple mentionné ci-dessus).

> Le contenu sera décompressé là où est placé l'archive. Dans notre cas, si les données doivent être restaurées à l'emplacement d'origine, le dossier actuel doit être `/`, c'est-à-dire le dossier racine. Si vous copiez ou déplacez le fichier d'archive à un autre endroit, c'est là que la structure sera établie.

gzip et gunzip

`gzip` compresse ou décompresse le fichier indiqué. Voici la commande correspondante !

```
gzip [options] fichier
```

L'extension `.gz` est ici assignée automatiquement au fichier compressé. Options importantes :

`-c`	le fichier d'origine n'est pas modifié, dans le cas contraire il est ensuite effacé
`-d`	décompresse le fichier indiqué (correspond à *gunzip*)
`-r`	(dé)compresse aussi les fichiers des sous-dossiers

Si vous compressez un fichier sans options avec `gzip`, le fichier d'origine est effacé et il ne vous reste plus que le fichier `.gz`.

Mais vous pouvez éviter cela grâce à `-c`, comme expliqué dans l'exemple suivant.

```
eric@ubuntu810:~$ gzip -c test > test.gz
eric@ubuntu810:~$ ls
Images   Documents        listing1        Private
bin      encrypted.asc    listing.txt     test
binn     Eric_Amberg.asc  Musique         test.gz
binnn    eric-arch.tar    Public
binnnn   Exemples         Packages
Desktop  Marchés secrets  Packages.gz
```

Nous indiquons ici à `gzip` qu'il ne doit pas modifier `test` (qui doit être toutefois compressé) et qu'il doit écrire ou rediriger le résultat dans `test.gz`.

Pour décompresser les fichiers, utilisez l'option `-d` !

```
gzip -d <fichier>.gz
```

Vous pouvez également utiliser `gunzip` !

```
gunzip <fichier>.gz
```

soit dans notre exemple `gunzip test.gz`.

Ici aussi, le fichier d'origine est restauré et le fichier compressé est supprimé, autrement dit il n'existe plus au final.

bzip2 une alternative à gzip

`bzip2` est un programme de compression très efficace. Il utilise un algorithme plus moderne (nommé Burrows-Wheeler) et obtient des taux de compression potentiellement meilleurs. Un fichier compressé avec `bzip2` porte l'extension `.bz2`.

`bzip` connaît grosso modo les mêmes options de commande – vous utilisez donc le programme comme `gzip`.

```
root@ubuntu810:~# bzip2 eric-arch.tar
root@ubuntu810:~# ls -l eric-arch.tar.*
-rw-r--r-- 1 eric eric 52514 2009-01-13 12:56 eric-arch.tar.bz2
-rw-r--r-- 1 root root 76106 2009-01-13 12:57 eric-arch.tar.gz
```

L'exemple présente l'extension de fichier `.bz2` au lieu de `.gz`. On obtient ici aussi un taux de compression nettement supérieur à celui du fichier `.gz`. Mais comme avec `gzip`, tout dépend du type de fichier que vous compressez.

Programmation d'un script shell

Nous disposons à présent d'une remarquable boîte à outils. Nous sommes prêts à étudier la façon d'automatiser l'une ou l'autre procédure.

> Sur un système Linux, pratiquement tout est géré par des scripts Shell ! Prenons par exemple un script quelconque dans `/etc/init.d`. C'est ici que sont gérés les services. J'ai déjà traité la question plus tôt dans ce manuel.

Je vous propose dans ce chapitre une brève introduction à la programmation de scripts shell. N'espérez donc pas devenir un programmeur chevronné : des années d'entraînement et d'expérience seront nécessaires. Mais avec un peu de patience et de pratique, vous serez déjà en mesure de créer vos propres scripts et de comprendre dans les grandes lignes les scripts de gestion du système.

Qu'est-ce qu'au juste un script shell ?

Les *bash* ainsi que les autres shells sous Unix/Linux disposent de leur propre langage de programmation. De loin pas aussi performant que *C/C++*, *Java* ou d'autres langages analogues, mais étonnamment flexible. Suffisant en tout cas pour automatiser de nombreuses tâches de routine.

Un script shell se compose de requêtes de l'utilisateur, d'opérations conditionnelles (fait ça si X ou ceci si Y, sinon ne fait rien), de boucles et surtout de programmes « normaux », que vous pourriez aussi invoquer manuellement via la console.

Vous pouvez créer un script shell en créant un nouveau fichier TXT à l'aide de votre éditeur préféré, en écrivant ceci sur la première ligne !

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

Le fichier est alors interprété comme un fichier de script *bash*.

Généralement, pour que cela fonctionne, vous devez d'abord rendre le fichier exécutable, par exemple un accès complet avec la commande suivante :

```
chmod 777 <nom_du_fichier>
```

On donne normalement à un script shell une extension `.sh`.

> `chmod` (change modus) vous permet de définir les droits du propriétaire (Owner), du groupe (Group) et du reste du monde (Other). Les chiffres définissent les autorisations pour les différents groupements ! 4 pour lecture, 2 pour écriture et 1 pour exécution. Combinez les valeurs pour définir les droits. 4+2+1=7 -> tous les droits.

Dans ce cas, tous les utilisateurs ont le droit de tout faire (lire, écrire et exécuter), ce qui n'est pas vraiment prudent. Pour assurer un minimum de sécurité, n'accordez tous les droits qu'à vous et à vous seul, octroyez au groupe uniquement les droits de lire et d'exécuter, et rien du tout au reste du monde. Vous obtiendrez alors ceci :

```
chmod 750 <nom_du_fichier>
```

Un script shell est invoqué comme un programme binaire. Nous plaçons nos scripts sous `~/bin`, c'est-à-dire dans notre répertoire personnel, sous `bin`. Pour que vous puissiez l'invoquer sans devoir indiquer le chemin, nous avons complété en conséquence la variable PATH (voir plus haut). Vérifiez éventuellement que cela a aussi été fait pour *root* avec `PATH=$PATH':~/bin'`.

> Pour les exemples suivants, nous avons souvent besoin de l'utilisateur root ou des privilèges d'administration. Dans ce cas, vous devez adopter l'identité root. Une façon de le faire provisoirement est d'invoquer sudo : `sudo -s` ou `su -` (le signe moins appartient à la commande !). Nous avons déjà abordé le sujet plus tôt dans le manuel.

Un premier script

Vous pouvez exécuter le script suivant uniquement en qualité de *root*, car seul root a le pouvoir de créer des utilisateurs. `useradd` vous permet de créer des utilisateurs. La commande connaît plusieurs paramètres que nous pouvons définir, dont :

`-d <répertoire_personnel>`	crée le répertoire personnel comme indiqué (avec un chemin absolu !)
`-g <groupe_principal>`	définit un groupe principal

`-c "commentaire"`	description pour l'utilisateur, par ex. son nom complet

Nous allons maintenant récupérer ces informations à l'aide d'un script interactif et nous en servir pour créer le nouvel utilisateur. En qualité d'utilisateur *root*, commencez par créer un fichier `nouvel_utilisateur.sh` sous `/home/<votre_utilisateur>/bin`, soit en abrégé : `~/bin`.

Voici d'abord le script. Vous pouvez l'écrire tel quel dans un fichier, sans la numérotation des lignes qui sert uniquement à apporter plus de clarté aux explications :

```
1  #!/bin/bash
2  # Nom
3  echo -n "Nom d'utilisateur ! "
4  read nom
5  # Répertoire personnel
6  echo -n "Dossier racine ! "
7  read rep_perso
8  # Groupe
9  echo -n "Groupe ! "
10 read groupe
11 # Commentaire
12 echo -n "Commentaire ! "
13 read commentaire
14 useradd -m -d $rep_perso -g $groupe -c
"$commentaire" $nom
```

Le signe dièse (#) est un signe de commentaire : la ligne n'est exécutée (à l'exception du #! de la ligne 1).

> Prenez l'habitude de bien documenter vos scripts : leur lisibilité n'en sera que meilleure et vous apprécierez des commentaires utiles lorsque vous reprendrez un vieux scripts quelques mois ou années plus tard.

Vous connaissez déjà la commande `echo` de la ligne 3 ; `-n` signifie ! *AUCUNE nouvelle ligne après.*

La ligne 4 indique la façon dont vous pouvez lire la valeur d'une variable avec `read <variable>`. C'est à cet endroit qu'apparaît sur la console une invite de saisie à l'attention de l'utilisateur. La saisie est terminée avec la touche **Entrée**.

> Les variables doivent disposer en principe d'une désignation expressive.

Nous lisons les valeurs `nom`, `rep_perso`, `groupe` et `commentaire` ; ligne 14, la commande `useradd` est ensuite « alimentée » de ces valeurs.

Le commentaire doit être entouré de guillemets à cause des espaces. La commande est écrite sur une ligne. Faute de place, nous avons été contraints ici de l'afficher sur deux lignes.

> **Remarque** : si vous n'avez pas adapté la variable PATH de façon appropriée et si le script est localisé dans le répertoire actuel, vous devez alors l'invoquer à l'aide de `./nouvel_utilisateur.sh`. Vous devez donc aussi indiquer le chemin.

Le fichier est rendu exécutable grâce à `chmod`. Vous pouvez maintenant travailler avec. Pour que la commande `useradd` puisse être exécutée, vous devez démarrer le script en tant qu'utilisateur normal avec `sudo` !

```
eric@ubuntu810:~/bin$ sudo ./nouvel_utilisateur.sh
Nom d'utilisateur : laurent
Dossier racine : /home/laurent
Groupe : users
Commentaire : Laurent est notre petit nouveau !
eric@ubuntu810:~/bin$
```

La demande d'informations simple bien fonctionner. Le nouvel utilisateur a-t-il aussi été créé ! Observons `/etc/passwd`.

```
hohorak:x:1003:1003:Horst Horak,,,,:/home/hohorak:/bin/bash
ststift:x:1004:1004:Stefan Stift,,,,:/home/ststift:/bin/bash
testuser:x:1005:1005::/home/testuser:/bin/sh
laurent:x:1006:100:Laurent est notre petit nouveau !:/home/laurent:/bin/sh
/etc/passwd (END)
```

L'enregistrement se trouve en fait à la dernière ligne. Félicitations ! Vous venez d'être promu adepte des scripts shell !

> **Attention** ! le groupe indiqué doit déjà exister, faute de quoi un message d'erreur est généré ! Si vous voulez créer votre propre groupe, vous pouvez ajouter le groupe (au préalable) à l'aide de `groupadd <groupe>`.

Si cela ne marche pas du premier coup, vérifiez encore minutieusement les commandes de votre script. Et au besoin une seconde fois ! La moindre faute invalidera le script.

Si je me base sur ma propre expérience, il n'existe *aucune* erreur que l'on ne commettra *pas* au moins une fois lorsque l'on apprend à programmer. Et même lorsqu'on a appris ☺.

Instructions conditionnelles : si..., alors...

Quoi de plus ennuyeux qu'un programme suivant toujours le même déroulement linéaire ! Vous pouvez faire en sorte que votre programme réagisse à certaines situations. Vous demandez

si <condition remplie> alors <réaction>

Vous pouvez même intégrer d'autres instructions conditionnelles plus élaborées !

sinon, si <condition> alors <2ème réaction>

et

sinon <action>

Cela ressemblera à ceci :

```
if [<condition>]; then
   <instruction>
elif [<condition>]; then
   <instruction>
else
   <instruction>
fi
```

> `fi` ? Tout juste ! il s'agit d'un `if` inversé. Vous y serez encore une fois confronté avec `case`, que l'on interrompt à l'aide de `esac`. Les programmeurs Linux sont un peu particuliers ;-)

Le bloc `elif` est facultatif et est interrogé lorsque la condition `if` n'est pas vraie. Il en va de même pour `else` ; ce bloc est appliqué lorsque ni la condition pour `if` ni celle pour `elif` n'est vraie. Bien entendu, plus aucune instruction conditionnelle n'est alors nécessaire.

La formulation des conditions découle de prises de décision. Vous pouvez formuler différents types de conditions. La condition est toujours placée entre crochets. Si elle est vraie, le bloc d'instructions suivant est alors exécuté ; dans le cas contraire, le script saute à l'instruction conditionnelle suivante (`elif`, `else` ou `fi`).

Chaînes de caractères :

`[ "$<Var1>" = "$<Var2>" ]`	les valeurs des variables sont identiques
`[ "$<Var>" != "$<Var>" ]`	les valeurs des variables sont différentes
`[ "$<Var>" ]`	la variable n'est pas vide
`[ -z "$<Var>" ]`	la variable est vide

Fichiers :

`[ -d $<Var> ]`	la variable est-elle un dossier ?
`[ -f $<Var> ]`	la variable est-elle un fichier ?
`[ -e $<Var> ]`	existe-t-il un fichier ou un dossier avec la valeur de la variable ?

> **Attention** ! les espaces indiqués sont indispensables. Il s'agit d'une source d'erreur fréquente !

Nous pouvons maintenant enrichir un peu notre script `nouvel_utilisateur` afin d'intercepter les saisies vides.

Remplacez la zone `# Répertoire personnel` à partir de la ligne 8 par

```
8  echo -n "Dossier racine [/home/$nom] ! "
9  read rep_perso
10 if [ -z "$rep_perso" ]; then
11    rep_perso=/home/$nom
12 fi
```

Vous pouvez ainsi définir une valeur par défaut. Si aucune saisie n'a été effectuée par l'utilisateur à la ligne 9, la variable reste vide, la condition de la ligne 10 est vraie et le bloc `then` (dans ce cas la ligne 11 uniquement) est exécuté jusqu'à `fi`.

En suivant la même procédure, vous pouvez à présent remplacer la zone `# Groupe` par une instruction conditionnelle après la variable `groupe`. Si elle est vide, assignez-lui la valeur *users*.

> **Remarque** : la réponse possible d'une condition doit toujours être soit *vraie*, soit *fausse*. Par définition, zéro (0) est faux et n'importe quelle autre valeur vraie.
>
> Une condition `if [1]; then` reste donc toujours vraie. Une telle utilisation est naturellement absurde. Ce qui a par contre du sens, c'est d'utiliser une variable et d'effectuer la vérification `if [ "$<Var>" ]; then` : si elle détient une valeur (n'importe laquelle), la condition est vraie, sinon (si elle est égale à 0 ou vide) elle est fausse.

Boucles : recommence !

Une condition va une nouvelle fois être vérifiée ici. L'instruction suivante s'applique !

tant que <condition vraie> exécute <instruction>

Cette ligne est contenue dans la construction suivante !

```
while [ <condition> ]; do
```

```
   <instruction>
done
```

Ainsi, nous pouvons par ex. imposer la saisie d'un nom d'utilisateur.

Remplacez la zone # Nom à partir de la ligne 2 par !

```
2 while [ -z "$nom" ]; do
3 echo -n "Nom d'utilisateur ! "
4 read nom
5 done
```

Nous trouvons un autre exemple dans le cadre de la manipulation du groupe. Il doit exister. Pour intercepter des erreurs, nous pouvons vérifier dans le fichier /etc/group si le groupe indiqué existe réellement. Si ce n'est pas le cas, nous invitons une nouvelle fois l'utilisateur à nommer le groupe. S'il n'indique aucun groupe, nous utilisons le groupe *users*, le groupe par défaut. Remplacez la zone # Groupe à partir de la ligne 9 par :

```
9  while [ -z $(grep ^$groupe:
/etc/group) ];
       do
10 echo "Groupe [users] ! "
11 read groupe
12 if [ -z "$groupe" ]; then
13    groupe = users
14 fi
15 done
```

C'est déjà un peu plus complexe. Vous aurez peut-être besoin de lire les lignes plusieurs fois pour en saisir complètement le sens.

Ligne 9, le fichier /etc/group est examiné avec grep. La commande recherche une expression correspondant au contenu de la variable groupe. Cette expression doit se trouver au début d'une ligne et être suivie d'un double-point. Pour que cela ait du sens à vos yeux, jetez un oeil dans /etc/group et observez la syntaxe du fichier.

Le fichier /etc/group contient tous les groupes créés sur le système. Comme vous pouvez le remarquer, une ligne est réservée à chaque groupe ; cette ligne commence par le nom du groupe suivi d'un double-point. Le tout pouvant être résumé par l'expression suivante !

```
grep ^$groupe: /etc/group
```

grep renvoyant également 0 ou rien lorsqu'il ne trouve rien, le résultat de la commande peut aussi être analysé avec -z. Nous utilisons pour cela la substitution de commande. Le résultat de la substitution $(<commande>) correspond ici au résultat de grep, c.-à-d. 0, lorsque le groupe n'existe pas.

La condition [-z $(grep ^$groupe: /etc/group)] est donc vraie précisément lorsque grep ne trouve rien. C'est là précisément que le groupe doit être redemandé, d'où les lignes 10 et 11. Par ailleurs, la boucle est exécutée au moins une fois, car une variable à laquelle aucune valeur n'est assignée possède d'abord la valeur 0 ou vide. grep ne renvoie alors rien lors de la première vérification de la condition (car $groupe n'a encore aucune valeur), et la condition est ainsi vraie.

Il y a vraiment de quoi en perdre son latin ! Prenez le temps de parcourir à nouveau la construction jusqu'à ce niveau, jusqu'à ce que vous la compreniez. Donnez-vous suffisamment de temps – vos efforts porteront leurs fruits, promis.

Nous pourrions fermer ici la boucle. Mais il est plus élégant de proposer également une valeur par défaut. Si l'utilisateur ne saisit rien, le groupe *users* est utilisé. La formulation des lignes 12 – 14 est la même que dans notre exemple précédent. La ligne 15 termine la boucle while avec done.

Il est difficile pour un débutant de saisir la logique de construction de certains programmes. Et une fois que vous y êtes parvenu, vous vous demandez si vous parviendrez un jour à réaliser vous-même une telle construction.

J'ai une bonne et une mauvaise nouvelle à vous annoncer. Commençons par la bonne : vous parviendrez à la comprendre et même à l'utiliser. La mauvaise : cela ne sera pas une mince affaire. Faites preuve de patience, d'un peu de créativité et surtout de ténacité.

Comme vous le savez probablement, un utilisateur a besoin en principe d'un mot de passe pour pouvoir se connecter. Nous ajoutons donc la commande passwd au niveau de la dernière ligne.

```
useradd ( ... ) && passwd $nom
```

L'expression useradd reste telle quelle est, mais elle est complétée par un branchement inconditionnel ET (vous en rappelez-vous !) – passwd est invoqué uniquement si useradd a réussi. Logique, non ?

Complément suivant : nous pouvons demander si la saisie est correcte ou si l'utilisateur souhaite effectuer une modification. Cela se présente sous la forme suivante. Cette fois-ci, nous cédons la responsabilité à l'utilisateur :


```
2  ok=n
3  while [ "$ok" = "n" ]; do
4    echo -n "Nom d'utilisateur !
"
5    read nom
6    echo "ok [y/n] ! "
7    read ok
8    if [ "$ok" = "n" ]; then
9    unset nom
10   fi
11 done
```

Ligne 2, notez que l'affectation d'une variable ne tolère aucun espace avant et après le signe d'égalité !

Si nous voulons vérifier une variable oui/non à l'aide d'une boucle while, nous devons définir au préalable la variable de manière appropriée et lui attribuer une valeur.

Ligne 9, la valeur de la variable est rejetée. Il s'agit d'une technique de programmation plus propre.

Paramètres lors du démarrage du script

Intéressons-nous à présent à la façon de transmettre des paramètres lors de l'invocation du script. Exactement comme avec des commandes normales.

Il existe pour cela des variables shell prédéfinies :

$?	valeur de retour de la dernière commande
$!	PID du dernier processus lancé en arrière-plan
$$	PID du shell actuel
$0	nom de fichier du script (chemin inclus)
$#	nombre de paramètres transmis au script shell
de $1 à $9	paramètres de 1 à 9
$* ou $@	ensemble de tous les paramètres transmis

Je ne peux pas vous présenter ici toutes les variables. Je me contente donc de vous en présenter quelques-unes, au cas où vous y seriez confronté au cours de l'analyse d'un script shell du système.

Nous nous intéressons en particulier aux paramètres transmis lors de l'invocation de la commande. Admettons que vous ayez un script gulugulu et que vous l'invoquiez avec gulugulu Par1 Par2. Le paramètre Par1 est alors enregistré dans $1 et Par2 dans $2 et peut être lu de la même façon.

Venons-en maintenant à notre exemple. Les choses commencent déjà à se compliquer un peu !

L'interface graphique de Linux dispose d'une *corbeille* où sont stockés les fichiers supprimés, tandis que les fichiers que vous effacez via la console sont supprimés immédiatement et définitivement. Nous voulons donc créer un script shell qui, dans un premier temps, stockera les fichiers à effacer dans une archive, puis les effacera dans un second temps. Nous sommes ainsi assurés de pouvoir les restaurer. Il doit bien sûr aussi être possible de jeter un oeil au contenu de la corbeille.

Commençons par créer un script répondant à tous ces critères. Nommons-le corbeille.sh, une nouvelle fois sous ~/bin dans le dossier personnel correspondant. Trois *liens symboliques* associés sont créés dans le même répertoire : del, undel et corb. Leur fonctionnement est le suivant :

- del <fichier> – place les fichiers dans l'archive puis les efface
- undel <fichier> – restaure l'original
- corb – affiche le contenu de l'archive

Cela vous semble complexe ? Pourquoi ne pas créer trois programmes individuels ?

L'exemple est un peu élaboré, j'en conviens, mais cela me permettra de vous présenter encore d'autres techniques de programmation utiles :

```
1  #!/bin/bash
2  if[ "$0" = "$PWD/del" ]; then
3    if [ -z "$1" ]; then
4      echo "Usage ! del <fichier>"
5      exit
6    elif [ ! -f "$1" ]; then
7      echo "$1 n'est pas un fichier !"
8      exit
9    fi
10   gunzip /home/archiv/poubelle.tar.gz 2>
/dev/null
11   tar -rf /home/archiv/poubelle.tar $1
2> /dev/null
12   gzip /home/archiv/poubelle.tar
13 fi
```

```
14 if [ "$0" = "$PWD/undel" ]; then
15   if [ -z "$1" ]; then
16     echo "Usage ! undel <fichier>"
17     exit
18   elif [ -z $(tar -tzf
/home/archiv/poubelle.tar.gz|grep $1) ];
then
19     echo "$1 n'est pas dans la cor-
beille !"
20     exit
21   fi
22   tar -xzf /home/archiv/poubelle.tar.gz
-C $PWD $1
23 fi
24 if [ "$0" = "PWD/corb" ]; then
25   tar -tzf /home/archiv/poubelle.tar.gz
2> /dev/null
```

Le programme étant invoqué via les liens `del`, `undel` et `corb`, nous devons vérifier sous quelle forme le script a été invoqué. La condition ligne 2 vérifie si l'invocation provient de `del`. La variable `$0` contient l'invocation accompagnée du chemin. Il s'agirait par exemple de `/home/eric/bin/del`. La variable système `$PWD` contient le chemin (absolu) actuel ; `$PWD/del` nous retourne donc également `/home/eric/bin/del`. Vous pouvez aussi constater le gros inconvénient de notre petit programme : il doit être invoqué directement à partir du dossier contenant le script et ce à l'aide du chemin absolu ☹.

> Dans la partie suivante, nous apprendrons comment résoudre le problème. Ici, je vous fais une démonstration de la variable `$PWD`.

Si cette condition est remplie, la ligne 3 vérifie si un argument a également été livré ; `$1` contient le premier argument transmis. Si ce premier argument est vide, la commande échoue, ce dont l'utilisateur est également informé, ligne 4. Nous quittons ensuite la commande en ligne 5 avec `exit`.

Ligne 6, vous découvrez `elif`. S'il a été défini ligne 2 que le script a été invoqué en tant que `del`, trois situations peuvent se présenter :

1. aucun argument, c'est-à-dire aucun fichier à supprimer n'a été indiqué,
2. un argument a bien été transmis mais il ne s'agit pas d'un fichier existant ou
3. tout va bien...

Ligne 6, nous interceptons le cas n°2. Nous demandons alors à l'aide de `[ ! -f "$1" ]` : si « `$1` est un fichier » devient « n'est pas un fichier », grâce à l'emploi du point d'exclamation. La condition est donc vraie précisément lorsque `$1` ne correspond à aucun nom de fichier existant. Nous en informons l'utilisateur de la même façon qu'à la ligne 7, et nous quittons à nouveau la commande (ligne 8).

Nous terminons ensuite la construction de l'instruction conditionnelle interne avec `fi`, ligne 9.

> **À titre indicatif** : nous nous trouvons encore toujours dans la branche de l'instruction conditionnelle externe de la ligne 2. En témoignent également les indentations du code source. Nous voilà donc parvenus à la situation 3.

A présent, nous nous intéressons à proprement parler à `del` (invoqué dans le *if* à la ligne 2). Comme vous pouvez le voir, dans les lignes 10 à 12, nous redirigeons les messages des commandes vers `/dev/null`, une sorte de « trou noir » signé Linux. Plus clairement, aucun message d'erreur ne sera affiché.

Ligne 11, nous créons l'archive `poubelle.tar` ou, si elle existe déjà, nous y ajoutons le fichier à supprimer. Si elle existe, l'archive doit d'abord être décompressée (ligne 10). Si elle n'existe pas, la commande tombe dans le vide avec le message d'erreur qui en découle. Ligne 12, l'archive est ensuite compressée avec `gzip`. Le bloc `del` ouvertà la ligne 2 est maintenant fermé, à la ligne 13.

La ligne 14 est identique à la ligne 2 et examine s'il s'agit de la commande `undel`. Là encore, il est possible que l'argument ait été omis, cette vérification est traitée aux lignes 15 à 17. Vous avez déjà été confronté à ce cas de figure aux lignes 3 à 5.

Ligne 18, le contenu de `poubelle.tar.gz` est d'abord listé, puis on fait une recherche du fichier (`$1`) à l'aide de `grep`. Si `grep` ne renvoie rien (dans le cas où le fichier n'est pas présent), l'ensemble de l'expression devient vrai par le biais de `-z` et le bloc `then` est exécuté. Message d'erreur et abandon sont gérés aux lignes 19 et 20. La ligne 21 ferme le bloc `if-elif`.

Si l'invocation est correcte, le fichier est restauré à la ligne 22. Le bloc `undel` est fermé.

Les lignes 24 et 25 entrent en action lorsque le script a été invoqué en tant que `corb`. La ligne 25 présente tout simplement le contenu de l'archive `tar`. Aucun `fi` n'est nécessaire car l'ensemble du bloc d'instructions n'englobe qu'une seule ligne. C'est terminé.

Exercice : complétez notre script de façon à ce que les fichiers prélevés de l'archive par `undel` soient ensuite supprimés dans l'archive.

Maintenant détendez-vous et respirez à fond. Ce n'était vraiment pas une mince affaire ! Je vous conseille de faire une petite pause avant de poursuivre.

Découper des chaînes de caractères

Dans les instructions conditionnelles, vous ne pouvez malheureusement pas vous contenter d'utiliser une expression régulière ou un *motif*. Vous pourriez ainsi par exemple vérifier si `*/del` est présent et ainsi éviter le problème du chemin.

J'ai une autre proposition à vous soumettre : coupez simplement le chemin inutile de la variable `$0` et comparez-le à `del`, `undel` et `corb`. Voici la syntaxe :

```
${<variable>#<chaîne_de_caractères>}
```

Va couper la *première* occurrence de la chaîne de caractères en partant de la *gauche* du contenu de la variable. Admettons que vous ayez défini `a=xx:yy:zz`. On obtient donc à partir de

```
${a#*:}
```

Donnera la chaîne de caractères `yy:zz`, car tout ce qui se trouve à gauche de la première occurrence du signe indiqué (le double-point) est coupé, lui-même y compris. L'astérisque (*) est régulièrement utilisé en tant que caractère de remplacement de n'importe quel signe.

```
${<variable>##<chaîne_de_caractères>}
```

Préserve la dernière occurrence de la chaîne de caractères indiquée en partant de la gauche :

`${a##*:}` renvoie `zz`.

```
${<variable>%<chaîne_de_caractères>}
```

Va couper la *première* occurrence de la chaîne de caractères indiquée en partant de la *droite*.

`${a%:*}` renvoie `xx:yy`.

```
${<variable>%%<chaîne_de_caractères>}
```

Va couper la *dernière* occurrence de la chaîne de caractères indiquée en partant de la droite.

`${a%%:*}` renvoie `xx`.

> Faites bien attention à l'emplacement auquel les astérisques doivent être placés, à gauche ou à droite de la chaîne de caractères.

Avez-vous déjà trouvé une piste pour la résolution de notre problème ? La découpe de la dernière occurrence à partir de la gauche fonctionne de la même façon que la découpe de la première occurrence à partir de la droite. Nous choisissons la variante 1 et corrigeons nos lignes 2, 14 et 24, soit en coupant tout le contenu à gauche à partir du dernier `/`.

```
2   if[ "${0##*/}" = "del" ]; then
14  if[ "${0##*/}" = "undel" ]; then
24  if[ "${0##*/}" = "corb" ]; then
```

Le problème du chemin est déjà résolu... Cela paraît très curieux, notre variable `$0` étant affichée sous la forme d'un `0` lorsque le `$` est supprimé. Ce dernier est nécessaire uniquement lors de l'affichage du contenu.

> Ces constructions d'apparence absconses nécessitent un certain temps d'adaptation. Mais vous les retrouvez régulièrement dans des scripts shell. D'après ma propre expérience, il faut insister et insister encore avant de véritablement commencer à se sentir à l'aise. Mis cela vient plus vite qu'on pourrait le croire. Donc, persévérez !

La boucle For

Si vous avez déjà rencontré la boucle For dans d'autres langages de programmation, vous allez devoir changer quelque peu votre vision des choses. La syntaxe est la suivante :

```
for <variable> in <liste>; do
   <instruction>
done
```

La variable prend successivement toutes les valeurs de la liste. Le nombre de cycles de la boucle correspond au nombre de points que comporte la liste. Voici un exemple très simple. Un fichier `liste.txt` contient

```
un
deux
trois
quatre
```

Ainsi

```
for i in $(cat liste.txt); do
  echo $i
done
```

renvoie précisément les quatre valeurs `un`, `deux`, `trois` et `quatre` présentées l'une au-dessous de l'autre.

Pour le côté spectaculaire, on repassera ! Un exemple pratique s'il-vous-plaît ! ...d'accord... Mais je compte sur votre participation, ok ?

Si vous voulez connaître les groupes auxquels appartient votre utilisateur, saisissez `groups` pour obtenir la liste correspondante.

```
eric@ubuntu810:~/bin$ groups
eric adm dialout cdrom plugdev lpadmin admin sambashare
```

Mais comment faire, sans rechercher dans /etc/group et /etc/passwd, pour savoir quels utilisateurs se trouvent dans un groupe bien précis !

Observons la syntaxe des deux fichiers !

/etc/group:

```
<groupe>:x:<GID>:<util1>,<util2>,...
```

/etc/passwd:

```
<user>:x:<UID>:<GID>:Commentaire:<
Rép. perso.>:<Shell>
```

Le script se nomme groupusers et adopte le nom du groupe en tant qu'argument. Sa structure est la suivante !

1. affichage d'un message d'erreur lorsque aucun $1 n'est présent ou si $1 ne correspond à aucun groupe
2. rechercher le GID (ID du groupe) à partir de /etc/group
3. à partir de /etc/passwd, rechercher tous les utilisateurs avec le GID
4. lire les utilisateurs supplémentaires à partir de /etc/group

Si vous avez des difficultés à comprendre ces points, reprenez la partie dans laquelle je vous ai présenté les deux fichiers.

Considérons l'implémentation technique :

Point 1, nous n'avons plus grand chose à ajouter, vous avez déjà appris cela dans le cadre du script de la corbeille.

Point 2, vous obtenez le GID de la façon suivante :

```
gid=$(grep ^$1: /etc/group)
```

Nous affectons d'abord une valeur à la variable gid ! il s'agit de la ligne complète contenant le groupe indiqué par la commande ($1).

```
gid=${gid%:*}
```

Nous recherchons à présent le premier double-point à partir de la droite et nous effectuons une découpe de tout ce qui se trouve à sa droite, double-point inclus. On obtient donc pour gid !

```
<groupe>:x:<GID>
```

S'ensuit une autre construction de ce type !

```
gid=${gid##*:}
```

Cette ligne coupe tout ce qui se trouve à gauche du dernier double-point, le résultat est <GID>. Voilà !

Point 3, pour retrouver l'utilisateur dont le groupe principal correspond au GID du groupe indiqué, nous avons besoin d'une boucle For.

```
for i in $(grep [0-
9]:$gid:.*:.*:.*$ /etc/passwd); do
   echo ${i%%:*}
done
```

Pour cette partie, vous devez garder constamment un oeil sur la syntaxe de /etc/passwd. Analysons la première ligne décisive, de l'intérieur vers l'extérieur ! avec un grep dans /etc/passwd, vous commencez par rechercher une expression localisée à la fin d'une ligne (exprimée par un $ à la fin de l'expression). En lisant de droite à gauche, cette dernière dispose de trois double-points entre lesquels peuvent être placés n'importe quels signes (ce qui est représenté par l'expression RegEx .*). Doit ensuite figurer l'ID du groupe ($gid). Pour s'assurer qu'il s'agit de l'ID du groupe, nous ajoutons à l'expression régulière un autre double-point (la lecture se fait toujours de droite à gauche !) suivi d'un chiffre quelconque. Ce chiffre fait partie de l'UID, l'ID de l'utilisateur.

> Comme vous pouvez le voir, vous pouvez parfaitement utiliser ici des expressions régulières. L'explication est la suivante ! ce n'est pas *bash* qui exploite ces expressions, mais la commande grep, qui les comprend également (contrairement à *bash*).

Point 4, les utilisateurs supplémentaires d'un groupe peuvent être extraits à partir de la ligne correspondante dans /etc/group. C'est précisément ce que nous allons faire !

```
user=$(grep ^$1: /etc/group)
echo "Utilisateurs supplémentai-
res ! ${user ##*:}
```

Nous commençons par chercher la ligne correspondante dans le fichier ; nous indiquons les utilisateurs tels qu'ils apparaissent dans le fichier (séparés par une virgule). Pour ce faire, nous coupons tout ce qui se trouve à gauche du dernier double-point – il reste la liste des utilisateurs. C'est terminé.

Exercice : vous vous êtes peut-être demandé pourquoi et de quelle façon vous allez devoir ici participer à l'exercice ? Réponse : écrivez le programme et exécutez-le. Vous connaissez à présent les points centraux. Vous parviendrez à vous débrouiller avec le reste ☺.

> Sur le blog de *Compétence Mac*, notre magazine dédié à Apple et Mac OSX (une variante d'Unix, comme Linux), nous avons publié un mini tuto qui pourrait vous intéresser :

> http://www.competencemac.com/index.php?2008/08/05/111-le-terminal-est-notre-ami-creer-un-tas-de-dossiers-rapidement
>
> Pour ne pas taper cette adresse à rallonge, tapez simplement « boucle », sans les « », dans le champ de recherche et lisez le billet *Créer un tas de dossiers rapidement*.

Case : suivant le cas...

Avec case, vous disposez d'une variante aux requêtes if.

Dans quasiment chaque script de gestion de service sous /etc/init.d, case sert à distinguer l'argument avec lequel le script a été invoqué (start, stop, etc.). Mais nous pouvons aussi très bien l'utiliser pour nos propres scripts. Par exemple pour distinguer les options des arguments – vous avez peut-être envie d'écrire un script comprenant également des options.

Commençons par la syntaxe :

```
case "<expression>" in
  <comparaison1>) <instruction>;;
  <comparaison2>) <instruction>;;
esac
```

L'expression est généralement une variable, et les valeurs de comparaison sont les valeurs possibles que la variable peut contenir. Chaque ligne d'instruction correspondant à un « cas » est terminée par deux points-virgules.

L'expression de comparaison peut comporter des *motifs* classiques, que *bash* comprend également. L'exemple suivant vous montre comment séparer proprement les options des arguments d'une commande et comment rassembler les options dans une variable et les arguments dans une autre. Nommons par ex. le fichier optarg.sh et plaçons-le également dans le dossier ~/bin.

```
1 #!/bin/bash
2 for i do
3   case "$i" in
4     -* ) opt="$opt $i";;
5     *  ) arg="$arg $i";;
6   esac
7 done
8 echo "Options ! $opt"
9 echo "Arguments ! $arg"
```

La ligne 2 montre une autre application de la boucle for : si elle se présente sans le in qui suit, la variable de la boucle adopte successivement tous les paramètres transmis lors de l'invocation.

La construction case dispose ici de deux cas : d'un côté les paramètres qui commencent par un signe moins, de l'autre tous ceux qui restent. L'ordre est ici essentiel. En effet, les arguments, qui commencent donc par un signe moins –, sont également pris en compte par la deuxième vérification (*). Mais vu que le code utilise toujours d'abord la première branche case et qu'il saute ensuite à la fin de la ramification case, les choses fonctionnent ainsi.

Test : vérifie si...

test vous permet de questionner certains états, par ex. la comparaison d'une variable avec une valeur numérique. Sinon, cela est impossible dans *bash*. L'instruction se trouve souvent dans des scripts shell et constitue une alternative à la formulation de conditions au coeur de boucles ou de ramifications. Il existe trois principales possibilités d'utilisation !

```
test "$a"
```

teste si la variable a contient quelque chose

```
test $x -eq 5
```

eq correspond à *equal* – x est-il égal à 5 ! Outre -eq, vous pouvez utiliser les opérateurs de comparaison suivants :

-gt	greater than – supérieur à
-ne	not eqal – différent de
-lt	less than – inférieur à
-ge	greater or equal – supérieur ou égal à
-le	less or equal – inférieur ou égal à

test -f $a teste s'il existe un fichier avec la valeur de a.

test est une fonction et renvoie une valeur. Il s'agit de 0 pour vrai et de 1 pour faux (*attention, elles sont inversées par rapport à la définition habituelle !*). Cette valeur est contenue dans $? (voir les variables shell prédéfinies) et peut être exploitée à souhait à l'aide d'une distinction de cas.

Nous voilà arrivé au bout de notre petit voyage dans l'univers de la programmation de scripts shell. Vous n'échapperez certainement pas aux messages d'erreur et autres moments de frustration. Je ne dis pas ça pour vous décourager, mais parce que c'est la réalité et ça fait partie du plaisir : découvrir que no-

tre savoir faire s'améliore et, aussi un peu quand même, chasser le bug.

Sur ce, bon amusement avec les scripts shell !

Le mot de la fin

Nous voilà déjà arrivés à la fin de ce deuxième volet sur Ubuntu-Linux. J'espère qu'il vous a plu et qu'il a su vous donner envie de pousser un peu plus loin l'expérience Linux.

Nous avons abordé de nombreux thèmes auxquels vous confronte d'entrée votre nouveau système d'exploitation Ubuntu. Que ce soit l'adaptation du bureau à vos goûts, l'utilisation des logiciels de gestion, l'installation de logiciels utiles et les mesures de sécurité. Vous avez appris une multitude de choses, en particulier concernant le système de cryptage GnuPG dont l'utilisation n'est pas totalement intuitive. Vous êtes également parti à la découverte de l'utilisation et de la programmation shell.

Et pourtant, il reste encore tellement de choses à découvrir... peut-être nous croiserons-nous bientôt au détour d'un chemin ou d'une page !

En attendant, je vous souhaite bon amusement dans le monde fantastique de Linux – Ubuntu !

Eric Amberg

Index

.bash_history, 75
.bash_logout, 74
.bashrc, 73
.profile, 73

Adresse IP, 34
ADSL, 32
Amarok, 29
aMSN, 30
Anonyme
  FTP, 46
Anti-virus
  programme, 64
APT, 7
apt-cache, 9
apt-get, 8
aptitude, 9
ARP, 42
Asymétrique
  cryptage, 52
ATI, 25
Audacity, 31
Audio
  éditeur, 31
Audio, codecs, 27
avahi, 22

Backport
  dépôts, 28
Backports, 28
  installer, 29
Bash, 74
Bit, 34
Bluefish, 30
bluetooth, 23
Boucles, 83
Boucles For, 87
Broadcast, 35
Bureau
  adapter, 10
  adapter l'arrière-plan, 11
  GNOME, 4, 13, 18
  installer des thèmes de bureau, 12
  KDE, 4, 18
  remplacer, 18
  thèmes, 12
  thèmes de bureau pour tous les utilisateurs, 13
  Xfce, 19
Bureau à distance
  Linux sur Linux, 45

CA, Certificate
  Authority, 52
Carte graphique
  adapter le pilote, 26
Case, 89
cat, 78
CAT 5, 32
Certificat, 57
Certificats, 52
Chain of Trust, 52
Chaînes de
  caractères
  découper, 87
chmod, 81, 82
Clé
  annuler, 57
  exporter, 54
  importer, 54
  mettre à disposition, 56
  signer, 55
  vérifier, 55
Clé privée, 52
Clé publique, 52
Clés
  serveur de, 56
Compresser avec tar, 79
Conditions, 83
Contrôle
  sommes de, 50
Copie de secours, 79
Cryptage avec
  TrueCrypt, 60
Cryptage dans
  Evolution, 57, 58, 59
CSS
  éditeur, 30

D4X, 29
Daemons (démons), 20
date, 78
D-Bus, 23
Démon HAL, 23
Dépôts, 6
Détection
  d'anomalies sur le réseau, 38
DHCP, 35
  serveur, 36
Dia, 30
Différentielle
  sauvegarde, 67
Distinction de cas, 89
DIVX, 28
dmesg, 65
DNS, 35
Domain Name
  System, 35
Dossiers
  crypter, 59, 60
DSL
  routeur, 33

echo, 82
Écran
  configuration, 10
Écrans
  reconnaître, 10
ecryptfs, 60
Éditeur HTML, 30
elif, 83
Emails
  crypter, 57, 58, 59
Environnement
  utilisateur, 72
eth0, 38
Ethernet, 32

Fichiers
  crypter, 59
Fichiers journal, 64
Filtre ADSL, 32
find, 76
Finger, 41
Flash, 27
FTP
  accès via Nautilus, 47
  accès via une console, 48
  serveur, 46

gdm, 18
Gestionnaire de
  réseau
  icône, 37
gFTP, 48
GFVS, 23
GNOME, 18
  par défaut, 4
GnuPG, 52, 53
grep, 77, 84
GStreamer, 28
gtkhash, 50, 51
gunzip, 80
gzip, 80

Hachage
  algorithmes de, 50
  cryptage, 51
hald, 23
Hôte, 34
HTML
  éditeur, 30

ICMP, 35
id, 79
if, 83
Incrémentale
  sauvegarde, 67
init, 21
Inkscape, 30
Installation, 4
  à partir d'Internet, 6
Internet
  la connexion ne fonctionne pas, 42
Intrepid Ibex, 3
IP, 33
  configurer manuellement, 38
ipconfig, 39
IPv4, 33

Jeux, 31
Journal
  fichiers, 64

KDE, 18
  installer
    séparément, 18
kill, 21
klogd, 22
Kubuntu, 18

Lecteur audio, 29
Lecteur vidéo, 28
Lemmings, 31
locate, 77
Logiciels
  sources de
    paquets, 6
Logiciels payants, 27

MAC
  adresse, 42
Main, 7
Masque de sous-
  réseau, 34
MD5, 50
messages, 65
Microsoft
  polices, 15
Mises à jour
  non supportées, 29
Mises à jour
  automatiques, 63
Modèle
  dossier utilisateur,
    73
Mot de passe sûr, 63
MP3, 28
  installer un lecteur
    MP3, 9
MPEG2, 28
Multiverse, 7

Nautilus, 43
  accès FTP, 47
netstat, 40
nmap, 41
nm-system-settings,
  23
Noyau
  processus, 22
Nvidia, 25

Octet, 34

PAO, 30
Paquets
  gestionnaire de
    paquets
    aptitude, 9
  sources de paquets
    pour logiciels, 6
Pare-feu, 64
Partages Windows,
  42
Passerelle par défaut,
  34
  vérifier, 42
passwd, 84
PATH, 75
Payants
  logiciels, 27
PHP
  éditeur, 30
Pilote
  pour ATI et
    Nvidia, 25
Pilotes propriétaires,
  25
ping, 39, 41
Pingus, 31
Ports, 35
Ports reconnus, 35
Private
  dossier, 60
Processus
  définition, 20
  gérer, 21
Programmes
  gérer, 5
Propriétaires
  pilotes, 25
Protection des
  données, 66
pstree, 21
Pulseaudio, 23
px, 20

Réseau, 32
  gestionnaire, 23,
    37
  Mac OSX, 43
  outil de diagnostic,
    38
  Windows, 42
reset, 60
Restricted, 7
RJ 45, 32
root
  compte, 62
Routeur, 34
Runlevel
  dossiers, 25

Sauvegarde
  complète, 67
Sauvegarde
  différentielle, 67
Sauvegarde
  incrémentale, 67
Sauvegardes, 66
  CD et DVD, 71
  créer une archive
    Tar, 79
  planifier, 67
  quoi sauvegarder
    ?, 66
  restaurer des
    données, 70
Sauvegardes de test,
  69
Scan de ports, 41
Scribus, 30
Script
  paramètres lors du
    démarrage, 85
Seahorse, 53
Services, 20
  administration, 23
  désactiver, 24
  scripts
    d'initialisation,
    24
SHA-1, 50
SHA-256, 50
Shell, 72
  astuces
    d'utilisation, 74
  programmation
    d'un script, 81
  variables, 75
Signature d'une clé,
  55
Simple Backup
  Config, 67, 68, 69
Simple Backup
  Restore, 70
Somme de
  vérification
  contrôler, 50
sort, 78
Sound, 23
SSH, 49
sudo, 61
SuperTuxKart, 31
Switch, 32
Symétrique
  cryptage, 51
Synaptic, 5
  APT, 7
  options des
    paquets, 5
  sélectionner des
    paquets, 5
syslog, 65
syslogd, 22

Table de routage, 39
tar, 79
TCP, 34
TCP/IP, 33
tee, 74
Téléchargement
  gestionnaire de, 29
  vérifier, 50
Terminal
  invoquer une
    console
    terminal, 22
  processus, 22
test, 89
Thèmes, 12
top, 64
Totem, 28
traceroute, 40
TrueCrypt, 60

Ubuntu
  variantes, 4
UDP, 34
Universe, 7
useradd, 84
Utilisateur
  créer, 73

Vidéo, codecs, 27
VNC, 45

Web
  éditeur de pages
    Web, 30
Web of Trust, 55
whoami, 79
Whois, 41
WPA_Supplicant, 23

Xfce, 19
Xubuntu, 19

Catalogue Compétence Micro

Pour obtenir une liste complète des publications disponibles, allez sur le site de Compétence Micro à www.competencemicro.com pour consulter une version en ligne ou télécharger la liste en PDF.

Divers	Paru	Réf.	Prix
Réseau et réseaux WiFi	03/06	0537	**6,00**
Gravez vos CD et DVD avec Nero 7	06/06	0550	**6,00**
Internet	**Paru**	**Réf.**	**Prix**
Créez votre site Web, HTML & CSS	03/06	0542	**6,00**
CSS, le Web avec du style	03/02	4487	**5,80**
CSS, un jeu d'enfant	12/06	0561	**6,00**
Firefox et Thunderbird	10/06	0556	**6,00**
Freewares, les indispensables	12/05	0527	**6,00**
Internet : à vos marques ...	02/05	4488	**6,00**
Internet, allez plus loin !	08/05	0520	**6,00**
Joomla 1.4.x & 1.5	05/07	0575	**6,00**
Le Web 2.0 pour tous	01/07	0563	**6,00**
Les mystères de PHP5 et MySQL	05/06	0549	**6,00**
Les Weblogs	05/04	4475	**6,00**
PHP 5	01/06	0533	**6,00**
Protégez votre PC	05/05	0514	**6,00**
Soyez numéro 1 sur Google	12/05	0529	**6,00**
Votre site Web facile et gratuit avec NVU	08/06	0552	**6,00**
Dreamweaver, un jeu d'enfant	05/08	64001	**6,00**
Joomla !	01/09	64014	**7,00**
Vidéo	**Paru**	**Réf.**	**Prix**
Du VHS au DVD	01/05	0500	**6,00**
Pinnacle Studio 9 (plus)	11/04	4497	**6,00**
Premiere Elements	07/05	0518	**6,00**
Premiere Pro	06/04	4477	**6,00**
Windows Movie Maker 2	09/05	0522	**6,00**
La vidéo de A à Z	12/07	0592	**6,00**
Programmation, script	**Paru**	**Réf.**	**Prix**
Ajax, un jeu d'enfant	07/07	0581	**7,00**
VBA sous Word, Initiation à la programmation	01/03	4420	**5,80**
JavaScript : bien débuter	03/00	429	**5,49**
Javascript niveau 2	09/02	4407	**5,80**
Les mystères de PHP5 et MySQL	05/06	0549	**6,00**
Photographie, graphisme	**Paru**	**Réf.**	**Prix**
Illustrator CS et CS2	11/05	0526	**6,00**
Les Freewares	12/05	0527	**6,00**
Photo numérique : 1, 2, 3 partez !	06/07	0578	**6,00**

Photographie, graphisme	Paru	Réf.	Prix
Photoshop 7	10/02	4409	**5,80**
Photoshop 7 niveau 2	10/03	4441	**6,00**
Photoshop CS	02/05	0503	**6,00**
Photoshop Elements 5	03/07	0568	**6,00**
Plus loin avec Photoshop CS2	04/06	0541	**6,00**
Photoshop CS3, les nouveautés	08/07	0583	**6,00**
Trucs et astuces pour Photoshop	09/06	0554	**6,00**
Traitez vos photos avec XnView	05/05	0511	**6,00**
InDesign CS à CS3	11/07	0587	**9,00**
Picasa et Gimp	02/08	0596	**6,00**
Photoshop, un jeu d'enfant	06/08	64004	**9,00**
Picasa	09/08	64007	**6,00**
Lightroom, un jeu d'enfant	11/08	64012	**6,00**
Bureautique	**Paru**	**Réf.**	**Prix**
Access XP	02/03	4466	**5,80**
Excel 2007 volume 1	05/07	0571	**7,00**
Excel XP niveau 1	11/02	4416	**5,80**
Excel XP niveau 2	01/03	4467	**5,80**
Office 2007	07/07	0576	**7,00**
Open Office – Calc	01/06	0534	**7,00**
Open Office – Writer	12/05	0530	**6,00**
PowerPoint 2007	05/07	0572	**7,00**
PowerPoint XP	06/03	4435	**5,80**
Word 2003	02/05	0502	**6,00**
Word 2007	08/07	0535	**7,00**
Systèmes d'exploitation	**Paru**	**Réf.**	**Prix**
La base de registre	01/02	0473	**5,80**
Les Freewares	12/05	0527	**6,00**
Linux Ubuntu, un jeu d'enfant	10/07	0585	**6,00**
Mon premier PC avec Windows XP	05/02	494	**5,80**
Windows en pleine forme	03/05	0509	**6,00**
Windows Vista, pas à pas !	02/07	0565	**6,00**
Windows XP	04/02	490	**5,80**
Windows XP trucs et astuces	03/03	4427	**5,80**
Vista, panneau de configuration	11/07	0588	**6,00**
Son et musique	**Paru**	**Réf.**	**Prix**
Musique et vidéo	10/05	0524	**6,00**