

Mai 2011

3

Muffin

Le magazine de la communauté Fedora francophone

FEDORA 15

Lovelock dans les blocs !

GNOME 3
en **7** questions



Les versions
personnalisées
de Fedora



Utilisation
de YUM :
allons plus loin



Une
introduction
au langage D

ÉDITORIAL

BONJOUR chers lecteurs ! Voici enfin un nouveau numéro de Muffin, tant attendu si je puis dire ? Pour l'équipe du magazine ainsi que pour moi, assurément !

Le Magazine des Utilisateurs Francophones de Fedora Indubitablement Novateur est plus que jamais présent pour accompagner la distribution qui nous est si chère à la sortie de Lovelock, la quinzième version mise au point par le projet Fedora. Dans ce numéro, nous allons partir à la conquête de la lune ! Comme à l'accoutumée, Fedora tient ses promesses d'innovation avec toutes sortes de nouveautés, de GNOME 3 à LibreOffice, en passant par Systemd pour ne citer qu'elles. Fedora 15 s'annonce d'ores et déjà comme un excellent cru, à n'en pas douter. Et Muffin se veut le reflet de cette réinvention constante de Fedora.

J'achèverai cette introduction par des remerciements sincères et mérités à l'ensemble des contributeurs de ce numéro qui ont durement œuvré en cuisine pour vous servir sur un plateau l'appétissant magazine que vous vous apprêtez à déguster. Nous avons en tous cas fait en sorte que ce numéro soit aussi délectable que les précédents, si ce n'est plus.

Il ne me reste plus qu'à vous souhaiter une bonne dégustation à la lecture de ce numéro 3, en attendant la suite du menu avec un numéro 4 plein de surprises...

Alexandre FRANDEMICHE

Fedora 15, un tour d'horizon 3

De nouveaux environnements • Un nouveau noyau • De nouveaux logiciels • Et bien plus encore...

GNOME 3 en sept questions. 6

Pourquoi une nouvelle version de GNOME ? • À quoi ressemble GNOME 3 ? • Ma machine pourra-t-elle exécuter GNOME 3 ? • Retrouverai-je mes applications favorites ? • Qu'en est-il de l'accessibilité ? • Comment personnaliser mon environnement ? • Pourrai-je retrouver l'ancienne interface de GNOME ?

Les paquets Fedora du moment 15

Lundi productif • Mardi artistique • Le « pourquoi ? » du mercredi • L'interface du jeudi • Vendredi divertissement

Les versions personnalisées de Fedora ... 18

Choisir l'environnement graphique • Choisir en fonction des programmes préinstallés

Changer le fond d'écran de GDM 20

Une introduction au langage D 21

Introduction • Ce dont vous avez besoin pour développer en D • Votre premier programme • Les bases du langage • Conclusion

Empathy et les thèmes 27

Un centre multimédia avec XBMC 28

Présentation • Installation et configuration sous Fedora • Premiers pas avec XBMC • Aller plus loin

Changer de gestionnaire de connexion. ... 31

Les évènements à ne pas manquer 32

Fedora-fr.org change de look • Solutions Linux 2011 • Rencontres Fedora 15 • Rencontres mondiales du logiciel libre 2011 • Restez informés

Réaliser des images du système. 33

Utilisation de YUM : allons plus loin 34

Quelques subtiles options de YUM • Retour vers le passé • Un interpréteur de commandes dans YUM

La gravure (trop) facile 36

Nous rejoindre en cuisine 37

Notre équipe de rédaction • Comment participer ?

Fedora 15, un tour d'horizon

La nouvelle version de la distribution Fedora, Lovelock, est sortie en grande pompe le 24 mai 2011. Fruit de six mois de développement, elle est sans doute l'une des versions les plus révolutionnaires que le projet Fedora ait jamais portées. De GNOME 3 à Systemd, en passant par LibreOffice, nous vous invitons à découvrir dans cet article les nouveautés les plus remarquables de Fedora 15.

De nouveaux environnements

GNOME 3

L'INNOVATION la plus visible de Lovelock est sans nul doute GNOME 3, une mise à jour majeure de l'environnement graphique par défaut de Fedora. À travers GNOME Shell, un nouveau gestionnaire de bureau faisant également office de tableau de bord, GNOME 3 se veut résolument novateur, simple et ergonomique. Parmi la pléthore de changements introduits dans cette version par rapport aux précédentes, signalons :

- l'introduction du concept d'activités, à travers une vue d'ensemble depuis laquelle il est aussi bien possible de lancer ses applications que d'organiser les fenêtres ouvertes ou encore de rechercher des fichiers récents (voir figure 1 page suivante) ;
- un système de notifications revu et corrigé : les alertes sont moins intrusives et sont enregistrées jusqu'à ce que l'utilisateur soit prêt à les lire ;
- l'intégration de la messagerie instantanée à l'environnement, avec la possibilité de répondre aux messages directement depuis les notifications sans perdre de vue son travail ;
- un panneau de configuration dans lequel tous les outils de réglage de GNOME sont centralisés.

Les nouveautés apparues dans GNOME 3 ne se limitent cependant pas au bureau : les applications majeures de GNOME telles que Nautilus, Evolution ou encore Empathy ont été mises à jour pour cette version et tirent désormais parti de la bibliothèque graphique GTK+ 3, parmi d'autres modifications¹.

Notre article dédié à GNOME 3 en page 8 vous éclairera sur les concepts qui ont orienté son développement et vous guidera dans sa prise en main.



KDE 4.6

Les utilisateurs de KDE n'ont pas été négligés puisque leur environnement favori a également été mis à jour, passant en version 4.6. Cette nouvelle version apporte, outre les habituelles corrections de bogues, un support amélioré de la gestion de l'énergie et du Bluetooth ainsi qu'un meilleur rendu du gestionnaire de fenêtres KWin, entre autres raffinements².



Autre évolution, Phonon, le framework multimédia de KDE, utilise désormais GStreamer en tant que moteur multimédia par défaut, ce dernier se substituant à Xine.

Notons enfin que Fedora 15 distribue KDE avec Oxygen-Gtk, un port de son thème emblématique Oxygen destiné aux applications GTK+ : son intérêt est de donner à ces dernières une apparence cohérente avec le reste de l'environnement lorsqu'elles sont exécutées dans KDE (voir figure 2).

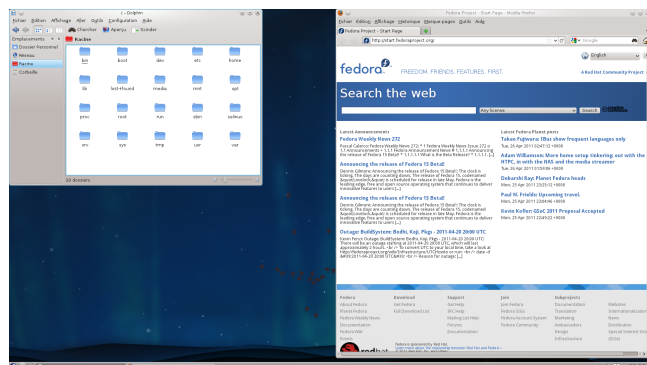


Fig. 2 KDE 4.6 et Firefox parfaitement intégré à l'environnement grâce au thème Oxygen-Gtk

Mais aussi...

Xfce 4.8 Fedora 15 propose la dernière version de l'environnement de bureau léger Xfce ; en plus d'inclure un tableau de bord complètement réécrit, cette version devrait améliorer notablement le support du multi-écrans ainsi que l'utilisation de la mémoire. Elle introduit également dans le navigateur de fichiers Thunar le support des lecteurs distants.

Sugar 0.92 Cette mise à jour de maintenance de l'environnement attiré de l'OLPC³, est également disponible dans Fedora 15.

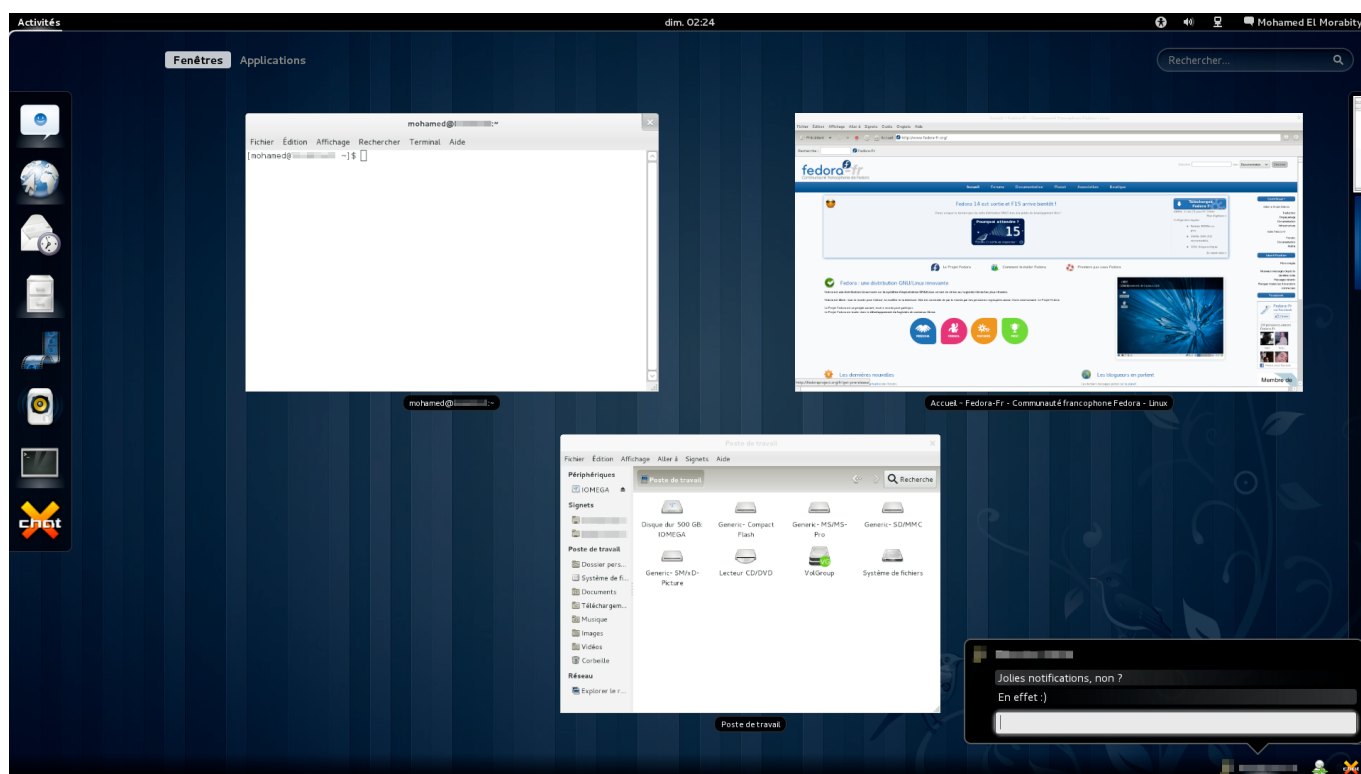


FIG. 1 GNOME 3, l'aperçu des activités et les notifications de messagerie instantanée

Un nouveau noyau

Lovelock repose sur la branche 2.6.38 du noyau dont les nouvelles fonctionnalités améliorent les performances du système ainsi que la prise en charge du matériel.

Cette version introduit ainsi les *Transparent Huge Pages*, autorisant, comme leur nom l'indique, l'utilisation de pages mémoire de plus grande taille (jusqu'à 4 Mio contre 4 Kio auparavant) et dont devraient tirer avantage, en termes de performance, les applications particulièrement gourmandes en mémoire telles que les serveurs de bases de données.

La couche VFS — la couche d'abstraction du système de fichiers — a également subi des modifications en profondeur puisque sa fonction de résolution de chemin (celle associant un chemin à un *inode*) a été optimisée pour cette version, améliorant du coup sensiblement tous les accès fichiers.

Enfin, les utilisateurs de pilotes graphiques libres se réjouiront de cette nouvelle mouture du noyau puisque le pilote nouveau prend désormais en charge les cartes Nvidia GeForce 400/500 (y compris le support de la 3D), le pilote radeon gère enfin les circuits graphiques des puces AMD Fusion ainsi que les cartes Radeon de la série HD 6000, enfin le pilote intel apporte une meilleure gestion de l'énergie.

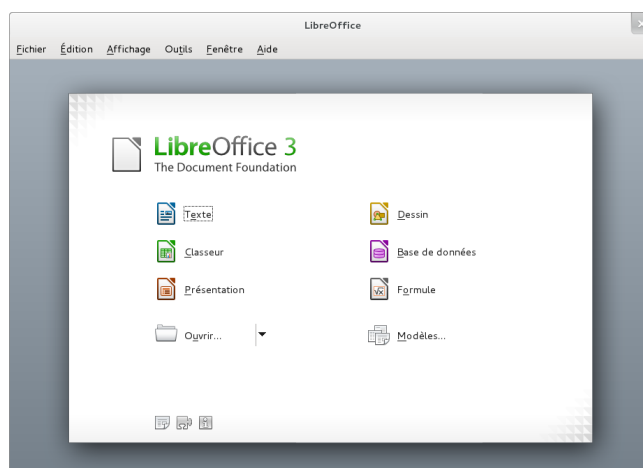


FIG. 3 LibreOffice et sa nouvelle charte graphique

De nouveaux logiciels

LibreOffice, la bureautique libérée



Fedora soutient, comme la plupart des distributions Linux majeures, la suite bureautique LibreOffice qui

remplace désormais OpenOffice.org à partir de Fedora 15.

Pour rappel, suite au rachat de la société Sun, à l'origine du projet OpenOffice.org, par Oracle, les principaux contributeurs à la suite bureautique ouverte se sont interrogés sur sa pérennité. Ces doutes les ont amenés à constituer en septembre 2010 une fondation indépendante, la *Document Foundation*⁴, et, surtout, à développer un *fork* d'OpenOffice.org, LibreOffice.

LibreOffice 3.3, première version de ce nouveau projet, est ainsi disponible dans Lovelock. S'il est sans doute trop tôt pour parler de véritable démarcation avec OpenOffice.org au niveau des fonctionnalités, relevons cependant quelques nouveautés exclusives telles que l'apparition d'une aide en ligne accessible depuis l'application, l'importation désormais possible des images au format SVG ou encore le support des documents Microsoft Works et Lotus Word Pro⁵.

Firefox 4, le renouveau du Web

La dernière version du navigateur de la fondation Mozilla, Firefox 4, fait également son entrée dans Fedora 15. Cette version se distingue des précédentes par une nouvelle interface, plus simple et épurée (les onglets se trouvent par exemple désormais au-dessus de la barre d'adresse et le bouton « Recharger » est discrètement intégré à cette dernière), qui doit laisser plus de place au contenu Web (voir figure 4 page suivante). L'interface du gestionnaire d'extensions a aussi été revue pour une meilleure ergonomie. Autre fonctionnalité visible, Panorama, un mode de gestion alternatif des onglets par glisser-déposer sur un panneau, permet de visualiser d'un coup d'œil les pages ouvertes.



Les nouveautés de Firefox 4 ne se limitent cependant pas à des changements esthétiques : le navigateur intègre ainsi désormais le module Sync⁶ par défaut, qui permet de synchroniser ses réglages et même ses onglets ouverts entre deux instances de Firefox, et du coup de retrouver son environnement Internet où que l'on soit. Cette version se distingue également par de meilleures performances grâce à des améliorations apportées à son moteur de rendu Gecko, mais aussi grâce à son nouveau moteur JavaScript, JägerMonkey.

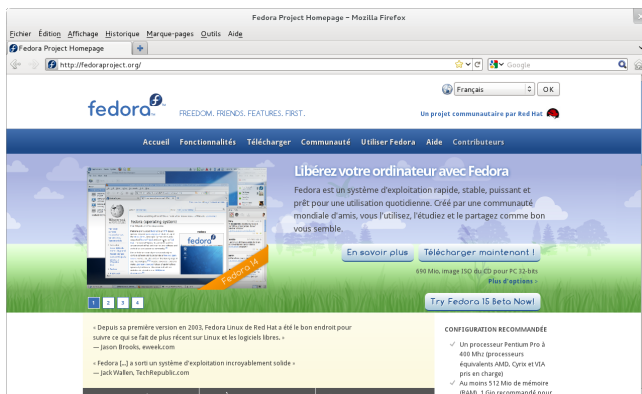


Fig. 4 Firefox 4 et sa nouvelle interface

Systemd, un démarrage du système encore plus rapide

Déjà disponible dans Fedora 14 en option, le système de démarrage Systemd^[1] est désormais intégré par défaut dans Fedora 15, remplaçant du coup Upstart jusqu'ici utilisé. Développé par Lennart POETTERING (à qui l'on doit également PulseAudio), Systemd est aussi bien capable de lancer les services en parallèle (tout en tenant compte de leurs éventuelles interdépendances) que d'éliminer le chargement de ceux inutiles. Le démarrage du système s'en trouve par conséquent considérablement accéléré.

Si Systemd définit ses propres fichiers de service (des fichiers `.service` placés dans le dossier `/lib/systemd/`), il reste cependant totalement compatible avec les traditionnels scripts d'`init` SysV définis dans le dossier `/etc/init.d/`. De plus, les commandes `service` et `chkconfig`, ainsi que l'interface de gestion des services `system-config-services`, ont été adaptées pour Systemd.

Le projet Fedora consacre une page^[2] à la présentation et à l'utilisation de Systemd qui devrait répondre aux questions les plus fréquentes concernant ce gestionnaire de services. Notons également la série de billets plus techniques rédigés par Lennart sur son blog à l'intention des administrateurs^[3].

Une plate-forme de développement toujours en pointe

Fedora est, de par sa politique d'innovation constante, une plate-forme de choix pour les développeurs souhaitant être au fait des dernières technologies Linux. Lovelock ne faillira pas à ce credo puisque, comme pour chaque nouvelle version de Fedora, celle-ci s'enrichit de nombreuses mises à jour des outils de développement.

La suite de compilateurs GCC passe ainsi en version 4.6 : incluant le support du langage de programmation Go de Google, cette version améliore également pour le compilateur g++ le support de la norme C++0x. Toujours pour C++ d'ailleurs, la dernière version de la bibliothèque Boost, la version 1.46, est également disponible dans Fedora 15. Cette dernière propose enfin une pré-version de la prochaine version 7.3 du débogueur GDB.

Les développeurs Web n'ont pas été oubliés puisque le *framework* Ruby On Rails est disponible dans sa dernière version 3.0.3. Signalons

par ailleurs l'arrivée dans les dépôts de Fedora 15 de Tryton^[4], un nouvel ERP écrit en Python, Python justement qui, dans sa branche 3, est disponible en version 3.2.

Et bien plus encore...

Cet article est loin de décrire toutes les nouveautés de Lovelock. Nous vous engageons, pour aller plus loin dans cette découverte de Fedora 15, à parcourir les fonctionnalités validées par le projet Fedora pour cette version^[5], à consulter également ses notes de version^[6], mais surtout... à l'essayer.

Comme les précédentes versions de Fedora, Lovelock est disponible en libre téléchargement sur les miroirs du projet Fedora^[7] ainsi que sur le réseau BitTorrent^[8], pour les environnements 32 et 64 bits, à partir du 24 mai. Vous avez également la possibilité de mettre à niveau votre version actuelle de Fedora grâce à PreUpgrade^[9].

N'hésitez pas à solliciter la communauté Fedora francophone^[10] pour vous épauler dans votre découverte de Fedora 15, via sa documentation^[11], son forum^[12] ou encore sur IRC^[13].

Mohamed EL MORABITY



- [1] <http://library.gnome.org/misc/release-notes/3.0/index.html.fr>
- [2] <http://www.kde.org/announcements/4.6/>
- [3] <http://one.laptop.org/>
- [4] <http://www.documentfoundation.org/>
- [5] <http://www.libreoffice.org/download/new-features-and-fixes/>
- [6] <http://support.mozilla.com/fr/kb/quest-ce-que-firefox-sync>
- [7] <http://www.freedesktop.org/wiki/Software/systemd>
- [8] <https://fedoraproject.org/wiki/Systemd>
- [9] <http://0pointer.de/blog/projects/systemd-for-admins-1.html>
- [10] <http://www.tryton.org/fr/>
- [11] <http://fedoraproject.org/wiki/Releases/15/FeatureList>
- [12] <http://docs.fedoraproject.org/release-notes/>
- [13] <http://mirrors.fedoraproject.org/publiclist/Fedora/15/>
- [14] <http://torrent.fedoraproject.org/>
- [15] http://doc.fedora-fr.org/wiki/PreUpgrade:_Passer_%C3%A0_la_version_sup%C3%A9rieure_de_Fedora
- [16] <http://www.fedora-fr.org/>
- [17] <http://doc.fedora-fr.org/>
- [18] <http://forums.fedora-fr.org/>
- [19] Canal #fedora-fr sur le serveur Freenode



GNOME 3 en sept questions

La nouvelle version de l'environnement de bureau GNOME est sortie le 6 avril 2011 et Fedora, fidèle à son credo d'innovation, sera la première distribution à la proposer par défaut dans Lovelock. Placée sous le signe de l'ergonomie et de l'efficacité, cette version introduit, au travers du gestionnaire de fenêtres GNOME Shell, un nouvel environnement utilisateur élégant et original.

Nous vous proposons, au travers de sept questions que vous pourriez vous poser à ce sujet, un tour d'horizon de GNOME 3 qui vous aidera à mieux l'appréhender.

Pourquoi une nouvelle version de GNOME ?

L'ENVIRONNEMENT de bureau GNOME est indéniablement populaire, au point d'être le bureau par défaut de nombreuses distributions Linux, voire d'être considéré *de facto* comme le bureau Linux par excellence. Cependant, son design, du moins tel qu'on le connaissait jusqu'ici avec GNOME 2, avec un panneau applicatif en haut de l'écran et une liste de fenêtres en bas (voir figure 5), n'avait guère évolué depuis sa première version publiée en 2002. Le poids des années s'est donc fait sentir pour GNOME tandis que d'autres environnements majeurs, qu'ils soient libres (KDE par exemple) ou propriétaires (Aero pour Windows, Aqua pour Mac OS X), se sont entre-temps renouvelés.

Par ailleurs, de l'aveu même de ses développeurs, GNOME 2 souffre de quelques défauts nuisant à son ergonomie¹, particulièrement pour les néophytes. Le projet GNOME a ainsi identifié dans GNOME 2 les composants suivants, parmi d'autres, susceptibles d'être améliorés :

- la liste des fenêtres, dans laquelle il devient difficile d'identifier une application particulière ouverte dès qu'il y en a plus de quatre ;
- les espaces de travail qui, en dépit de leur utilité, sont souvent dédaignés par les utilisateurs novices ;
- le menu applicatif, avec ses multiples rubriques imbriquées, depuis lequel lancer une application n'est peut-être pas aussi intuitif qu'on pourrait l'espérer d'un des éléments les plus importants de l'environnement ;
- la personnalisation des tableaux de bord, notamment à travers l'ajout d'*applets* ou la possibilité de les placer n'importe où sur le bureau, superflue au regard du peu de gain retiré par les utilisateurs.

Outre ces défauts d'ordre ergonomique, la conception par trop modulaire de GNOME 2, loin d'apporter dans le cadre du développement d'un environnement de bureau une robustesse et une flexibilité supposées, nuit également à la cohérence de ses différents composants. GNOME 2 est en effet la réunion de plusieurs programmes tels que le tableau de bord GNOME Panel, le gestionnaire de fenêtres Metacity, le navigateur de fichiers Nautilus ou encore les divers services visibles au travers des icônes dans la zone de notifications (NetworkManager par exemple). Chacun de ces composants a été conçu *indépendamment* des autres et ne se comporte pas forcément de façon homogène avec l'ensemble de l'environnement, nuisant au final à l'harmonie de GNOME 2².

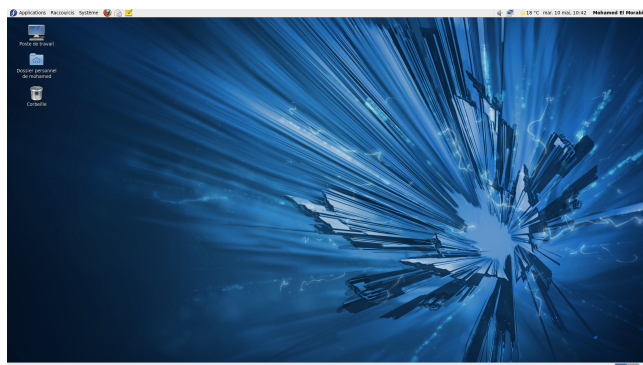


Fig. 5 GNOME 2 et son interface (ici GNOME 2.32 sous Fedora 14)

Il est dès lors apparu clairement aux développeurs de GNOME qu'une réponse efficace et rapide à ces problématiques ne pouvait s'exprimer au travers d'une évolution de GNOME 2, mais par une réécriture complète de l'environnement dans le cadre d'une nouvelle version majeure. Les développements autour de GNOME 3 ont ainsi été initiés dès la fin 2008, parallèlement à ceux de GNOME 2.

À quoi ressemble GNOME 3 ?

Refondre complètement l'interface de GNOME selon les paradigmes énoncés plus haut a constitué à juste titre un défi énorme pour tous les contributeurs du projet GNOME. Si le risque était grand pour ces derniers de dérouter nombre d'utilisateurs conditionnés par l'interface de GNOME 2, voire de les perdre, le pari a néanmoins été pris de proposer une nouvelle expérience graphique avec en contrepartie un bureau plus intuitif, plus simple et plus efficace pour tous.

Le résultat de ce travail de longue haleine est GNOME Shell, intégré au cœur même de GNOME 3, qui se substitue non seulement à GNOME Panel, mais fait également office de gestionnaire de fenêtres : il est en effet bâti comme une extension de Mutter, un gestionnaire de fenêtres basé sur le vénérable Metacity³ et s'appuyant sur la bibliothèque Clutter⁴, tirant ainsi profit de l'accélération 3D de la carte graphique pour le rendu. GNOME Shell répond ainsi à la fragmentation

de GNOME 2 en modules hétéroclites. Particularité de GNOME Shell, celui-ci est écrit en C ainsi qu'en Javascript et s'appuie aussi sur des feuilles de style CSS pour définir son apparence. Parcourons donc immédiatement et plus en détail les différents composants de GNOME Shell.

Le tableau de bord

Comme on peut le voir sur la figure 6 page suivante, GNOME Shell comporte un unique tableau de bord *statique*, composé de trois parties :

- à gauche, un bouton *Activités* permettant d'accéder à la vue des activités, que l'on découvrira plus loin ;
- au centre, une horloge, qui donne accès d'un clic à un calendrier (voir figure 7), synchronisé comme dans GNOME 2 avec les agendas configurés dans Evolution ;
- à droite enfin (voir figure 8), une zone de statut donnant accès :
 - à la configuration des outils d'accessibilité (voir la question « Qu'en est-il de l'accessibilité ? ») ;
 - aux paramètres du son (au travers de PulseAudio) ;
 - au Bluetooth, le cas échéant ;
 - aux réglages réseau (via NetworkManager) ;
 - à la gestion de l'énergie, sur les PC portables notamment ;
 - aux préférences utilisateur enfin ; ces dernières sont rassemblées au sein d'un menu permettant entre autres à l'utilisateur de définir son statut de présence pour la messagerie instantanée, d'accéder aux préférences de GNOME (voir la question « Comment personnaliser mon environnement ? »), de fermer sa session ou mettre en veille prolongée sa machine (notez que pour l'éteindre, une pression sur la touche **Alt** fera apparaître une entrée à cette fin dans ce menu).

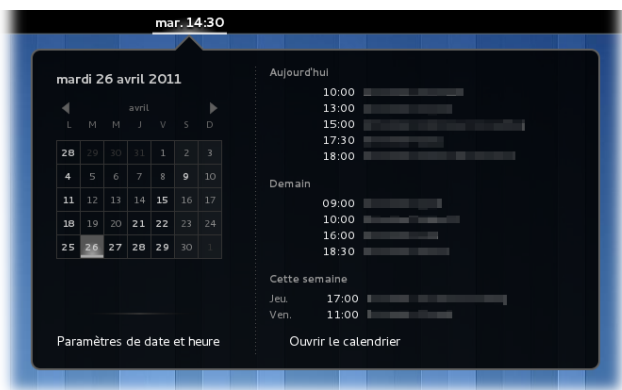


FIG. 7 L'applet « calendrier » de GNOME Shell



FIG. 8 La zone de statut et le menu utilisateur de GNOME Shell

Les activités

GNOME 3 implémente, au travers de GNOME Shell, le concept d'*activités* : il s'agit ici non plus de considérer l'utilisation du bureau en fonction des applications lancées ou des documents ouverts mais selon ce qu'on souhaite *faire*, en regroupant les outils à cette fin par activités. Vincent UNTZ, une des figures du projet GNOME, résume d'ailleurs ainsi ce point de vue :

« Ce que veulent les utilisateurs, ce n'est pas « utiliser » un environnement de bureau ! Ce qu'ils veulent, c'est surfer sur le Web, lire leurs courriels, discuter en ligne, écouter de la musique, rédiger leur correspondance, créer du contenu, etc. Voilà ce qui doit nous guider ! Nous voulons rendre GNOME moins envahissant, plus intuitif, plus cohérent, ceci afin de faciliter la vie de nos utilisateurs. »

Ce concept est donc visible dans GNOME Shell au travers d'une vue dédiée, accessible à la souris soit en pressant le bouton *Activités* en haut à gauche de l'écran, soit en déplaçant simplement le curseur dans ce coin, ou encore au clavier avec le raccourci clavier **Alt** **F1** ou la touche **W**. Cette vue, présentée en figure 6 page suivante, se décompose en quatre parties.

Le dash Le *dash* d'applications, à la gauche de la vue, joue un double rôle en permettant à la fois :

- de lancer ses applications favorites, faisant ainsi office de barre de lanceurs ;
- d'accéder aux applications déjà ouvertes, se substituant à la liste des fenêtres que l'on retrouvait dans GNOME 2 ; les applications déjà en cours d'exécution sont entourées d'un halo lumineux pour les distinguer des lanceurs.

On notera d'ailleurs la similitude de ce *dash* en termes d'utilisation avec la barre de tâches de Windows 7.

La vue des fenêtres La vue des activités s'ouvre par défaut sur celles des fenêtres (voir figure 6 page suivante) : toutes les fenêtres ouvertes sont organisées en mosaïque de miniatures, visibles d'un simple coup d'œil. Il est également possible depuis cette même vue de fermer les fenêtres en survolant à la souris celle à clôturer et en cliquant sur la croix apparaissant dans le coin droit de la miniature associée.

C'est par le biais des bureaux virtuels, accessibles aussi depuis cette vue, que le concept d'activité prend également son sens : on trouve en effet sur la droite un volet surgissant au passage de la souris dessus, listant le bureau courant ainsi que ceux créés par l'utilisateur. D'un simple glisser-déposer de la miniature d'une fenêtre ouverte vers ce volet, on crée ainsi un nouveau bureau dynamiquement. Chaque bureau peut donc être dédié à une activité : un pour la navigation Internet par exemple, un autre pour la correspondance, ou encore le développement, etc. Pour naviguer entre les bureaux au clavier, vous pourrez utiliser le raccourci **Ctrl** **Alt** **↓** pour accéder au bureau suivant, ou **Ctrl** **Alt** **↑** pour le précédent.

La vue des applications Cette vue rassemble toutes les applications graphiques, identifiées par leurs icônes (voir figure 9) ; comme dans le *dash*, celles en cours d'exécution sont auréolées d'un halo. Un menu à droite permet de filtrer les applications par thèmes, les mêmes que l'on retrouvait dans le menu applicatif de GNOME 2. L'ajout d'applications favorites dans le *dash* se fait par ailleurs depuis cette vue, d'un simple glisser-déposer d'une icône d'application vers le *dash*.



FIG. 9 La vue des applications

La recherche d'activités Dernier élément constitutif de la vue des activités, un champ, en haut à droite de celle-ci, autorise la recherche de ses activités parmi les applications, les raccourcis et les fichiers ouverts récemment. Au fur et à mesure de la saisie des termes de recherche, la vue se met à jour en fonction des éléments trouvés (voir figure 10 page suivante). Notons les deux boutons au bas de la vue permettant aussi via ce champ de lancer une requête sur Google ou Wikipedia, le résultat de la recherche s'ouvrant dans le navigateur.

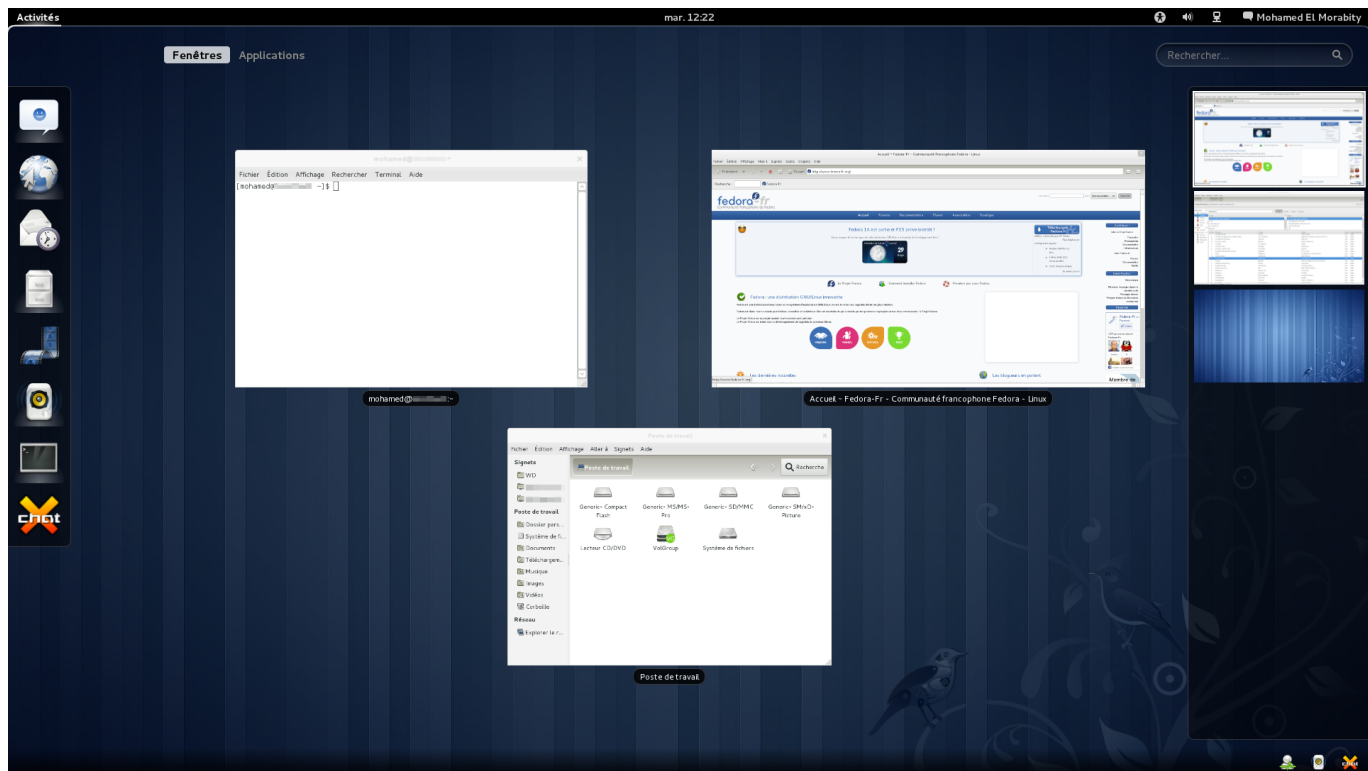


Fig. 6 GNOME Shell, au cœur de GNOME 3

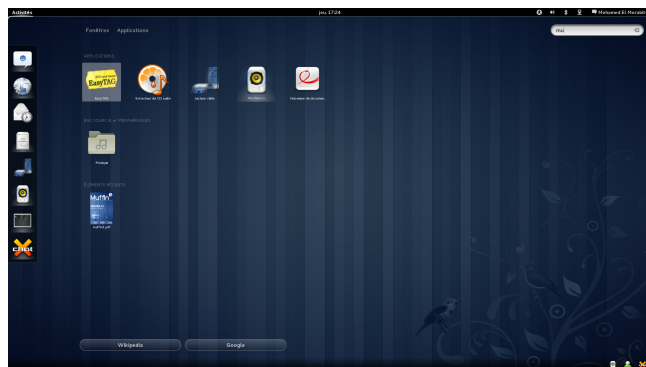


Fig. 10 La recherche d'activité en pleine action

Les notifications

Le système de notifications de GNOME Shell a été conçu de telle façon à ce que celles-ci perturbent le moins possible l'utilisateur dans son activité, tout en étant suffisamment visibles et accessibles. Ainsi toute notification apparaîtra subtilement en bas au centre de l'écran — là où l'utilisateur risquera le moins de cliquer dessus par accident, là où il ne risquera pas non plus, en voulant la sélectionner, d'interagir involontairement avec un élément du bureau —, son contenu étant dévoilé d'un survol dessus à la souris (voir figure 11).

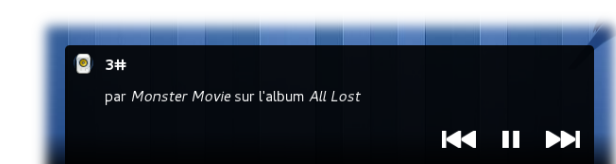


Fig. 11 Une notification « surgissante » de Rhythmbox

Si elle n'est pas traitée avant qu'elle s'éclipse, elle n'est pas perdue pour autant puisqu'elle est conservée (selon un mécanisme de pile) en bas à droite de l'écran, dans une zone dédiée (voir figure 12) surgissant au passage de la souris ; on retrouvera d'ailleurs à cet emplacement les icônes d'état d'applications qui se font finalement plus discrètes au profit des notifications, toujours dans l'optique de laisser l'utilisateur se concentrer sur ses tâches.

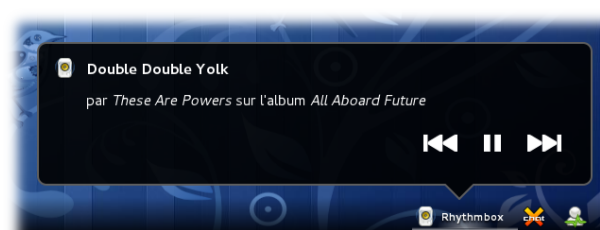


Fig. 12 La zone de notifications de GNOME Shell

Mais l'intérêt de ces nouvelles notifications réside surtout dans leur intégration de la messagerie instantanée : il est en effet possible, à la notification d'un nouveau message d'un contact dans Empathy, de répondre *directement* dans celle-ci (voir figure 13), et ce sans perdre de vue encore une fois sa fenêtre en cours.

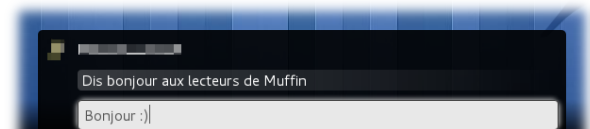


Fig. 13 Une notification de nouveau message

La gestion des fenêtres

En tant que gestionnaire de fenêtres, GNOME Shell propose quelques nouveautés en matière d'interaction avec les fenêtres ouvertes.

Si le sélecteur d'applications ouvertes est toujours accessible via le raccourci **Alt + F3**, celui-ci regroupe désormais les fenêtres par application (voir figure 14 page suivante) ; les fenêtres en question sont alors identifiables grâce à une sous-vue en miniatures et pourront être sélectionnées à la souris ou par le raccourci **Alt + F4**. On notera au passage la séparation entre les fenêtres du bureau courant et celle des autres bureaux par un filet discret dans ce sélecteur.



FIG. 14 Le sélecteur d'applications de GNOME Shell

GNOME Shell apporte également quelques fonctions pratiques inspirées des gestionnaires de fenêtres pavants pour redimensionner les fenêtres : en déplaçant une fenêtre vers le bord gauche de l'écran, et une autre vers le bord droit, les deux fenêtres occuperont chacune une moitié de l'écran (voir figure 15), ce qui peut s'avérer utile pour des comparaisons. De même, déplacer une fenêtre vers le bord haut de l'écran (ou double-cliquer simplement sur sa barre de titre) lui fera occuper tout l'écran.

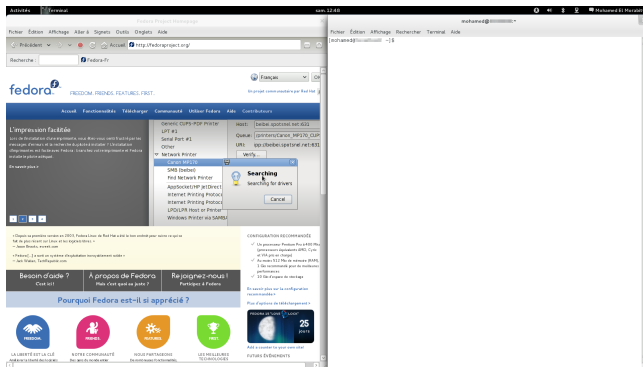


FIG. 15 Epiphany et GNOME Terminal placés en mode « comparaison »

Enfin, une nouveauté qui paraîtra sans doute déroutante aux yeux de certains, la disparition sur les barres de fenêtres des boutons de réduction et de maximisation : ce choix est justifié⁶ par le design de GNOME Shell puisque :

- il n'y a simplement plus de liste de fenêtres directement accessible dans laquelle réduire les fenêtres ;
- la vue des activités répond amplement au besoin d'organiser ses fenêtres, là où effectivement la réduction des fenêtres y suppléait dans GNOME 2 ;
- le redimensionnement des fenêtres par déplacements sur les bords rend caduc un bouton de maximisation.

Ma machine pourra-t-elle exécuter GNOME 3 ?

Ainsi qu'on l'a vu précédemment, GNOME Shell tire profit, à travers Mutter, de l'accélération graphique matérielle pour ses différents effets de transparence et de transitions. Par conséquent, une carte graphique relativement récente (datant de quatre à cinq ans au plus selon les recommandations du projet GNOME) est nécessaire, ainsi qu'un pilote pour celle-ci prenant en charge ses capacités 3D, pour exécuter GNOME Shell.

Les pilotes libres intel, radeon et nouveau devraient, sans aucune configuration particulière, prendre en charge respectivement la plupart des cartes graphiques Intel, AMD et Nvidia du marché sous Fedora 15. Cependant ce support libre de la 3D risque d'être partiel voire nul, avec certains des modèles les plus récents des deux dernières marques ; dans ce dernier cas, l'installation des pilotes propriétaires constructeurs s'avérera nécessaire. On pourra s'assurer, avant passage à GNOME 3, de la prise en charge de la 3D sur son système en lançant simplement la commande suivante :

```
$ glxinfo | grep "direct rendering"
```

Si l'accélération graphique est bien prise en charge par le système, la commande devrait retourner le message suivant :

```
direct rendering: Yes
```

Dans le cas contraire, l'installation des pilotes constructeurs, pour les cartes graphiques AMD⁷ et Nvidia⁸, s'imposera sans doute ; ces derniers seront disponibles dans le dépôt logiciel RPM Fusion⁹ à l'occasion de la sortie de Fedora 15.

Néanmoins, ceux qui malheureusement ne disposeraient pas du matériel ou des pilotes adéquats pour exécuter GNOME Shell pourront tout de même utiliser GNOME 3, mais en mode *restreint* (voir figure 16). Ce mode est un environnement somme toute assez similaire à GNOME 2, les tableaux de bords GNOME Panel et le gestionnaire de fenêtres Metacity prenant la place de GNOME Shell, avec quelques limitations cependant (voir la question « Pourrai-je retrouver l'ancienne interface de GNOME ? »).

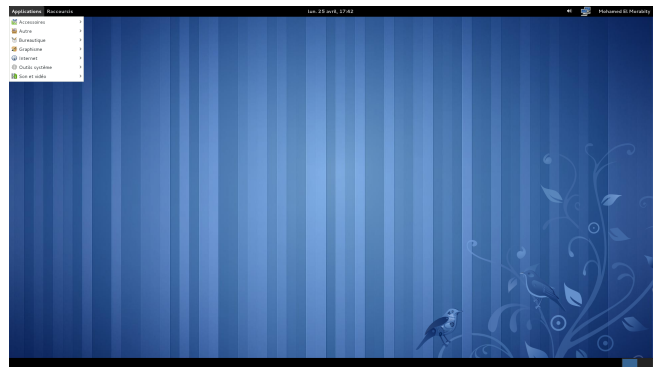


FIG. 16 GNOME 3 en mode restreint

Retrouverai-je mes applications favorites ?

GNOME ne se résume pas seulement à un environnement de bureau, mais englobe également toute une suite applicative dont les composants les plus représentatifs sont, entre autres, l'explorateur de fichiers Nautilus, le client de messagerie instantanée Empathy ou encore le lecteur de vidéos Totem. Toutes ces applications sont évidemment toujours disponibles dans GNOME 3 mais utilisent désormais la nouvelle version 3.0 de la bibliothèque graphique GTK+ ; en plus de bénéficier d'une interface de programmation consolidée pour les développeurs, cette version propose un système de thèmes basé sur la technologie CSS. Cependant, ces applications GNOME ne se distinguent pas seulement de leurs versions précédentes par le passage à GTK+ 3, mais également par quelques nouvelles fonctionnalités exclusives à GNOME 3, parmi d'autres¹⁰ :

Nautilus l'explorateur de fichiers de GNOME bénéficie d'une refonte de son interface (voir figure 17 page suivante) : le panneau latéral dédié aux raccourcis occupe désormais une place prépondérante dans l'application, la barre d'état disparaît (au profit d'une bulle d'information apparaissant à la sélection d'un élément dans le coin droit en bas), et les barres d'outils et d'adresse sont fusionnées ; au passage, le raccourci clavier pour supprimer un fichier est désormais **Ctrl** **Suppr** et non plus simplement **Suppr**, sujet aux suppressions accidentelles ;

Gedit l'éditeur de texte se dote d'une nouvelle interface de recherche de texte qui n'interfère pas avec l'édition (voir figure 18 page suivante) et propose un mode de groupement des onglets permettant de visualiser plusieurs fichiers dans la même fenêtre ;

Epiphany le navigateur Web léger voit ses performances améliorées et supporte désormais la géolocalisation ; il inclut aussi un nouveau gestionnaire de téléchargement intégré à la fenêtre principale inspiré de celui de Google Chrome.

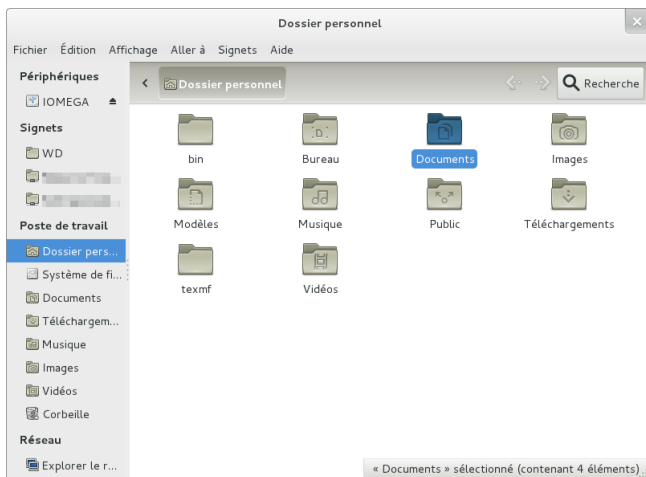


Fig. 17 Nautilus dans GNOME 3

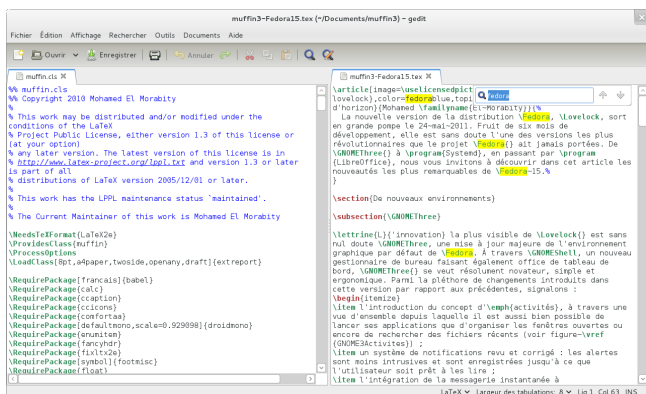


Fig. 18 Gedit, le groupement d'onglets et la nouvelle interface de recherche de texte

Le fait que GNOME 3 repose désormais sur GTK+ 3 n'empêchera évidemment nullement les applications basées sur d'autres bibliothèques graphiques de s'exécuter dans l'environnement, qu'elles reposent encore sur GTK+ 2 (Firefox par exemple, Pidgin ou encore XChat), ou bien sur Qt (les applications KDE, VirtualBox ou Skype, entre autres), tout comme GNOME 2 l'autorisait. Seul bémol, les différents *applets* que l'on pouvait ajouter sur le tableau de bord de GNOME 2 — GNOME Applet Sensors en est un exemple typique — ne seront ni utilisables, ni tout simplement compatibles avec GNOME 3, même en mode restreint. Ces *applets* devraient d'ailleurs être retirés des dépôts Fedora. Il est cependant possible qu'à l'avenir un système similaire soit mis en place dans GNOME Shell^[1], en conformité avec les impératifs d'ergonomie définis pour l'environnement.

Qu'en est-il de l'accessibilité ?

Un des objectifs du projet GNOME est de proposer un bureau utilisable par *tous*, et ce sans contraintes. Si l'ergonomie a été un leitmotiv dans la conception de GNOME Shell, l'accessibilité a aussi été au cœur des préoccupations de ses développeurs. GNOME Shell intègre ainsi par défaut un *applet* dédié dans la zone de statut (voir figure 19), proposant d'un clic un menu d'activation de diverses fonctionnalités d'accessibilité telles que la loupe d'écran, l'utilisation d'un thème GTK+ à meilleur contraste pour les applications, les alertes visuelles ou encore le contrôle du pointeur de la souris par le clavier, ainsi que l'accès au panneau de configuration associé.



Fig. 19 L'applet « accessibilité »

Ce panneau de configuration a d'ailleurs été complètement revu et corrigé dans le cadre de son intégration dans GNOME 3 : là où celui de GNOME 2 se contentait seulement de proposer des pointeurs vers les configurations du logiciel Orca, du clavier et de la souris, ce nouveau panneau rassemble dans une unique interface tous les réglages d'accessibilité, groupés par besoin. Signalons, pour les paramètres relatifs à la vision, le soin porté à leur lisibilité à l'intention des malvoyants par l'utilisation d'une police de texte plus grande (voir figure 20).

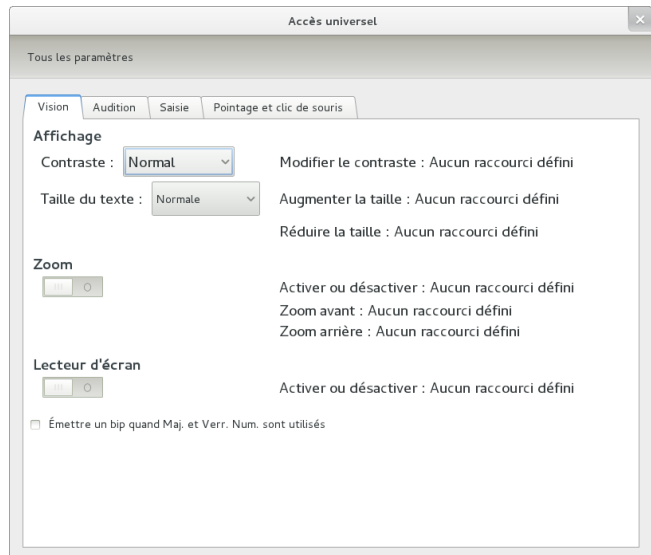


Fig. 20 Les paramètres d'accessibilité de GNOME

Si les versions précédentes de GNOME proposaient, *via* Orca, une loupe d'écran, GNOME Shell intègre lui sa propre loupe, plus réactive car tirant parti, comme l'environnement, de l'accélération graphique, et plus agréable à l'usage grâce à un lissage du zoom. Pouvant être démarrée depuis le menu de l'*applet* « accessibilité », elle autorise aussi ses propres raccourcis clavier pour son activation ainsi que pour l'augmentation ou la réduction du facteur de zoom, depuis l'outil de configuration du clavier (voir figure 21), dans le panneau de configuration de GNOME (voir la question « Comment personnaliser mon environnement ? »). Notez également la possibilité de limiter l'agrandissement au texte uniquement, toujours depuis le même menu.



Orca reste cependant toujours disponible, puisque c'est par son biais que GNOME supporte la lecture d'écran, à l'intention des non-voyants.

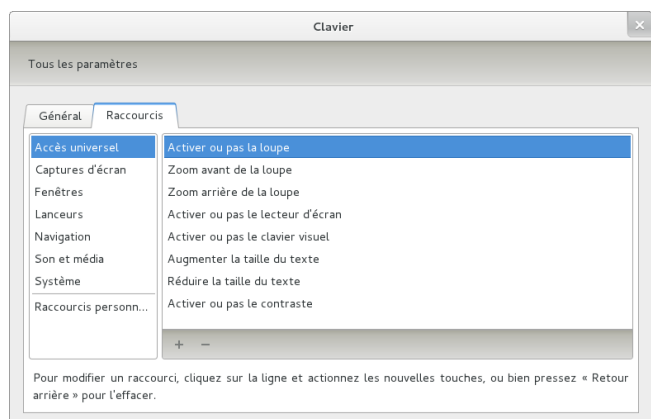


Fig. 21 La configuration des raccourcis de la loupe d'écran dans les paramètres clavier

Comment personnaliser mon environnement ?

GNOME s'est depuis toujours distingué des autres environnements par une volonté claire de ses contributeurs de *simplifier* la prise en main de l'environnement par tous les utilisateurs, tout spécialement les néophytes. À ce titre, les divers paramètres de configuration de GNOME ont été conçus de façon à être d'une part les plus *accessibles* possibles et

d'autre part *compréhensibles*, quitte à masquer les moins communs ou les moins évidents à saisir. GNOME 3 ne faillit pas à cette règle, comme nous allons le découvrir dans cette section.

Un panneau de configuration central

GNOME 3 propose ainsi un panneau de configuration dans lequel la plupart des outils de configuration hérités des versions précédentes (configuration de l'accessibilité, de l'affichage, du matériel...) s'y retrouvent centralisés et organisés par thèmes. Ce panneau, simplement accessible depuis le menu utilisateur sur le tableau de bord, répond ainsi à la fragmentation des différents réglages dans le menu de préférences de GNOME 2, facilitant du coup leur accès (voir figure 22).

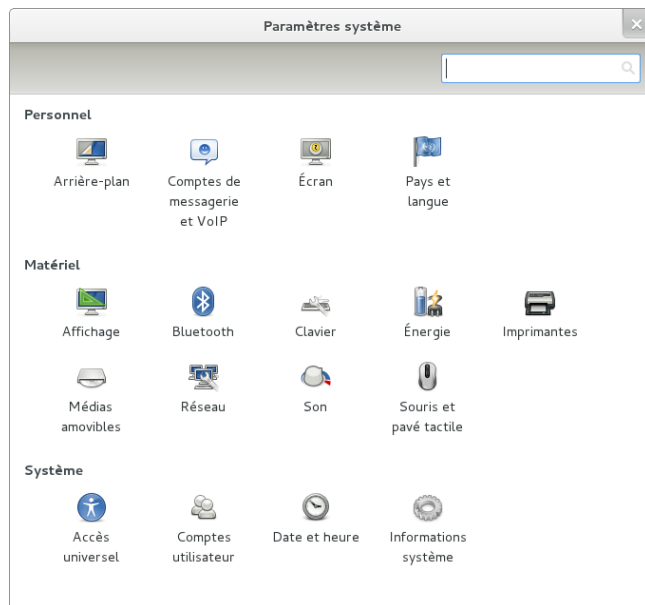


FIG. 22 Le panneau de configuration système de GNOME 3

GSettings

Les changements au niveau des préférences de GNOME 3 ne se limitent cependant pas à des changements « visibles », puisque c'est le cœur même du système de préférences, leur stockage en fait, qui a été complètement revu. GNOME 3 abandonne ainsi GConf pour GSettings, ainsi que les applications de l'écosystème GNOME. Ce changement vise avant tout à proposer une interface de gestion des préférences pour les développeurs plus robuste et ne dépendant plus des bibliothèques GNOME mais de la bibliothèque GLib — ce qui permettrait à d'autres applications, non liées particulièrement à GNOME, d'en profiter.

L'enregistrement des paramètres au travers de GSettings repose, comme pour GConf, sur un mécanisme de clés. Ces dernières pourront être lues et modifiées grâce à la commande `gsettings` dont on pourra consulter la page de manuel¹. Notez que, à l'instar de GConf avec `gconf-editor`, il existe une interface graphique pour parcourir et modifier les différentes clés de réglages disponibles, `dconf-editor` (voir figure 23), disponible dans le paquet du même nom.

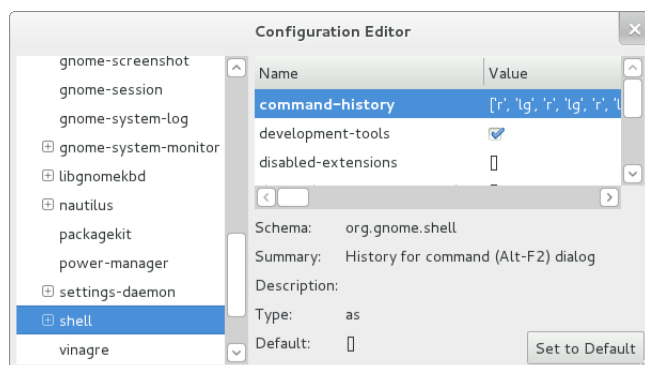


FIG. 23 `dconf-editor`

¹ De l'hindi अद्वैत, qu'on pourrait traduire par « le seul et l'unique ».

Modifier l'apparence de son environnement

Certains auront noté dans le panneau de configuration central de GNOME 3 l'absence de tout réglage relatif à l'apparence de l'environnement. Il s'agit d'un choix délibéré des contributeurs de GNOME (mais sujet à révision toutefois, à terme), qui considèrent que les différents éléments graphiques de GNOME 3, du thème GTK+ 3 Adwaita² à la police de texte Cantarell³, en passant par le noir du tableau de bord, font partie intégrante du design de l'environnement et contribuent à donner à GNOME une identité graphique forte. S'il lui apporte par ailleurs évidemment une touche de modernité, le look à la fois élégant et discret de GNOME 3 vise également à permettre à l'utilisateur de se concentrer sur ses tâches sans être en quelque sorte « distrait » par un environnement trop clinquant.

Il sera néanmoins possible de personnaliser l'apparence de son environnement, en dépit de ces restrictions somme toute légitimes. Si divers réglages « cachés » relatifs à l'apparence existent dans GNOME et peuvent être surchargés via `gsettings` ou `dconf-editor`, il existe heureusement un outil tiers, GNOME Tweak Tool, qui rend la personnalisation de GNOME 3 plus accessible. Par ailleurs, GNOME Shell intègre un mécanisme d'extensions étendant ses fonctionnalités.

GNOME Tweak Tool GNOME Tweak Tool⁴ est un outil graphique développé par John STOWERS proposant divers réglages de GNOME 3 non accessibles depuis le panneau de configuration, et tout particulièrement des réglages relatifs à l'apparence de l'environnement (voir figure 24). Ces différents réglages sont organisés par thèmes, chacun faisant l'objet d'un panneau dédié dans l'application, que voici :



Environnement on pourra depuis ce panneau afficher, en plus de l'heure, la date dans l'*applet* « calendrier », réactiver les boutons de réduction et agrandissement dans les barres de fenêtres ou gérer plus finement les actions à la fermeture de l'écran d'un portable ;

Polices ce thème rassemble tous les réglages relatifs aux polices utilisées dans les applications ;

Interface cet item propose une sélection de thèmes à appliquer aux applications GTK+ 3 : thèmes applicatifs, d'icônes, de curseurs de souris...

Navigateur de fichiers ce groupe propose une unique option, celle d'utiliser le bureau pour y placer des fichiers et des lanceurs comme dans les précédentes versions de GNOME, là où GNOME 3 laisse le bureau « nu » ;

Fenêtres ce dernier thème regroupe les réglages associés aux fenêtres : sélection du thème à appliquer aux barres de fenêtres et configuration des actions à la souris sur celles-ci.

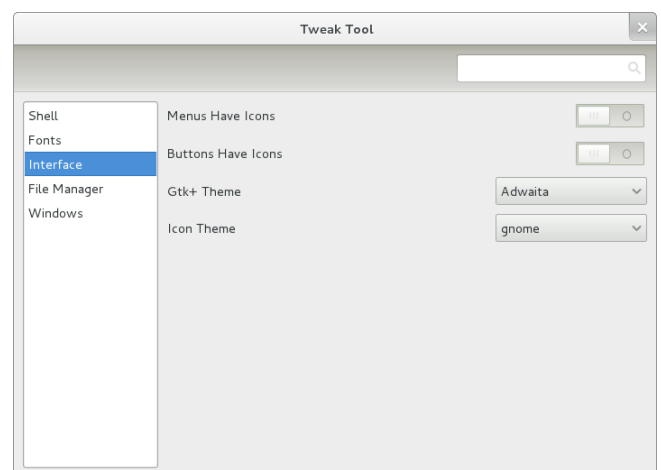


FIG. 24 GNOME Tweak Tool

Notez que les thèmes d'icônes jusqu'ici disponibles dans les dépôts (Elementary ou Faenza entre autres) ou sur GNOME-Look⁵ (que l'on pourra toujours installer dans le dossier `~/.icons/`) demeurent compatibles avec GNOME 3, de même que les thèmes de décorations de fenêtres (installés dans `~/.themes/`), Mutter pouvant utiliser sans problème les thèmes Metacity. En revanche, la plupart des thèmes d'applications ne sont utilisables qu'avec GTK+ 2, tandis que les thèmes

GTK+ 3 restent encore rares en comparaison. Ce manque devrait cependant être comblé dans le futur avec l'adoption généralisée de GNOME 3 ; signalons déjà les thèmes GTK+ 3 proposés par les artistes du groupe GNOME Shell du site DeviantART⁴⁵. GNOME Tweak Tool est disponible dans Fedora 15 et pourra être installé ainsi en ligne de commande :

```
$ su -lc "yum install gnome-tweak-tool"
```

On pourra alors le lancer depuis la vue applicative de GNOME Shell (sous l'intitulé *Tweak advanced settings*) ainsi qu'en ligne de commande :

```
$ gnome-tweak-tool
```

Les extensions de GNOME Shell GNOME Shell a été conçu de telle façon à ce que ses fonctionnalités puissent être étendues au travers d'extensions écrites en Javascript. Les *GNOME Shell Extensions*⁴⁶ sont une compilation de telles extensions aux rôles divers et variés, preuve des possibilités infinies offertes par GNOME Shell en matière de personnalisation. Notez que la liste de ces extensions, décrites dans la suite, est loin d'être figée et pourra s'enrichir en fonction des développements.

alternate-tab Cette extension fait en sorte que le sélecteur d'applications ouvertes, accessible par le raccourci **Alt + Tab**, ne regroupe plus les fenêtres par application, tout en affichant pour chacune d'elle une miniature (voir la figure 25, à comparer à la figure 14 page 9).

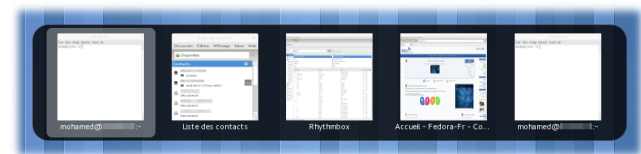


FIG. 25 Un sélecteur d'applications ouvertes alternatif

alternative-status-menu Cette extension modifie le menu utilisateur de la zone de statut en lui rajoutant deux nouvelles entrées *Hiberner* et *Éteindre...* (voir la figure 26, à comparer à la figure 8 page 7).

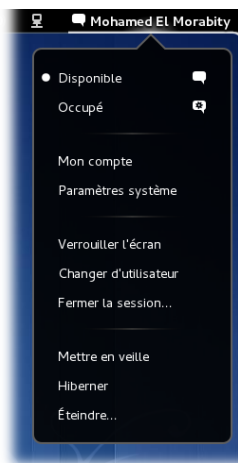


FIG. 26 Le menu utilisateur de la zone de statut, une fois l'extension *alternative-status-menu* installée

auto-move-windows Cette extension permet d'assigner une application graphique à un bureau virtuel particulier à chaque lancement. Sa configuration se fera au travers de *gsettings* : par exemple, pour qu'Epiphany s'ouvre toujours dans le bureau numéro 2 et Gedit dans le bureau numéro 4, on lancera la commande suivante :

```
$ gsettings set org.gnome.shell.extensions.auto-move-windows application-list "[ 'epiphany.desktop:2', 'gedit.desktop:4' ]"
```

On notera d'une part que les applications sont désignées dans cette commande par leur fichier *.desktop* (installés dans */usr/share/applications/*), et qu'autant de bureaux virtuels que nécessaires seront créés à la volée pour les y placer.

dock Cette extension active une barre des tâches, similaire au *dash* de la vue des activités, à la différence que celui-ci est à droite de l'écran et est *statique* (voir figure 27).

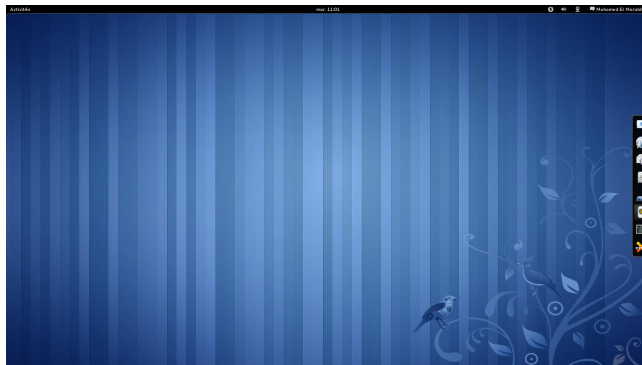


FIG. 27 Un dock permanent à droite du bureau grâce à l'extension du même nom

drive-menu Cette extension rajoute au tableau de bord un nouvel *ap-plet* permettant de ce fait d'accéder aux différents médias amovibles du système, ainsi que les monter ou de les démonter (voir figure 28).

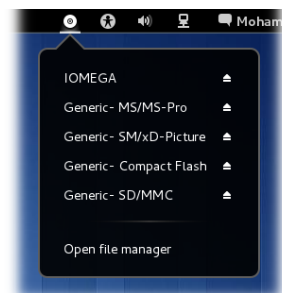


FIG. 28 L'applet « médias amovibles »

native-window-placement Cette extension modifie la façon dont les fenêtres sont placées dans la vue des activités : tandis que les fenêtres sont organisées par défaut selon une grille, cette extension s'affranchit de cette dernière et place les fenêtres en tenant compte de leur position sur le bureau et de leur taille (voir la figure 29, à comparer à la figure 6 page 8).

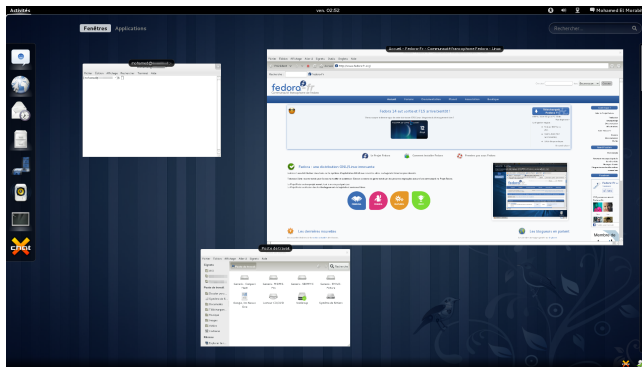


FIG. 29 L'extension *native-window-placement* réorganise les fenêtres de la vue des activités selon leur position et leur taille

places-menu Cette extension active un nouvel *ap-plet* dans le tableau de bord (voir figure 30 page ci-contre) donnant accès :

- aux emplacements définis dans les signets de Nautilus ;
- aux médias amovibles ;
- aux différents montages réseau configurés dans Nautilus également.

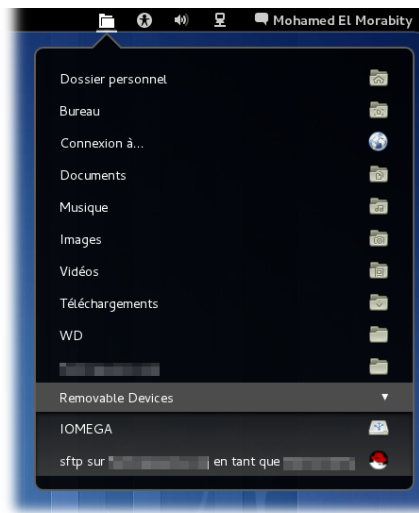


Fig. 30 L'extension places-menu et son menu de signets

user-theme Cette extension autorise la personnalisation de GNOME Shell par l'intermédiaire de thèmes, tels ceux disponibles sur la page dédiée à GNOME Shell sur DeviantART (voir figure 31 page suivante). Un thème GNOME Shell est constitué d'une feuille de style CSS (toujours nommée `gnome-shell.css`) et d'éventuels scripts Javascript et images ; tous ces fichiers devront être installés dans le dossier `~/.themes/<nom du thème>/` (ou bien dans le dossier `/usr/share/themes/<nom du thème>/gnome-shell/` pour rendre le thème disponible pour tous les utilisateurs du système). Pour activer alors le thème en question, on passera par `gsettings` ainsi :

```
$ gsettings set
org.gnome.shell.extensions.user-theme
name "<nom du thème>"
```

Le thème sera alors aussitôt chargé. Pour restaurer le thème par défaut de GNOME Shell, on lancera la commande suivante :

```
$ gsettings reset
org.gnome.shell.extensions.user-theme
name
```

windowsNavigator Cette dernière extension permet à l'utilisateur de naviguer au clavier dans la vue des activités : dans cette dernière, une pression sur la touche `[Alt]` assignera un numéro à chaque fenêtre de la mosaïque que l'on pourra sélectionner toujours en pressant cette même touche suivi de son numéro (voir figure 32). De même, on pourra sélectionner un bureau virtuel parmi ceux ouverts en pressant `[Ctrl]` suivi du numéro du bureau.

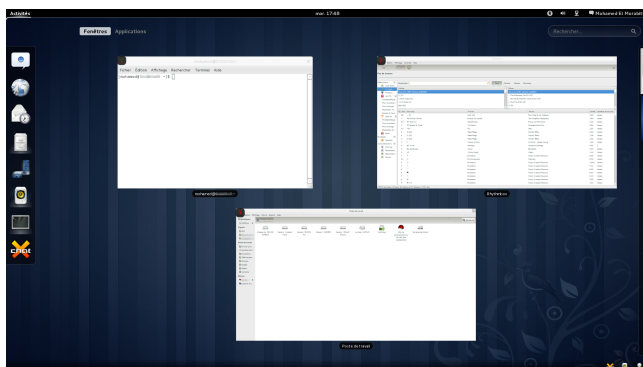


Fig. 32 L'extension windowsNavigator permet la navigation au clavier dans la vue des activités

Pour activer l'une de ces extensions, il suffira d'installer le paquet `gnome-shell-extensions-<extension>` correspondant ; par exemple, pour l'extension dock :

```
$ su -lc "yum install
gnome-shell-extensions-dock"
```

Une fois l'extension installée, GNOME Shell devra être relancé pour que celle-ci soit immédiatement chargée : il suffira pour cela de taper `[Alt] [F2]`, de saisir « r » dans la boîte de saisie de commande qui apparaît, puis de valider.

Notez que d'autres extensions, indépendantes des GNOME Shell Extensions, sont disponibles, telles celles développées par Finnbar P. MURPHY¹⁸. Une extension pour GNOME Shell est au moins composée d'un fichier `extension.js` et d'un fichier `metadata.json`. Ces fichiers, en plus d'éventuels autres fichiers de style ou d'images, pourront être installés manuellement dans le dossier `~/.local/share/gnome-shell/extensions/<identifiant>/`, où `<identifiant>` correspond à la valeur du champ `uuid` dans le fichier `metadata.json` de l'extension. Là encore, l'activation du thème nécessitera un redémarrage de GNOME Shell. De telles extensions « tierces » pourraient cependant être amenées à intégrer les dépôts logiciels de Fedora, telle déjà d'ailleurs l'extension `fedora-logo`, dans le paquet `gnome-shell-extension-fedora-logo`, qui rajoute le logo de Fedora au bouton Activités (voir figure 33).

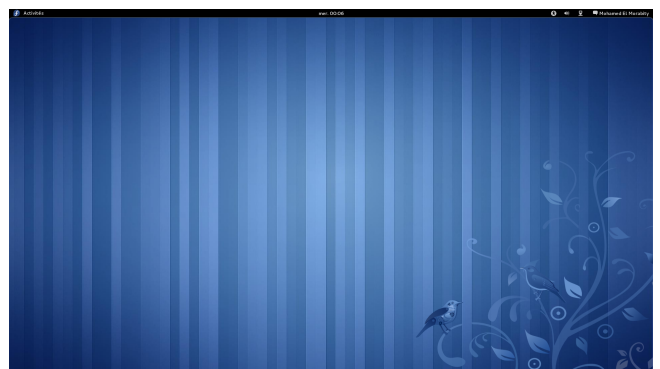


Fig. 33 GNOME Shell et l'extension fedora-logo

Pourrai-je retrouver l'ancienne interface de GNOME ?

Si, malgré la lecture de cet article, GNOME Shell ne devait pas convaincre, le retour à GNOME 2 risque d'être ardu, pour ne pas dire impossible, sous Fedora 15. Le projet GNOME a en effet décidé de ne pas maintenir la branche 2 de l'environnement pour mieux se concentrer sur les prochaines versions de GNOME 3. Si certaines distributions Linux projettent néanmoins de continuer à maintenir et proposer à l'installation la dernière version 2.32 de GNOME 2, il n'est cependant pas dans les intentions du projet Fedora de la fournir en parallèle de GNOME 3 dans Fedora 15, conformément à sa politique d'innovation constante. En revanche, GNOME 2 continuera à être maintenu dans Fedora 14, mais utiliser cette version n'accorde qu'un sursis de six mois, durée de sa maintenance à compter de la sortie de Lovelock, avant la disparition complète de GNOME 2 des versions maintenues de Fedora[†].

Il reste néanmoins le mode restreint de GNOME, déjà décrit plus haut (voir la question « Ma machine pourra-t-elle exécuter GNOME 3 ? » page 9). Sans reprendre fidèlement l'interface de GNOME 2, ce mode s'en rapprochera suffisamment pour ravir peut-être certains nostalgiques des versions précédentes (voir figure 16 page 9). Si ce mode est automatiquement utilisé en lieu et place de GNOME Shell sur les machines aux ressources insuffisantes pour exécuter ce dernier, on pourra aussi forcer son utilisation depuis le panneau de configuration, dans les Informations système : l'activation du mode restreint est alors proposée dans la rubrique Carte graphique

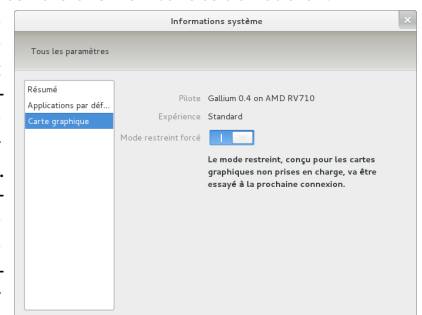


Fig. 34 Activation du mode restreint de GNOME 3 depuis le centre de contrôle

[†] À titre d'anecdote, signalons qu'un fork de GNOME 2, EXDE (<http://www.exde.org/>) a été envisagé ; cependant son initiateur y a finalement renoncé, pour des raisons inconnues.

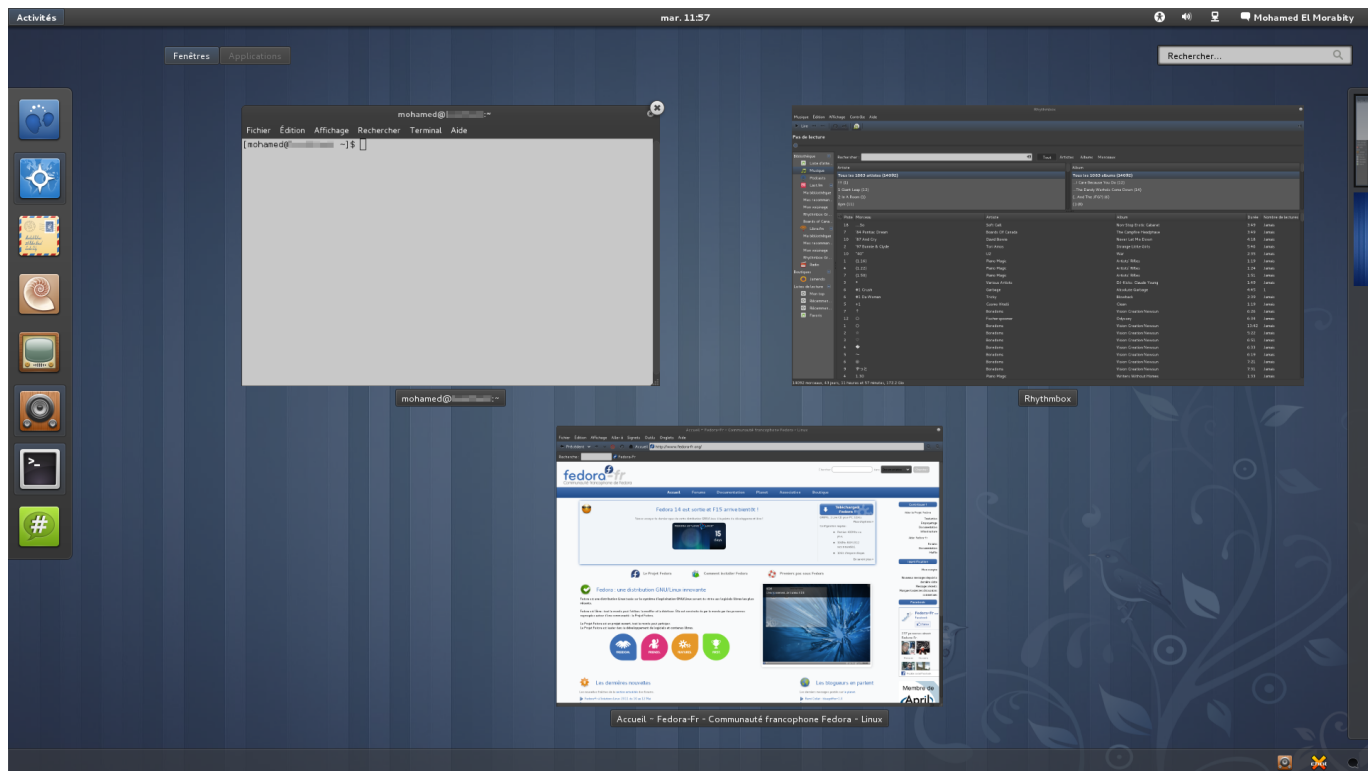


FIG. 31 Le thème Atolm appliqué à GNOME Shell (avec le thème GTK+ 3 du même nom et le thème d'icônes Faenza)

(voir figure 34 page précédente). Ce mode demeure néanmoins limité en termes de personnalisation par rapport à GNOME 2, même si GNOME Tweak Tool pourra suppléer en partie à ce manque ; en particulier, les *applets* hérités de GNOME 2 que l'on pouvait ajouter sur son tableau de bord n'en demeureront pas moins inutilisables en mode restreint. Le projet GNOME n'est pas disposé à faire de ce mode un environnement à part entière qui bénéficierait de la même attention que GNOME Shell, dans la mesure où ses contributeurs souhaitent ancrer GNOME 3 dans son temps avec ce dernier et que le mode restreint est avant tout une solution de secours pour les machines qui ne satisferaient pas aux standards actuels du marché.

Si finalement vous ne vous reconnaissiez vraiment pas en GNOME 3, il ne resterait plus qu'à changer d'environnement de bureau. Xfce serait un candidat idéal du fait de sa grande proximité, aussi bien logicielle qu'ergonomique ou esthétique, avec GNOME 2 ; par ailleurs, ses possibilités de configuration et ses nombreuses extensions sont telles qu'en finalement peu de réglages, on peut retrouver dans Xfce un environnement très proche de GNOME 2 (voir figure 35).

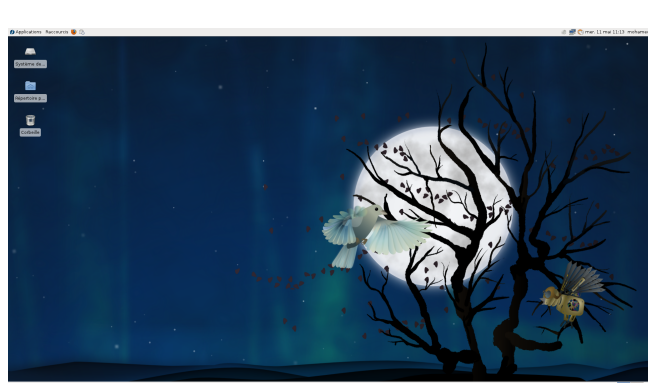


FIG. 35 Une session Xfce, configurée de façon à imiter GNOME 2

Pour conclure, même si GNOME 3 ne plaira pas forcément à tous, force est d'admettre la révolution apportée par cette nouvelle version dans la façon même d'appréhender son système. En remettant en question le concept de bureau, les concepteurs de GNOME Shell s'éloignent certes du chemin tracé par les premières interfaces graphiques des années 80, et suivi jusqu'ici sans guère de détours par la plupart des environne-

ments graphiques grand public. Mais être en phase avec l'*homo informaticus* du *xxi*^e siècle et ses besoins valait bien au projet GNOME de sortir des chemins battus...

Mohamed EL MORABITY



- [1] <http://www.vuntz.net/journal/post/2008/10/22/494-desktop-shell-from-the-user-experience-hackfest-general-overview>
- [2] <http://gnomejournal.org/article/117/how-we-got-here-part-i-of-a-design-history-of-gnome-3-the-shell>
- [3] <http://fr.wikipedia.org/wiki/Metacity>
- [4] <http://www.clutter-project.org/>
- [5] <http://www.h-online.com/open/features/The-path-to-GNOME-3-0-780071.html>
- [6] <http://mail.gnome.org/archives/gnome-shell-list/2011-February/msg00192.html>
- [7] http://doc.fedora-fr.org/wiki/Carte_graphique_AMD_Radeon:_installation_du_pilote_propri%C3%A9taire
- [8] http://doc.fedora-fr.org/wiki/Carte_graphique_NVIDIA:_installation_des_pilotes
- [9] <http://rpmfusion.org/>
- [10] <http://library.gnome.org/misc/release-notes/3.0/#rusers.applications>
- [11] <http://mail.gnome.org/archives/desktop-devel-list/2009-April/msg00267.html>
- [12] <http://manpages.ubuntu.com/manpages/natty/man1/gsettings.1.html>
- [13] <http://abattis.org/cantarell/>
- [14] <http://live.gnome.org/GnomeTweakTool>
- [15] <http://gnome-look.org/>
- [16] <http://gnome-shell.deviantart.com/>
- [17] <http://live.gnome.org/GnomeShell/Extensions>
- [18] <http://blog.fpmurphy.com/>

Les paquets Fedora du moment

L'équipe du blog « Les paquet Fedora du jour »¹ vous propose à chaque numéro une sélection de ses articles les plus appréciés dans cette rubrique. Ces articles regroupent des présentations de logiciels qui permettront à chacun de les installer et de les découvrir. Ces descriptions seront organisées selon cinq thèmes :

Lundi productif utilitaires et outils de productivité ;

Mardi artistique médias et outils de créativité, artwork et thèmes ;

Le « pourquoi ? » du mercredi un aperçu des détails de configuration et des choix de packaging faits par les développeurs de Fedora ;

L'interface du jeudi des outils graphiques qui valent le détour ;

Vendredi divertissement jeux, distractions et outils pour embellir votre environnement.

Lundi productif

GOURMET est un gestionnaire de recettes de cuisine. Il permet la création et l'édition de vos propres recettes ainsi que l'import d'autres recettes depuis une page Web. Vous pourrez avec noter vos recettes, les classer, faire des recherches également par ingrédient, catégorie (entrée, dessert...) ou type de cuisine (voir figure 36). Il sera également possible de créer une liste de courses à partir des recettes sélectionnées. Gourmet propose en outre un minuteur — histoire de ne pas laisser brûler vos plats — et un convertisseur d'unités. Il intègre par ailleurs un système d'extensions que l'on pourra activer, comme celle affichant les informations nutritionnelles des plats (qui malheureusement va chercher ses informations dans une base de données en anglais et nécessitera du coup de traduire ses ingrédients dans cette langue). Enfin, il est évidemment possible d'imprimer ses recettes mais également de les exporter, en HTML par exemple.

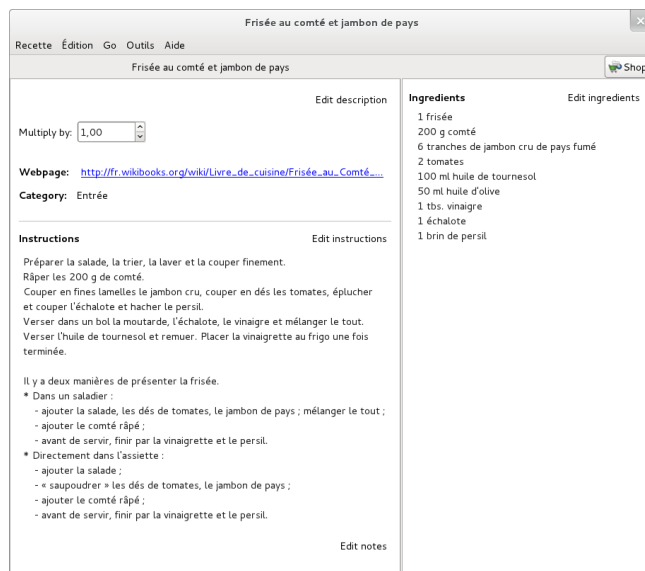


FIG. 36 Gourmet et la recette de la frisée au comté et jambon de pays

Installation en ligne de commande

```
$ su -lc "yum install gourmet"
```

Installation avec l'interface graphique **Bureautique** ➤ **Autres** ➤ **Gourmet Recipe Manager**

Localisation dans le menu **Applications** ➤ **Autres** ➤ **Gourmet Recipe Manager**

Lancement en ligne de commande

```
$ /usr/bin/gourmet
```

Site Internet <http://grecipe-manager.sourceforge.net/>

Mardi artistique

MyPaint est une application graphique simple et rapide pour réaliser des dessins sur votre distribution Fedora. Ce programme propose à cette fin un large choix de types de rendus (crayon, brosse, pinceau...) et de couleurs. Si MyPaint ne dispose pas d'une panoplie d'outils aussi complète que celle de Gimp, il permettra en revanche de réaliser rapidement et simplement de véritables chefs-d'œuvre (voir figure 37 page suivante) ; il s'agit donc avant tout d'un outil de dessin plutôt qu'un outil complet de manipulation graphique.

Les principales fonctionnalités de MyPaint sont les suivantes :

- ➔ support des tablettes graphiques (en particulier, les pressions sur le stylet sont restituées dans l'application) ;
- ➔ possibilité de définir ses propres effets de rendu ;
- ➔ canevas illimité (le dessin n'est pas limité par une taille prédéfinie) ;
- ➔ fonction d'annulation.

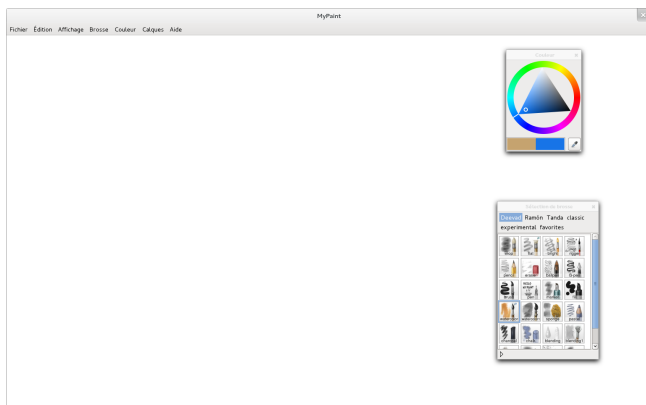


Fig. 37 MyPaint prêt à immortaliser un nouveau chef-d'œuvre

Installation en ligne de commande

```
$ su -lc "yum install mypaint"
```

Installation avec l'interface graphique Applications  Multimédia  MyPaint

Localisation dans le menu Applications  Graphisme  MyPaint

Lancement en ligne de commande

```
$ /usr/bin/mypaint
```

Site Internet <http://mypaint.intilinux.com/>

Le « pourquoi ? » du mercredi

Vous aurez sans doute noté, à la fin d'un rapport de bogue en cours de résolution sur Red Hat Bugzilla², une invitation à tester un correctif depuis le dépôt updates-testing et à laisser un *karma* sur une page dédiée³ (voir figure 38).

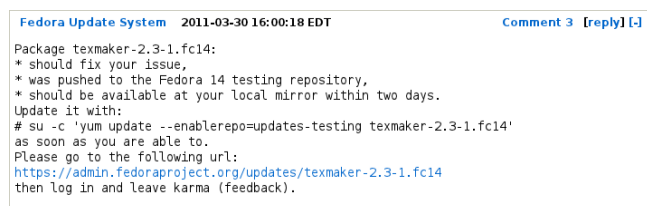


Fig. 38 Un message automatique à la fin d'un rapport de bogue résolu invitant à tester un correctif depuis le dépôt update-testing

Le karma d'un paquet est une note que quiconque enregistré sur le FAS (Fedora Account System) peut lui donner ; si par défaut un paquet placé dans le dépôt updates-testing intègre les dépôts stables au bout de trois semaines, un karma suffisamment grand peut accélérer cette intégration... ou au contraire l'empêcher si son karma est négatif. Ce système de notations offre ainsi un moyen facile de contribuer à la stabilité de la distribution Fedora.

Nous vous proposons un petit outil en ligne de commande, *fedora-easy-karma*, qui vous permettra de donner un karma simplement depuis votre terminal. Celui-ci prend en paramètre votre identifiant FAS, à créer le cas échéant⁴ :

```
$ fedora-easy-karma --fas-username=(identifiant)
```

La commande va rechercher parmi tous les paquets installés sur votre système ceux provenant du dépôt updates-testing. Elle va aussi afficher pour chacun d'entre eux le rapport de bogue pour lequel il a été généré ainsi que les karmas et commentaires qu'il a déjà reçus :

```
$ fedora-easy-karma --fas-username=(identifiant)
Getting list of installed packages...
Getting list of packages in updates-testing...
=====
texmaker-2.3-1.fc14
=====
Update ID: FEDORA-2011-4410
Release: Fedora 14
Status: testing
Type: bugfix
Karma: 1
Bugs: https://bugzilla.redhat.com/660215
      - Bold and Highlighted text are
        persistent
Notes: Update to version 2.3
Submitter: deji
Submitted: 2011-03-30 03:46:26
Comments: bodhi - 2011-03-30 03:46:42 (karma
           0)
           This update has been submitted for
           testing by deji.
           bodhi - 2011-03-30 20:00:19 (karma
           0)
           This update has been pushed to
           testing
           marbolangos - 2011-04-01 13:49:00
           (karma 1)
           Update works!
```

<https://admin.fedoraproject.org/updates/F14/FEDORA-2011-4410>

inst. RPMS: texmaker-2.3-1.fc14.x86_64 - LaTeX editor (installed 0 days ago)

Comment? -1/0/1 ->karma, other -> skip>

Vous êtes alors invité à donner un karma positif en saisissant 1 si le correctif vous satisfait ou bien un karma négatif (-1) si le correctif ne résout pas le bogue pour lequel il a été émis :

```
Comment? -1/0/1 ->karma, other -> skip> 1
```

Dans tous les cas, vous avez la possibilité de laisser un commentaire :

```
Comment? -1/0/1 ->karma, other -> skip> 1
Comment> Works fine
```

Votre contribution sera alors prise en compte, ce que vous pourrez vérifier en relançant la commande.

Installation en ligne de commande

```
$ su -lc "yum install fedora-easy-karma"
```

Installation avec l'interface graphique Développement  Fedora update feedback made easy

Localisation dans le menu Non disponible dans le menu

Lancement en ligne de commande

```
$ /usr/bin/fedora-easy-karma
```

Site web https://fedoraproject.org/wiki/Fedora_Easy_Karma/

L'interface du jeu

Déjà Dup est un outil de sauvegarde de vos données. Plus précisément, il s'agit d'une interface à la commande `duplicity`^[5] qui est un logiciel de sauvegarde basé sur `rsync`. Déjà Dup est particulièrement bien intégré à l'environnement de bureau GNOME et offre les fonctionnalités suivantes :

- outre la sauvegarde d'un ou plusieurs répertoires ou fichiers de son ordinateur, il autorise la définition de filtres d'exclusion ; les sauvegardes peuvent être envoyées vers un répertoire local, un répertoire distant (serveurs SSH, FTP ou HTTP) et même un serveur Amazon S3 (*cloud computing*) ;
- il est capable de chiffrer et compresser les données sauvegardées ;
- les sauvegardes qu'il réalise sont incrémentales ;
- il permet enfin la planification de sauvegardes automatiques.

La manipulation de ce logiciel s'avère par ailleurs particulièrement facile ; avec un outil aussi simple, vous n'aurez plus d'excuse pour ne pas faire de sauvegardes régulières !

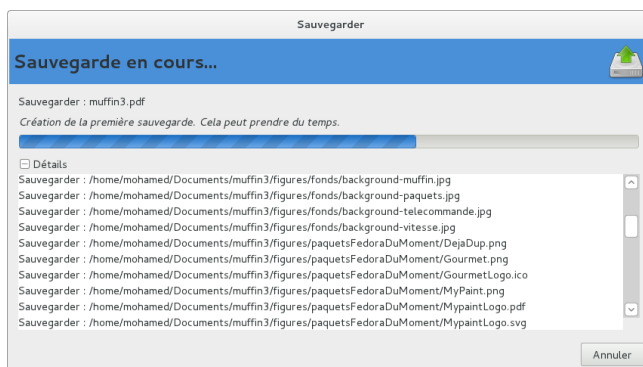


FIG. 39 Déjà Dup en action

Installation en ligne de commande

```
$ su -lc "yum install deja-dup"
```

Installation avec l'interface graphique Bureau GNOME ➤ Simple backup tool and frontend for duplicity

Localisation dans le menu Applications ➤ Outils Système ➤ Outil de sauvegarde Déjà Dup

Lancement en ligne de commande

```
$ /usr/bin/deja-dup
```

Site Internet <https://launchpad.net/deja-dup>

Vendredi divertissement

Teeworlds est un TPS (*Third person shooter*, ou jeu de tir objectif) en 2D multijoueur. Le principe du jeu est simple : vous guidez un Tee, une petite créature ronde tenant dans une main un grappin et dans l'autre une

arme qui peut être un maillet, un pistolet, un fusil à pompe ou encore un katana. Le personnage se balade dans les niveaux et doit tuer ses adversaires ; pour cela, il faut se déplacer très vite ! C'est là que le grappin se révèle utile, pour sauter d'îlot en îlot ou bien capturer ses adversaires... Teeworlds propose divers modes de jeu :

Death match ce mode consiste simplement à tuer le plus de Tees possible... et survivre ;

Team Death match ce mode est la déclinaison en équipe du précédent ;

Capture the flag le but de ce mode, en deux équipes, est de capturer le drapeau de l'équipe adverse et de le ramener dans son propre camp tout en évitant de perdre le sien.

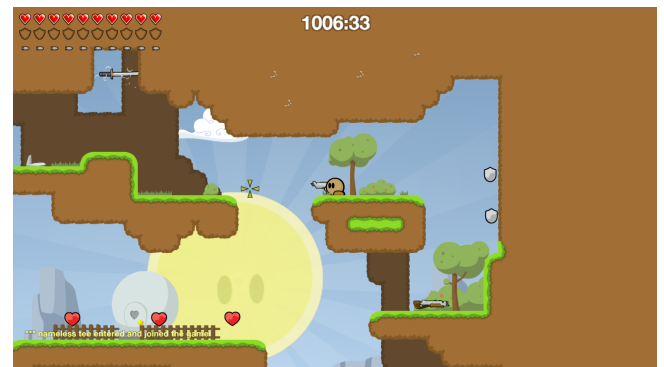


FIG. 40 Teeworlds en mode « Death match »

Installation en ligne de commandes

```
$ su -lc "yum install teeworlds"
```

Installation via l'installateur graphique Jeux ➤ Online multi-player plateforme 2D shooter

Localisation dans le menu après installation Applications ➤ Jeux ➤ Teeworlds

Lancement en ligne de commande

```
$ /usr/bin/teeworlds
```

Site Internet <http://www.teeworlds.com/>

Alexandre FRANDEMICHE

- [1] <http://blog.fedora-fr.org/paquet-fedora-du-jour>
- [2] <https://bugzilla.redhat.com/>
- [3] <https://admin.fedoraproject.org/updates/>
- [4] <https://admin.fedoraproject.org/accounts>
- [5] <http://duplicity.nongnu.org/>

Les versions personnalisées de Fedora

La version de Fedora par défaut, dite « GNOME », installe l'environnement graphique GNOME 3 et un ensemble d'applications de base parfaitement intégrées qui doivent répondre aux besoins de la plupart des utilisateurs, tels qu'une suite bureautique, un navigateur Internet et un lecteur multimédia, entre autres. Les utilisateurs ayant cependant une préférence pour un autre environnement graphique — KDE, LXDE ou encore Xfce — ou souhaitant bénéficier d'un ensemble d'applications spécifiques à un besoin particulier peuvent choisir une version personnalisée, ou spin, même s'il est tout à fait possible d'installer la compilation d'applications d'une version personnalisée sur n'importe quelle version de Fedora.

Choisir l'environnement graphique

PLUSIEURS environnements graphiques peuvent être installés sur le même ordinateur. C'est lors de la connexion que l'on peut choisir avec quel environnement démarrer. Après l'installation d'un nouvel environnement, vous devrez donc fermer votre session puis choisir l'environnement à démarrer lorsque vous vous reconnecterez depuis le gestionnaire de connexion. Vous pourrez d'ailleurs modifier ce dernier en suivant les instructions de notre encart dédié page 31. Si vous avez installé une version personnalisée qui modifie l'environnement graphique mais que vous souhaitez installer GNOME, saisissez dans un terminal la commande suivante :

```
$ su -lc "yum groupinstall 'GNOME Desktop Environment'"
```

Fedora KDE

KDE est un des environnements offrant le plus de possibilités de personnalisation. Cette version installe l'environnement KDE ainsi que les principales applications basées sur Qt (voir figure 41 page ci-contre). Pour essayer cet environnement, saisissez dans un terminal la commande suivante :

```
$ su -lc "yum groupinstall 'KDE Software Compilation'"
```

Vous pouvez également télécharger l'image de Fedora avec KDE préinstallé sur sa page dédiée ¹.

Fedora LXDE

LXDE est un environnement léger, rapide et performant. Il est conçu pour les configurations légères et adapté aux écrans de petite taille, ce qui en fait un environnement de choix pour les *netbooks* (voir figure 42). Pour essayer cet environnement, saisissez dans un terminal la commande suivante :

```
$ su -lc "yum groupinstall LXDE"
```

Vous pouvez également télécharger l'image de Fedora avec LXDE préinstallé sur sa page dédiée ².

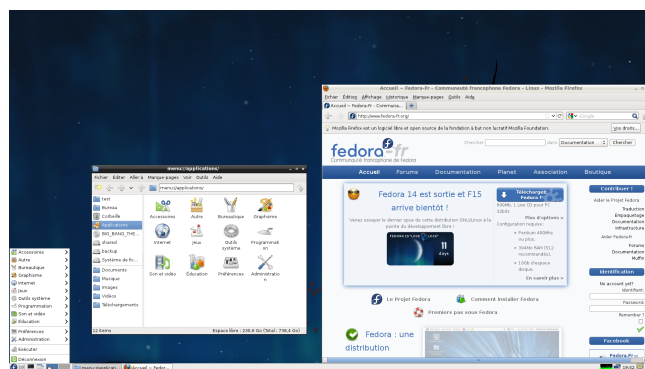


Fig. 42 Aperçu de LXDE

Fedora Xfce

Xfce est un environnement qui allie la simplicité d'utilisation à la légèreté, donc la rapidité. Il repose en outre sur la bibliothèque graphique GTK+ 2 et il suit les normes de freedesktop.org (voir figure 43).

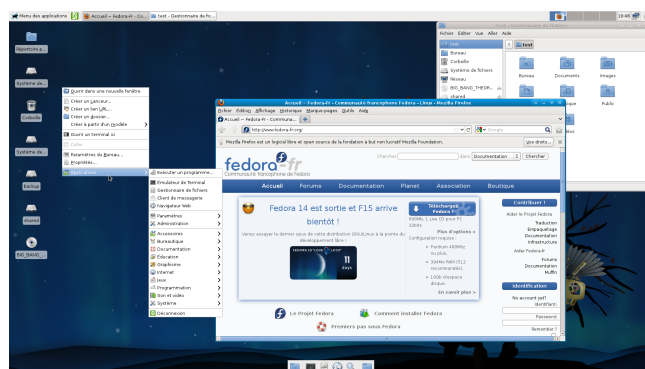


Fig. 43 Aperçu de Xfce

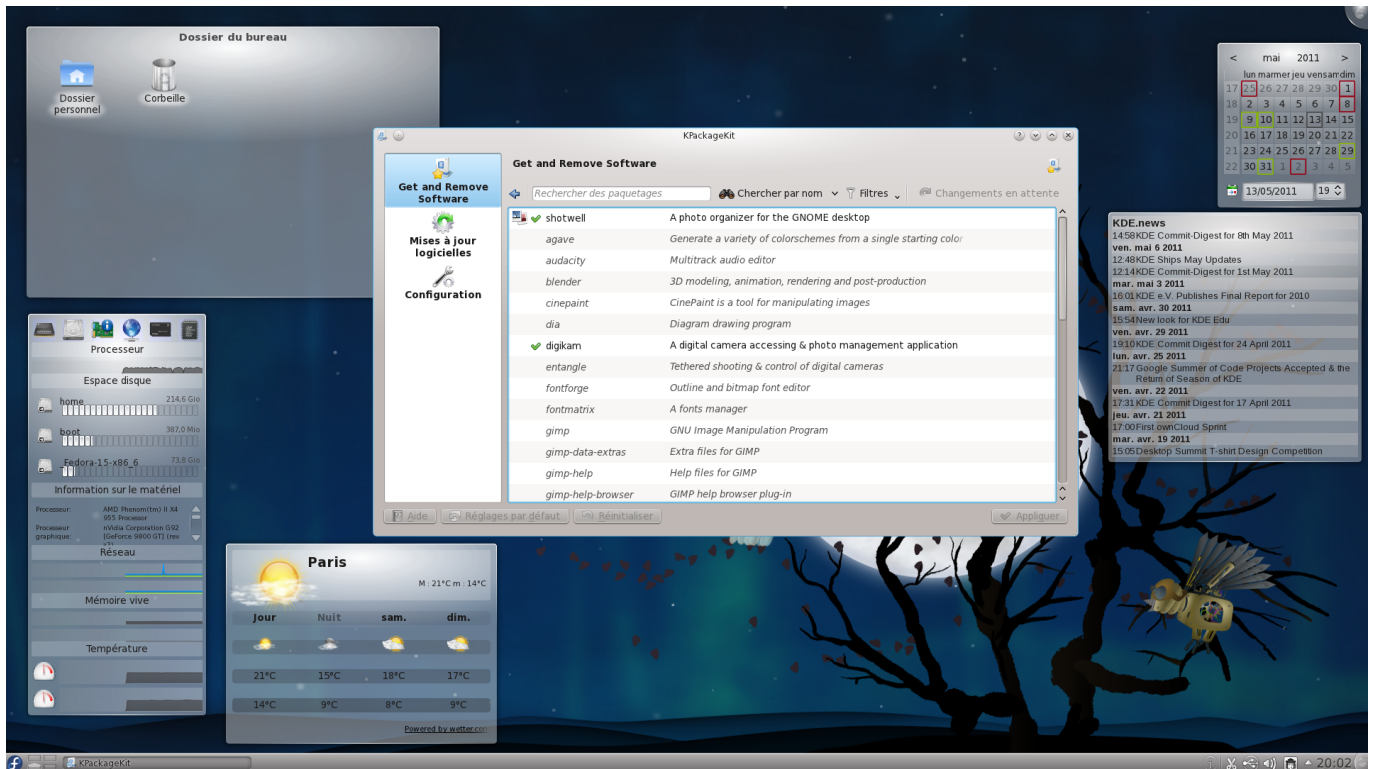


Fig. 41 Aperçu de KDE 4.6

Pour essayer cet environnement, saisissez dans un terminal la commande suivante :

```
$ su -lc "yum groupinstall XFCE"
```

Vous pouvez également télécharger l'image de Fedora avec Xfce préinstallé sur sa page dédiée³.

Fedora SoaS

Sugar est le système d'exploitation installé sur les XO, les ordinateurs du projet OLPC ayant pour but de promouvoir l'informatique en milieu scolaire dans les pays en voie de développement (déjà présenté dans notre numéro 2⁴). Le nom « SoaS » est un acronyme de *Sugar on a Stick* signifiant « Sugar sur clé » (dans le sens clé USB). Cette version personnalisée propose le même environnement graphique sur un ordinateur personnel, qui, par conception, est très différent d'un XO. Cette version personnalisée permet donc de tester des applications et d'en faire des démonstrations, voire tout simplement de bénéficier de cet environnement adapté à l'auto-apprentissage des jeunes enfants (voir figure 44).

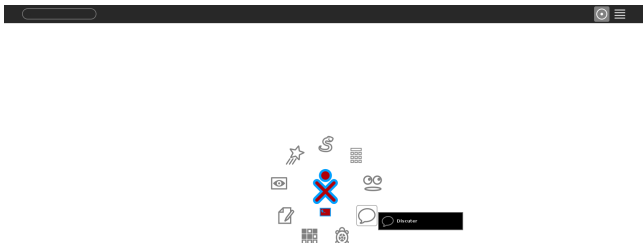


Fig. 44 L'écran d'accueil de Sugar

Pour essayer cet environnement, saisissez dans un terminal la commande suivante :

```
$ su -lc "yum groupinstall 'Sugar Desktop Environment'"
```

Vous pourrez aussi toutes les applications de cet environnement grâce à la commande suivante :

```
$ su -lc "yum install sugar*"
```

Notez que le thème Sugar pour Plymouth sera installé. Pour utiliser le thème de démarrage par défaut, utilisez la commande suivante :

```
$ su -lc "plymouth-set-default-theme charge"
```

Vous pouvez également télécharger l'image de Fedora avec Sugar préinstallé sur sa page dédiée⁵.

Choisir en fonction des programmes préinstallés

Certaines versions ont été conçues pour répondre à un besoin particulier. De ce fait, certains types d'outils sont préinstallés et configurés.

Fedora Security

La version personnalisée *Fedora Security* embarque tous les outils nécessaires à un audit de sécurité. Elle est basée sur LXDE. Il n'existe pour le moment pas de moyen rapide d'installation des outils sur un environnement Fedora existant. Il existe cependant un script permettant l'installation automatisée de toutes les applications du groupe « sécurité »⁶ ; si vous ne comprenez pas ce script, n'essayez pas de l'appliquer, passez plutôt à la solution suivante.

Si vous avez besoin de faire un audit de sécurité, la sauvegarde d'un système ou simplement pour tester les divers outils, la solution la plus simple est de télécharger la version *live* sur le site du projet⁷.

Fedora Electronic Lab

La version *Fedora Electronic Lab* est une plate-forme complète de CAO électronique libre conçue pour la conception, la simulation et la programmation électronique. Si c'est votre domaine, vous y trouverez probablement de nouveaux outils à essayer.

Pour essayer cet environnement, saisissez dans un terminal la commande suivante :

```
$ su -lc "yum groupinstall 'Electronic Lab'"
```

Vous pouvez également télécharger l'image de Fedora avec ce *spin* préinstallé sur sa page ⁸.

Fedora Design

La version personnalisée Fedora *Design* comprend toutes les applications dédiées à la conception graphique. Les applications préinstallées sont, entre autres, Agave, Blender, GIMP, Inkscape, Shotwell, Scribus, Xournal, Dia et bien d'autres.

Pour installer ces différentes applications, saisissez dans un terminal la commande suivante :

```
$ su -lc "yum groupinstall 'Design Suite'"
```

La version *live* de Fedora *Design* est disponible sur sa page ⁹.

Fedora Games

La version personnalisée Fedora *Games* met en avant une sélection de nombreux jeux libres : il n'y a pas plus facile pour découvrir de nouveaux jeux ! La liste complète des jeux pouvant être installés est disponible sur la page dédiée au *spin* ¹⁰, ainsi que l'image ISO associée.

Pour installer tous les jeux disponibles par défaut dans cette version, saisissez dans un terminal la commande suivante :

```
$ su -lc "yum groupinstall 'Games and Entertainment'"
```

Retrouvez toutes les versions personnalisées en détail sur la page du projet Fedora consacrée aux *spins* ¹¹.

Kévin RAYMOND



- [1] <http://spins.fedoraproject.org/fr/kde/>
- [2] <http://spins.fedoraproject.org/fr/lxde/>
- [3] <http://spins.fedoraproject.org/fr/xfce/>
- [4] http://mag.fedora-fr.org/w/images/3/3a/Muffin_num%C3%A9ro_2.pdf
- [5] <http://spins.fedoraproject.org/fr/soas/>
- [6] <http://falken.tuxfamily.org/?p=105>
- [7] <http://spins.fedoraproject.org/fr/security/>
- [8] <http://spins.fedoraproject.org/fr/fel/>
- [9] <http://spins.fedoraproject.org/fr/design/>
- [10] <http://spins.fedoraproject.org/fr/games/>
- [11] <http://spins.fedoraproject.org>

Changer le fond d'écran de GDM

La procédure qui suit vous permettra, sous Fedora 15, de modifier le fond d'écran du gestionnaire de connexion GDM.

On copiera dans un premier temps l'image à utiliser comme nouveau fond dans le dossier `/usr/share/backgrounds/gnome/` dédié :

```
$ su -
# cp (chemin vers l'image à utiliser)
  /usr/share/backgrounds/gnome/ \
  (nom de l'image)
```

La configuration de GDM est pilotée par un compte utilisateur dédié, `gdm`. Celui-ci n'étant pas accessible directement, on pas-

sera par le compte `root` pour s'y connecter :

```
# su - gdm -s /bin/bash
```

Une fois connecté, on récupérera les paramètres du démon D-Bus, nécessaire pour modifier les paramètres dans GSettings :

```
$ eval $(dbus-launch
--auto-syntax)
```

Ne reste plus qu'à utiliser la commande `gsettings` pour modifier proprement dit

le fond d'écran de GDM :

```
$ gsettings set
org.gnome.desktop.background
picture-uri
"file:///usr/share/backgrounds/\
gnome/(nom de l'image)"
$ exit
# exit
```

À la prochaine déconnexion de session, vous constaterez que votre nouveau fond d'écran a été pris en compte dans GDM. ●

Matthieu SAULNIER

Une introduction au langage D

Dans cet article, nous vous proposons une introduction à un nouveau langage, le langage D. Ce dernier, relativement récent puisque présenté pour la première fois en 1999, se présente comme un langage moderne, successeur des langages C et C++ ; il profite de leur héritage tout en introduisant de nouveaux concepts et paradigmes propres aux langages les plus récents. Cet article sera par ailleurs le premier d'une longue série consacrée à ce langage. Dans celui-ci nous nous attacherons à présenter le langage ainsi que sa syntaxe.

Introduction

Ce premier article va introduire les bases du langage D et les outils indispensables pour développer avec. D'autres articles, prévus pour les prochains numéros, présenteront plus en détail des notions de développement plus avancées, notamment la programmation objet ou encore les outils de documentation du code. Nous vous mettons en garde dès maintenant : cet article sera truffé d'exemples concrets mais n'abandonnera pas pour autant certains aspects théoriques. Cette approche résume à elle seule la philosophie du langage D : orienté pratique mais reposant sur des bases théoriques nécessaires. Le langage D est donc un langage de programmation système qui se présente comme le successeur des langages C et C++. Son objectif est d'allier la puissance de ces derniers à la simplicité et à la productivité de langages comme Ruby et Python. Son créateur, Walter BRIGHT¹, résume d'ailleurs cette philosophie en ces mots :

« Si un langage peut récupérer 90 % de la puissance du C++ mais seulement 10 % de sa complexité, cela vaut largement le coup. »

Les paradigmes supportés par ce langage sont les suivants :

- programmation impérative² ;
- programmation orientée objet³ ;
- méta-programmation⁴ ;
- programmation par contrat⁵.

Fedora et le D

Fedora propose depuis Fedora 14 tout un environnement de travail dernier cri pour développer en D, notamment un compilateur, ldc, basé sur LLVM⁶, ainsi qu'une bibliothèque standard dédiée, Tango. Fedora propose en outre :

- la bibliothèque Mango⁷, dédiée à la programmation réseau ;
- la bibliothèque Derelict, qui permet de développer des applications OpenGL en D.

À qui s'adresse ce langage ?

- Aux programmeurs lassés d'utiliser des analyseurs de code ou des outils similaires pour éliminer les bogues ;
- à tout ceux qui en ont assez des messages d'erreur incompréhensibles lors de la compilation de code C++ ;

- à tout ceux qui veulent faire de la programmation objet sans la complexité du C++ ;
- aux codeurs qui aiment la puissance du C++ mais qui sont frustrés par le temps de débogage nécessaire ;
- à ceux qui veulent intégrer les tests unitaires et activer des sections de code en mode débogage ;
- aux équipes qui écrivent efficacement des applications avec des millions de lignes de code ;
- aux développeurs qui pensent qu'un langage doit fournir suffisamment de fonctionnalités pour pallier la nécessité de manipuler des pointeurs ;
- aux programmeurs qui ont besoin de manipulations avancées sur les nombres ;
- aux équipes qui doivent écrire en partie leurs applications dans des langages interprétés comme Python et Ruby pour gagner du temps de développement.

À qui ne s'adresse pas ce langage ?

D ne s'adresse clairement pas aux puristes : il est avant tout un langage orienté pratique, non un langage théorique. Chaque fonctionnalité a ainsi été pensée dans cette optique : par exemple, D est construit et fournit une sémantique telle qu'elle élimine la nécessité d'utiliser des pointeurs dans la plupart des cas. Mais les pointeurs sont toujours là, parce que leur utilisation est parfois incontournable.

Ce dont vous avez besoin pour développer en D

Le socle minimal pour développer en D est constitué par le compilateur ldc et la bibliothèque standard du langage Tango. L'un et l'autre sont fournis respectivement par les paquets ldc et tango-devel, qu'on installera ainsi :

```
 $ su -lc "yum install ldc tango-devel"
```


Les éditeurs pour développer en D

Un compilateur ne suffit évidemment pas à développer, encore faut-il disposer d'un environnement de développement adéquat. Nous vous proposons dans la suite une sélection d'éditeurs prenant en charge le langage D.

Emacs Un mode D est disponible pour Emacs⁸. Une fois celui-ci déployé, on activera ce mode à l'ouverture d'un fichier de code D en tapant  puis en saisissant et en validant la chaîne d-mode.

Gedit Gedit supporte le langage D, en proposant une coloration syntaxique idoine.

Scite Si Scite propose lui aussi une coloration syntaxique pour le langage, son explorateur de fichier possède un filtre tel que les fichiers de code D `.d` ne sont pas considérés comme tel ; ils sont par conséquent invisibles. Pour y remédier, il suffira dans les paramètres dudit filtre d'autoriser la visualisation de tous les types de fichier.

Vim Le célèbre éditeur n'a rien à envier à ses semblables et propose lui aussi nativement une coloration syntaxique adaptée.

Code::Blocks Cet environnement de développement complet prend en charge le langage D et propose ainsi des fonctionnalités d'auto-complétion de code, parmi d'autres. Il n'est en revanche pas configuré par défaut pour utiliser le compilateur `ldc` ; il sera donc nécessaire de configurer un projet D de façon à l'utiliser explicitement.

Eclipse Les nombreuses extensions disponible pour Eclipse font sa force ; une extension dédiée au langage D, Descent⁹, est ainsi disponible pour Eclipse. Notez que, comme pour Code::Blocks, une configuration du projet D s'imposera pour que le compilateur `ldc` soit utilisé. Son installation est au passage décrite dans un tutoriel vidéo¹⁰.

Votre premier programme

Avant de commencer

Avant de commencer, vous devez savoir que tous les programmes doivent être composés d'une fonction `main` qui est la fonction principale du programme. Pour faire court, c'est le point de départ du programme. Une fonction est quant à elle composée des éléments suivants :

un en-tête celui-ci va faire office de carte d'identité de la fonction ;

un corps il définit le contenu de la fonction, c'est-à-dire ce qui est exécuté lors de l'appel de la fonction, délimité par des accolades ;

une valeur de retour elle correspond au résultat que la fonction va retourner.

L'en-tête de la fonction `main` peut s'écrire de plusieurs manières. En voici quelques-unes :

→ une fonction principale retournant un entier et ne prenant aucun argument :

```
int main () {  
    [...]  
}
```

→ une fonction principale retournant un entier et pouvant prendre des arguments issus de la ligne de commande :

```
int main (char[][] args) {  
    [...]  
}
```

→ une fonction principale ne retournant aucune valeur et ne prenant aucun d'argument :

```
void main (char[][] args) {  
    [...]  
}
```

Bonjour le monde

Quoi de plus naturel que de commencer son premier programme en D par le classique *Hello world* ?

Un exemple simple Créez un fichier `hello.d` dans lequel vous inscrirez les lignes suivantes :

```
import tango.io.Stdout;  
void main() {  
    // On imprime un message  
    Stdout("Bonjour le monde").nl;  
}
```

Ne reste plus qu'à compiler ce code.

Compilation et exécution Pour compiler ce fichier, lancez simplement la commande suivante :

```
$ ldc hello.d
```

Pour exécuter le programme ainsi généré :

```
$ ./hello
```

Nous vous laissons le plaisir d'admirer le résultat...

Quelques explications Prenons un instant pour commenter le code source précédent :

- sur la première ligne, on déclare l'utilisation du module `Stdout` sur lequel nous reviendrons plus tard ; sachez simplement que tous les modules D que vous aurez installés sous Fedora sont installés dans le dossier `/usr/include/d/` ; ici, le module `Stdout` est défini dans le fichier `/usr/include/d/tango/io/Stdout.d` ;
- la seconde ligne déclare notre fonction principale ; elle ne prend aucun argument et ne retourne aucune valeur ;
- La quatrième ligne, dans la fonction principale, permet d'écrire sur la sortie standard le message « Bonjour le monde » suivi d'un retour à la ligne (`.nl` pour *new line*).
- La dernière ligne clôt simplement notre fonction.

Les commentaires Vous avez sûrement remarqué dans l'exemple précédent la ligne suivante :

```
// On imprime un message
```

Cette ligne définit un commentaire ; les commentaires vous permettent de laisser des explications dans votre code (c'est d'ailleurs très fortement recommandé). Il existe plusieurs formes de commentaires :

→ commentaire sur une ligne :

```
// Mon commentaire  
///  
/// Ceci est une documentation
```

→ commentaires sur plusieurs lignes :

```

/*
  Ceci est un commentaire
*/

/*
 * Ceci est un commentaire
 */

/+
  Ceci est un commentaire
+/

/+
+ Ceci est un commentaire
+/

/**
  Ceci est une documentation
*/

/**
 * Ceci est une documentation
 */

/++
+ Ceci est une documentation
+/

```

Vous noterez que les commentaires estampillés `Ceci est une documentation` correspondent à une forme particulière relative à la documentation du code : le texte ainsi commenté pourra être lu par un programme externe qui le formatera dans un format plus lisible. Les connaisseurs auront reconnu un principe identique à celui mis en œuvre par les programmes Javadoc ou Doxygen. Ce point sera de toute façon développé dans une suite de cet article.

Les bases du langage

Nous nous attacherons, dans cette section, à présenter les différents éléments de syntaxe du langage D.

Les types

Comme tout langage de programmation, le D définit un certain nombre de types primitifs, présentés dans le tableau 1 page suivante. Notez toutefois que les valeurs des bornes pour chacun de ces types ont été générées sur un système 64 bits. Vous pourrez néanmoins générer un tableau pour votre architecture, si elle est différente, à l'aide d'un programme dédié¹.

Les opérateurs arithmétiques

On retrouvera dans D les opérateurs arithmétiques binaires usuels :

- + permet d'additionner deux variables numériques ;
- permet de soustraire deux variables numériques ;
- * permet de multiplier deux variables numériques ;
- / permet de diviser deux variables numériques ;
- % permet de renvoyer le reste de la division euclidienne de deux variables numériques (modulo).

Le programme suivant met en œuvre tous ces différents opérateurs :

```

import tango.io.Stdout;
void main() {
    uint a = 2;
    uint b = 2;
    Stdout.formatln("une_addition:_{}_+_{}_=_{}",
        a, b, a+b);
    Stdout.formatln("une_soustraction:_{}_-{}_=_{}",
        a, b, a-b);
    Stdout.formatln("une_multiplication:_{}_*_{}_=_{}",
        a, b, a*b);
    Stdout.formatln("une_division:_{}_/_{}_=_{}",
        a, b, a/b);
    Stdout.formatln("le_modulo:_{}_%{}_=_{}", a,
        b, a%b);
}

```

Incrémentation et décrémentation

Dans la même catégorie, signalons également les opérateurs d'incrément et de décrémentation, permettant respectivement d'ajouter et de retirer une unité à une variable :

```

int a = 2;
a = a + 1;
a += 1;
a++;
++a

```

Vous remarquez que les deux dernières formes sont plus rapides à écrire que les deux premières. Y a-t-il une différence entre `++a` et `a++` ? Oui, il y en a une et elle est subtile :

→ `a++` utilise la valeur de `a`, puis l'incrémente ;

→ `++a` incrémente la valeur de `a`, puis l'utilise.

Ce programme mettra en évidence cette différence :

```

import tango.io.Stdout;
void main() {
    uint a = 2;
    Stdout.formatln("une_incrementation:_{}_+_{}_=_{}",
        a, b, a++);
    a = 2;
    Stdout.formatln("une_incrementation:_{}_+_{}_=_{}",
        a, b, ++a);
    a = 2;
    Stdout.formatln("une_decrementation:_{}_-{}_=_{}",
        a, b, a--);
    a = 2;
    Stdout.formatln("une_decrementation:_{}_-{}_=_{}",
        a, b, --a);
}

```

On notera au passage que la notation `+=` ne se limite pas à l'addition :

```

uint a = 5;
a *= 2;
a /= 2;
a %= 2;

```

Remarque importante Nous vous mettons en garde quant aux types utilisés dans les opérations arithmétiques. Jusqu'ici le type `uint` était utilisé et nos résultats étaient des entiers. Mais ceci n'est pas systématiquement vrai : par exemple, la division de 1 par 3 renvoie un résultat réel ne pouvant être stocké dans une variable de type `uint`. Par conséquent, vous devez utiliser un type plus approprié, en l'occurrence les types `float` ou `double`, pour recevoir un tel résultat :

```

double a = 1;
a /= 3; // a vaut 0.33333333
int b = 1;
int c = 3;
double d = b / c; // d vaut 0.33333333

```

Les opérateurs de comparaison

Que serait un langage de programmation sans opérateur de comparaison ? Le langage D fournit les opérateurs usuels de comparaison décrits dans le tableau 2.

Opérateur	Description
« Égal à »	<code>a == b</code>
« Différent de »	<code>a != b</code>
« Strictement inférieur à »	<code>a < b</code>
« Inférieur ou égal à »	<code>a <= b</code>
« Strictement supérieur à »	<code>a > b</code>
« Supérieur ou égal à »	<code>a >= b</code>
« Et » logique	<code>a && b</code>
« Ou » logique	<code>a b</code>

TAB. 2 Les opérateurs de comparaison du langage D

Les structures conditionnelles

Nous allons, dans cette partie, mettre en œuvre les opérateurs de comparaison dans le cadre de structures de condition.

Type	Description	Borne inférieure	Borne supérieure
byte	Entier sur 1 octet	-128	127
ubyte	Entier positif sur 1 octet	0	255
short	Entier sur 2 octets	-32 768	32 767
ushort	Entier positif sur 2 octets	0	65 535
int	Entier sur 4 octets	-2 147 483 648	2 147 483 647
uint	Entier positif sur 4 octets	0	4 294 967 296
long	Entier sur 8 octets	-9 223 372 036 854 775 808	9 223 372 036 854 775 807
ulong	Entier positif sur 8 octets	0	18 446 744 073 709 551 615
float	Flottant sur 4 octets	$1,18 \times 10^{-38}$	$3,40 \times 10^{38}$
ifloat	Flottant imaginaire pur sur 4 octets	$1,18i \times 10^{-38}$	$3,40i \times 10^{38}$
double	Flottant sur 8 octets	$2,23 \times 10^{-308}$	$1,80 \times 10^{308}$
idouble	Flottant imaginaire pur sur 8 octets	$2,23i \times 10^{-308}$	$1,80i \times 10^{308}$
real	Flottant réelle sur 16 octets	$3,36 \times 10^{-4932}$	$1,19 \times 10^{4932}$
ireal	Flottant complexe imaginaire pur réelle sur 16 octets	$3,36i \times 10^{-4932}$	$1,19i \times 10^{4932}$
char	Caractère imprimable encodé en UTF-8 sur 1 octet	0	255
wchar	Caractère imprimable encodé en UTF-16 sur 2 octets	0	65 535
dchar	Caractère imprimable encodé en UTF-32 sur 4 octets	0	4 294 967 293
cfloat	Nombre complexe à composantes flottantes	$1,18 \times 10^{-38} + 1,18i \times 10^{-38}$	Non définie
cdouble	Nombre complexe à composantes doubles	$2,23 \times 10^{-308} + 2,23i \times 10^{-308}$	$1,19 \times 10^{4932}$
creal	Nombre complexe à composantes réelles	$3,36 \times 10^{-4932} + 3,36i \times 10^{-4932}$	$1,19 \times 10^{4932}$

TAB. 1 Les types du langage D

if...else Les structures if...else permettent la définition de traitements conditionnels, en fonction de la valeur d'une ou plusieurs variables :

```

int a = 2;
if (a == 2) {
    Stdout("la_variable_a_vaut_bien_2").nl;
}
else if (a == 3) {
    Stdout("la_variable_a_vaut_bien_3").nl;
}
// Le mot-cle else gere tous les cas non
// traites par les clauses if precedentes
else {
    Stdout.formatln("la_variable_a_vaut_{", a);
}
    
```

Comme on s'en doute, un tel code affichera le message la variable a vaut bien 2.

Un exemple plus complexe cette fois, mais abondamment commenté, mettant en œuvre des combinaisons d'opérateurs de comparaison :

```

int a = 2;
int b = 3;
// Ce bloc de code ne sera traite que si a vaut
// 2 ET b vaut 3
if (a == 2 && b == 3) {
    Stdout.formatln("la_variable_a_vaut_{_et_la_
        variable_b_vaut_{", a, b);
}
// Ce bloc de code sera traite si a vaut 2 OU b
// vaut 3
if (a == 2 || b == 3) {
    Stdout.formatln("la_variable_a_vaut_{_et_la_
        variable_b_vaut_{", a, b);
}
// Ce bloc ne sera traite que si a > 1 ET
// b >= 3
if (a > 1 && b >= 3) {
    Stdout.formatln("la_variable_a_vaut_{_et_la_
        variable_b_vaut_{", a, b);
}
// Ce bloc sera traite si a <= 2 OU b < 3
if (a <= 2 || b < 3) {
    Stdout.formatln("la_variable_a_vaut_{_et_la_
        variable_b_vaut_{", a, b);
}
    
```

Ce code affichera logiquement quatre fois le message la variable a vaut 2 et la variable b vaut 3.

switch...case Considérons le code suivant, comprenant une suite de conditions sur un entier :

```

int a = 4;
if (a == 0) {
    [...]
}
else if (a == 1) {
    [...]
}
else if (a == 2) {
    [...]
}
else if (a == 3) {
    [...]
}
else if (a == 4) {
    [...]
}
else {
    [...]
}
    
```

Sachez qu'il existe une manière plus élégante d'écrire ces conditions à l'aide des mots clés switch et case :

```

int a = 3;
switch (a) { // on met la variable a tester
    case 1: // cas ou a == 1
        [...]
        break
    case 2: // cas ou a == 2
        [...]
        break
    case 3: // cas ou a == 3
        [...]
        break
    case 4: // cas ou a == 4
        [...]
        break
    default: // equivalent du "else"
        [...]
}
    
```

Attardons-nous sur la clause break : celui-ci permet d'interrompre le parcours du bloc switch... case une fois une condition validée. Sans ce mot dans le code précédent, le code du bloc case 3 : sera non seulement traité, mais aussi tous les suivants (default compris), même si la variable a ne vaut pas 4. Cet autre exemple devrait vous éclairer à ce sujet :


```

int a = 1;
switch (a) { // on met la variable a tester
case 1: // cas ou la valeur est 1
[...]
```

Dans ce code, les blocs 1, 2 et 3 seront traités, mais pas les suivants, compte tenu de la clause `break` introduite dans le bloc 3. Cette structure se retrouve également en C et en C++. Mais à la différence de ces derniers, son utilisation en D ne se limite pas à des tests d'entiers. Ainsi, le code suivant met en œuvre une structure `switch...` case comparant des chaînes de caractères :

```

char[] prenom = "jonathan";
switch (prenom) {
case "jonathan":
[...]
```

Les boucles

On retrouve dans le langage D les différentes structures de boucles usuelles.

while Le langage D propose une structure `while`, qui exécute une série d'instructions tant qu'une condition donnée est vraie : dans le code suivant, tant que la variable `i` est inférieure à 10, sa valeur est affichée sur la sortie standard avant d'être incrémentée :

```

import tango.io.Stdout;
void main()
{
    uint = 0;
    while (i < 10)
    {
        Stdout(i).nl;
        i++;
    }
}
```

do...while Pendant de la boucle `while`, la boucle `do...while` se distingue de la précédente simplement par le fait que le passage dans la boucle est garanti au moins une fois :

```

import tango.io.Stdout;
void main()
{
    uint i = 10;
    do{
        Stdout(i).nl;
        i++;
    }
    while(i < 10)
}
```

for Les boucles `for` sont utilisées lorsque le nombre d'instructions à répéter est connu d'avance. En D, une telle boucle pourra être utilisée comme dans l'exemple suivant :

```

for (uint i = 0; i < 10; ++i)
{
    Stdout(i).nl;
}
```

Ce code affichera ainsi dix fois les valeurs successives de la variable `i`, incrémentée à chaque tour de boucle.

foreach Cette structure particulière s'applique aux parcours de tableaux (voir la section « Les tableaux ») : le code suivant permet ainsi de parcourir tous les éléments du tableau `a` et d'en afficher la valeur :

```

uint[5] a = [1,5,4,6,8];
foreach(element;a)
{
    Stdout(element).nl;
}
```

Les tableaux

Déclaration Il y a cinq façons de déclarer un tableau, listées dans la table 3.

Code	Description
<code>int* p;</code>	Pointeur vers les données
<code>int[3] s;</code>	Tableau statique
<code>int[] a;</code>	Tableau dynamique
<code>int[char[]] x;</code>	Tableau associatif
<code>int[][] m;</code>	Matrice

TAB. 3 Les différentes façon de déclarer un tableau en D

Les pointeurs Cette instruction déclare un pointeur de données :

```

int* p;
```

En plus d'une syntaxe identique à celle utilisée en C, le principe des pointeurs en termes d'adresses et de mémoires reste similaire. Pour cette raison, nous vous invitons à lire le tutoriel consacré aux pointeurs sur le Site du Zéro¹⁰.

Les tableaux statiques Un tableau statique est un tableau dont la taille est spécifiée dès sa déclaration. Sa taille ne variera pas d'un iota par la suite, étant fixée au moment de la compilation. Le code suivant crée un tableau de trois entiers et le « remplit » :

```

int[3] s;
s[0] = 50;
s[1] = 24;
s[2] = 98;
```

Les tableaux dynamiques Contrairement aux tableaux statiques, la taille des tableaux dynamiques est déterminée au cours de l'exécution du programme. Leur gestion fait appel aux pointeurs ainsi qu'au ramasse-miettes (*garbage collector*). Pour déclarer un tel tableau, la syntaxe suivante sera utilisée :

```

int[] a;
```

Le code suivant présente quelques exemples de manipulation des données de ces tableaux dynamiques :

```

 int[] a;
a.length = 2; // on donne une taille de 2
a[0] = 5;
a[1] = 8;
// ici 2+5 = 7, soit la nouvelle taille du
// tableau
a.length = a.length + 5
a[2] = 5;
a[3] = 7;
a[4] = 9;
a[5] = 3;
a[6] = 1;
    
```

Il est cependant conseillé de limiter les redimensionnements de tableaux, ceci afin de conserver de bonnes performances à l'exécution. Ainsi, un tel code serait à éviter :

```

 int[] a;
for(i = 0, i < 50, i++) {
    a.length = a.length + 1;
    a[i] = 0;
}
    
```

Une alternative serait la suivante :

```

 int[] a;
for(i = 0, i < 50, i++) {
    if (a.length == i) {
        // Si la taille du tableau est egale a
        // l'indice i, doublons sa taille
        a.length = a.length * 2;
    }
    a[i] = 0;
}
// Afin d'epargner la memoire, il faut ajuster
// exactement la taille du tableau selon le
// besoin, soit la valeur de l'indice i
a.length = i;
    
```

Les tableaux associatifs Un tableau associatif est un tableau associant à chacune de ses valeurs une *clé*. D'une certaine façon, le tableau associatif est une généralisation du tableau : là où les valeurs d'un tableau « classique » sont référencées par un index entier, un tableau associatif autorise n'importe quel type d'index.

Dans le code qui suit, nous déclarons un tel tableau d'entiers, indexé par des chaînes de caractères :

```

 int[char[]] x;
    
```

Nous pouvons alors référencer chaque élément du tableau avec une chaîne unique, comme dans l'exemple suivant (qui pourrait correspondre à un tableau décrivant pour des fruits leur quantité) :

```

 x["pomme"] = 2;
x["poire"] = 5;
x["orange"] = 7;
    
```

Les propriétés Les tableaux en D sont caractérisés par diverses propriétés, décrites dans la table 4 et accessibles selon la syntaxe suivante :

```

 (tableau). <propriété>;
    
```

Propriété	Description
sizeof	Taille du tableau multipliée par le nombre d'octets du type de ses éléments
length	Nombre d'éléments dans le tableau, non modifiable pour les tableaux statiques
ptr	Pointeur vers le premier élément du tableau
reverse	Renvoie le tableau avec l'ordre de ses éléments inversé (le premier devient le dernier et ainsi de suite)
sort	Renvoie le tableau trié
dup	Renvoie une copie du tableau sous la forme d'un tableau dynamique

TAB. 4 Propriétés des tableaux statiques en D

Nous aurons l'occasion, dans un prochain article, de tirer profit de ces propriétés.

Le slicing Le *slicing* est une technique particulière qui permet de copier une série d'éléments d'un tableau dans un autre. Le code suivant met en pratique cette technique :

```

 // On declare un tableau statique de 10 entiers
int[10] a = [1,2,3,4,5,6,7,8,9,10];
// On declare 2 tableaux dynamiques
int[] b;
int[] c;
// On declare un tableau statique de 3 elements
int[3] d;
// Slicing : on copie ici les elements
// d'indices 1 a 3 dans le tableau b (le
// premier indice est indexe a 0)
b = a[1..3];
// On copie tous les elements a partir de
// l'indice 4 jusqu'au dernier
c = a[4..$];
// On assigne la valeur 5 a tous les elements
// du tableau. Cela equivaut a :
// d[0] = 5; d[1] = 5; d[2] = 5;
d[0..2] = 5;
    
```

Concaténation de tableaux Le langage D fournit un mécanisme très simple de concaténation de tableaux, à travers l'opérateur ~ :

```

 int[10] a = [1,2,3,4,5];
int[10] b = [6,7,8,9,10];
int[] c;
// Le tableau dynamique c contiendra
// [1,2,3,4,5,6,7,8,9,10]
c = a ~ b;
    
```

Une combinaison de *slicing* et de concaténation est aussi possible :

```

 int[10] a = [1,2,3,4,5];
int[10] b = [6,7,8,9,10];
int[] c;
// c contiendra [1,2,3,4,5,6,7,8]
c = a ~ b[0..2];
    
```

Les matrices Comme bien d'autres langages, D prend en charge les *matrices*, soit un tableau à deux dimensions. Le code suivant créera ainsi une matrice de trois lignes et quatre colonnes :

```

 uint[3][4] matrix;
foreach (ligne; matrix)
{
    // On remplit chaque ligne du tableau avec
    // des 0 en faisant appel au slicing
    ligne[0..$] = 0;
}
    
```

Les chaînes de caractères

Tout bon langage de programmation prend en charge les chaînes de caractères, ce qui est évidemment le cas des langages C et C++, mais surtout de D, qui les considère avant tout comme des tableaux de caractères. La gestion des chaînes en D s'avère cependant plus simple, notamment grâce aux fonctions de *slicing* et de concaténation implémentées nativement :

```

 // Declaration et initialisation d'une chaine
char[] str = "Le_D_c'est_genial";
// Copie de la chaine de caracteres precedente
char[] str1 = str.dup;
// On ajoute a la suite de la chaine str une
// autre chaine
str += "_et_je_l'ai_adopte";

char str2[] = "Vive_le_D";
// on concatene str et str2 dans str3
char str3 = str ~ str2;
    
```

Comme indiqué dans le tableau 1 page 24, les caractères sont encodés par défaut en UTF-8 (type `char`). Il en est donc de même des chaînes de caractères. On pourra cependant définir des chaînes encodées en UTF-16 et UTF-32, respectivement à partir des types `wchar` et `dchar`, à l'aide de l'opérateur de `cast` :

```
// Une chaîne encodée en UTF-16
(wchar [])"J'aime_le_D";
// Une chaîne encodée en UTF-32
(dchar [])"J'aime_le_D";
```

Conclusion

Nous voici arrivés à la fin de cette introduction au langage D. Si elle est loin de couvrir tous les éléments du langage, elle devrait néanmoins vous permettre d'écrire dès maintenant vos premiers programmes en D. Les prochains articles consacrés à ce langage aborderont des notions plus poussées de programmation telles que la manipulation d'entrées-sorties, la programmation objet, ou encore la gestion des exceptions. Ils présenteront également les techniques de débogage de programmes D ou encore la documentation de codes source. Ne manquez donc pas les prochains numéros pour continuer votre initiation au langage D. ●

Jonathan MERCIER

- [1] http://fr.wikipedia.org/wiki/Walter_Bright
- [2] http://fr.wikipedia.org/wiki/Programmation_imp%C3%A9rative
- [3] http://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet
- [4] <http://fr.wikipedia.org/wiki/M%C3%A9ta-programmation>
- [5] http://fr.wikipedia.org/wiki/Programmation_par_contrat
- [6] http://fr.wikipedia.org/wiki/Low_Level_Virtual_Machine
- [7] <http://www.dsourc.org/projects/mango>
- [8] http://fr.wikibooks.org/wiki/Programmation_D/Annexes#Fichier_pour_ajouter_le_support_D_dans_Emacs
- [9] <http://www.dsourc.org/projects/descent>
- [10] <http://www.youtube.com/watch?v=EBHQdpreTfk>
- [11] http://fr.wikibooks.org/wiki/Programmation_D/Annexes#Programme_D_pour_connaitre_la_taille_des_types
- [12] http://www.siteduzero.com/tutoriel-3-14005-a-1-assaut-des-pointeurs.html#ss_part_2

Empathy et les thèmes

EMPATHY intègre depuis sa version 2.28 un mécanisme de thèmes pour les conversations. Ce dernier autorise en particulier l'utilisation des riches et nombreux thèmes disponibles pour Adium, un logiciel libre de messagerie instantanée pour Mac OS X. Pour installer et utiliser l'un d'entre eux :

- on le choisira parmi ceux proposés sur la page Adium dédiée aux thèmes^[1], sous la forme d'une archive ZIP ;
- on décompressera l'archive en question ; celle-ci contient un dossier suffixé par `.AdiumMessageStyle` qu'on déplacera dans le dossier `~/.local/share/adium/message-styles/` (à créer au besoin) ;
- on relancera Empathy le cas échéant

puis on sélectionnera le thème installé depuis la fenêtre de préférences du logiciel (*Édition* ➤ *Préférences* ➤ *Thèmes*).

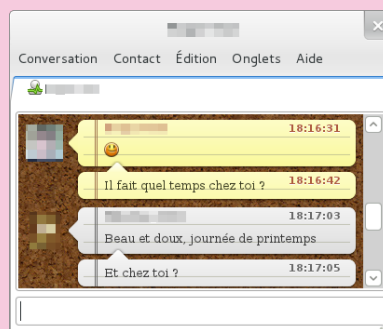


FIG. 45 Empathy et le thème NoteBoard

Notez toutefois que même si une large majorité de thèmes Adium sont compatibles avec Empathy, certains d'entre eux risquent d'avoir un mauvais rendu dans ce dernier ; les développeurs ont dressé une liste de thèmes populaires avec leur niveau de compatibilité avec Empathy^[2]. ●

Mohamed EL MORABITY

- [1] http://www.adiumxtras.com/index.php?a=search&cat_id=5
- [2] <http://live.gnome.org/Empathy/Themes>

Un centre multimédia avec XBMC

À l'heure où la révolution numérique a fait entrer l'informatique et l'Internet dans les mœurs, la dématérialisation des supports multimédia marque une nouvelle étape dans l'histoire des loisirs : le développement des lecteurs multimédia portables, des liseuses électroniques, et, surtout, des box Internet faisant office de centres multimédia en sont une preuve flagrante. Linux n'est pas en reste dans ce mouvement, bien au contraire, puisque de nombreuses solutions logicielles existent pour mettre en place votre propre station multimédia dans votre salon. À cette fin, nous vous proposons dans cet article de découvrir plus en détails XBMC, un lecteur multimédia complet disponible pour Fedora, qui fera de n'importe quelle machine un centre multimédia polyvalent.

Présentation

AL'ORIGINE, XBMC signifiait *XBox Media Center* : en effet, ce projet visait initialement à faire de la Xbox, la console de Microsoft, un centre multimédia. Depuis, XBMC a été réécrit et porté sur d'autres systèmes d'exploitation tels que Linux, Mac OS X et Windows. La particularité de ce logiciel est qu'il se présente comme une solution complète de lecture multimédia puisqu'il est capable de lire la plupart des formats audio et vidéo ainsi que les CD, les DVD ou encore les disques Blu-ray (ceux non protégés pour le moment). Son intérêt réside également dans le fait qu'il est capable de construire une bibliothèque multimédia à partir de fichiers aussi bien disponibles localement que sur un réseau via les protocoles SMB (*Server Message Block*), UPnP (*Universal Plug and Play*) ou encore NFS (*Network File System*). Il est en outre capable de recueillir des informations relatives aux vidéos répertoriées (titre, scénario, nom des acteurs, réalisateurs, producteur, type de film, année de sortie...). Ces informations sont récupérées de manière automatique par XBMC en se connectant à un *scraper* qui n'est ni plus ni moins qu'un service disponible sur Internet mis à disposition par les contributeurs du projet. Toutes ces informations sont alors présentées par le logiciel par le biais de menus élégants et agréables (voir figure 46 page suivante).

Installation et configuration sous Fedora

Prérequis

- Un PC, relativement récent (pas plus de quatre ans), et peu bruyant, sous Fedora ;
- une carte graphique avec support matériel de l'accélération graphique (voir ci-dessous) ;
- un écran, un système sonore 7.1, un vidéoprojecteur, une salle de cinéma, du pop-corn...

Notez que XBMC requiert d'importantes ressources graphiques pour peu que vous souhaitiez utiliser un port HDMI avec votre écran. L'installation des pilotes propriétaires *pourrait* donc s'avérer nécessaire pour les cartes ATI¹ et Nvidia², si les pilotes libres ne devaient pas donner entière satisfaction.

Installation de XBMC

XBMC requiert des codecs pour le support de formats soumis à brevets logiciels, tels que le MP3 ou le MPEG2 pour le support des DVD. Par conséquent, il n'est pas disponible dans les dépôts par défaut de Fedora. Il l'est en revanche sur le dépôt RPM Fusion³ que l'on configurera donc au préalable⁴. Une fois ce dépôt rajouté, on installera XBMC avec la commande suivante :

```
$ su -lc "yum install xbmc"
```

Lancement automatique de XBMC au démarrage

On configurera notre système de telle façon à ce que XBMC se lance automatiquement au démarrage de la machine, sans aucune intervention de votre part. Pour cela, on fera d'une part en sorte que le gestionnaire de connexion GDM nous connecte automatiquement à notre session, et d'autre part que XBMC soit lancé automatiquement au démarrage de cette dernière.

On modifiera le fichier `/etc/gdm/custom.conf` dans un premier temps ainsi :

```
$ su -lc "nano /etc/gdm/custom.conf"
```

On rajoutera alors sous la section `[daemon]` les lignes suivantes :

```
[daemon]
TimedLoginEnable=true
TimedLogin=(identifiant de connexion)
TimedLoginDelay=0
```

Sauvegardez alors vos modifications ; au prochain démarrage, votre système ouvrira automatiquement votre session utilisateur.

Afin que XBMC se lance à l'ouverture de la session, on lancera le programme de gestion des programmes au démarrage d'une session `gnome-session-properties` ainsi :

```
$ gnome-session-properties
```



FIG. 46 L'écran d'accueil de XBMC

À l'ouverture de l'application (voir figure 47), on cliquera sur le bouton *Ajouter* pour rajouter XBMC parmi les applications lancées au démarrage de la session. On saisira dans la boîte de dialogue apparaissant les données suivantes :

Nom XBMC

Commande xbmc

Commentaire Centre multimédia

Validez alors vos modifications en cliquant sur le bouton *Ajouter* et fermez l'application. Désormais XBMC sera lancé au démarrage de votre machine.

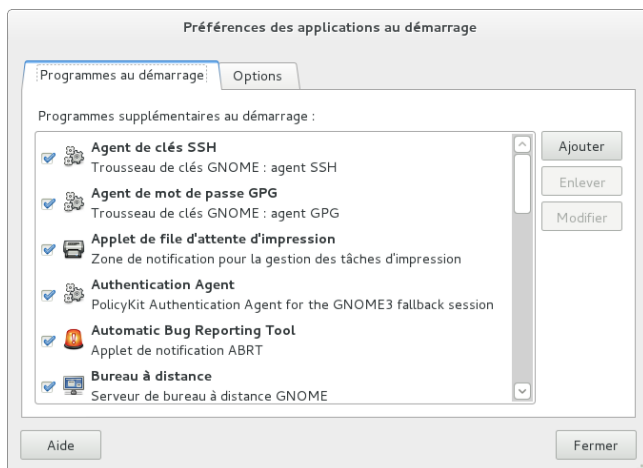


FIG. 47 gnome-session-properties

Utilisation de son téléphone Android comme télécommande

Les développeurs de XBMC ont mis au point une application Android, XBMC Remote, permettant d'utiliser son téléphone comme télécommande pour piloter XBMC (voir figure 50 page suivante). L'utilisation de cette application imposera quelques réglages sous Fedora que l'on détaillera.

Configuration du pare-feu XBMC Remote nécessite un accès sur le port 8080 sous Fedora pour fonctionner. On ouvrira ce port à l'aide

de l'interface de gestion du pare-feu system-config-firewall (accessible depuis le menu applicatif : **Système** ➤ **Administration** ➤ **Pare-feu**). Une fois l'interface lancée, on cliquera dans le volet latéral sur l'entrée *Autres ports*, puis sur le bouton *Ajouter* (voir figure 48) ; dans la boîte de dialogue qui apparaît suite à cette action, on cochera la case *Défini par l'utilisateur* de façon à pouvoir renseigner les champs suivants :

Port / Intervalle de ports 8080

Protocole tcp

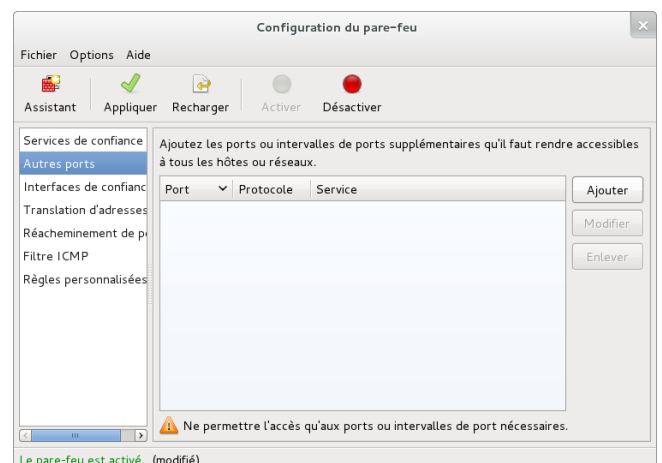


FIG. 48 system-config-firewall

Configuration de XBMC Il ne reste plus qu'à activer dans XBMC le contrôle distant via HTTP, en accédant au panneau dédié (**Paramètres** ➤ **Système** ➤ **Réseau** ; voir aussi la figure 49 page suivante). Dans celui-ci, à la section *Permettre le contrôle de XBMC en HTTP*, on renseignera les champs suivants :

Port 8080

Nom d'utilisateur (identifiant)

Mot de passe (mot de passe)

Notez que les nom d'utilisateur et mot de passe peuvent être choisis selon votre convenance. On cochera également les cases suivantes :

- ➔ Autoriser le contrôle de XBMC par des programmes locaux ;
- ➔ Autoriser le contrôle de XBMC par des programmes distants.

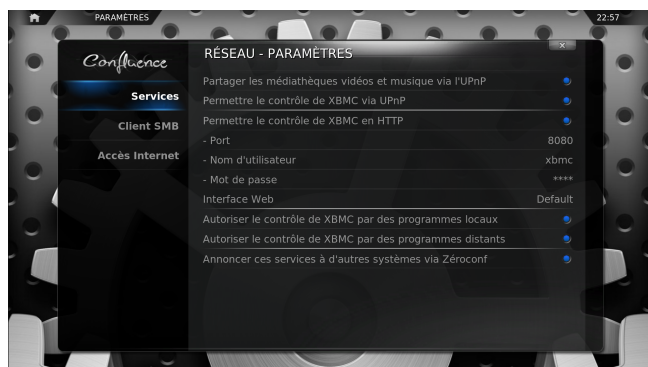


Fig. 49 Le panneau de configuration réseau de XBMC

Installation et configuration de XBMC Remote sous Android

L'application XBMC Remote est disponible sur l'Android Market⁹. Vous pourrez l'installer depuis ce dernier en lançant une recherche sur le terme « Official XBMC Remote » ou en scannant le code QR ci-contre.



Une fois installée, on lancera l'application sur son téléphone ; celle-ci nous invite à rajouter un nouvel hôte XBMC à piloter : on sélectionnera le bouton *Add Host* dans le menu de XBMC Remote et on remplira alors les champs proposés avec les valeurs suivantes :

Name of this instance Fedora XBMC

Host or IP address (nom de la machine ou adresse IP de la machine XBMC)

HTTP API port 8080

Username (identifiant défini dans les réglages réseau de XBMC)

Password (mot de passe pour l'identifiant précédent)

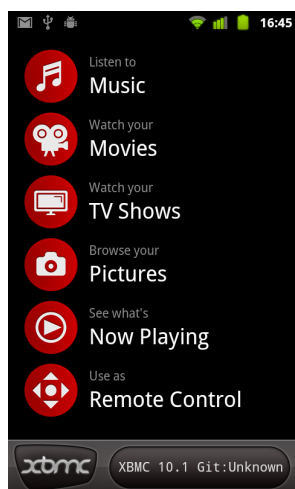


Fig. 50 XBMC Remote



Premiers pas avec XBMC

De nombreux sites Internet proposent des informations très complètes sur la façon d'utiliser au mieux cet outil (voir la section « Aller plus loin »). Cependant nous vous proposons un petit guide d'initiation qui vous permettra de prendre en main votre nouveau centre multimédia.

Navigation Sachez qu'il existe deux modes de navigation dans XBMC :

Le mode « fichier » comme son nom l'indique, ce mode permet de naviguer parmi vos différents fichiers média de la même manière que dans un navigateur de fichiers ; c'est aussi depuis ce mode que l'on définira de nouvelles sources de fichiers qui iront enrichir la bibliothèque⁶ (voir figure 51) ;

Le mode « médiathèque » une fois la bibliothèque constituée, vous pourrez naviguer parmi vos médias par type (musique, films, séries...) via de magnifiques menus (voir figure 52).

⁹ Notez que la connexion à ce service est devenue payante depuis peu.

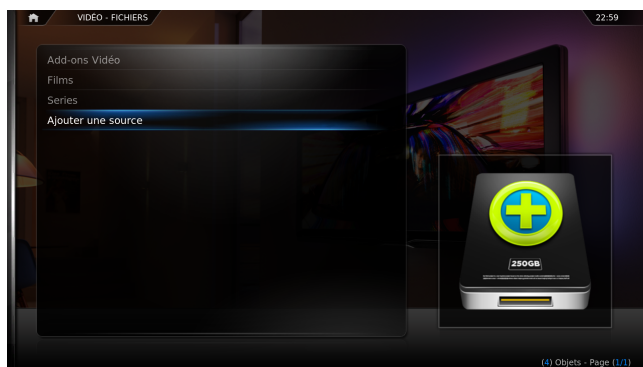


Fig. 51 XBMC en mode « fichier »

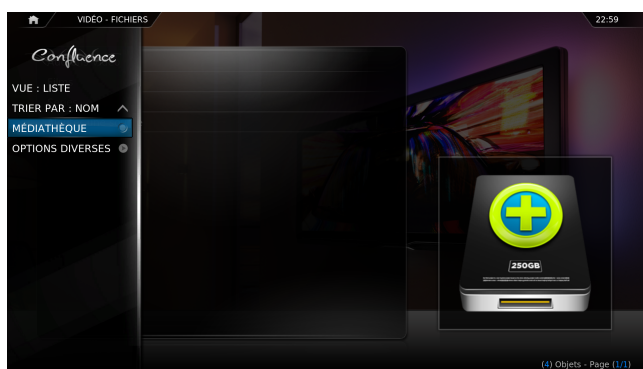


Fig. 52 XBMC en mode « médiathèque »

Les scrapers Un *scraper* est un service Web permettant à XBMC de récupérer des informations concernant les fichiers de la bibliothèque. Parmi ceux disponibles, en voici quelques-uns :

TheTVDB ce *scraper* regroupe une immense base de données relatives aux séries, avec résumés et images d'illustration⁷ (voir figure 53) ;

Ciné-Passion ce *scraper* est une base communautaire dédiée aux films, avec leur scénario, leur casting et même leur affiche⁸ (voir figure 54 page suivante) ;

allmusic pour les mélomanes, ce *scraper* récupérera pour chacun de vos albums leur pochette ou encore une biographie de vos artistes⁹.



Fig. 53 Informations relatives à une série récupérées grâce au scraper TheTVDB



FIG. 54 Informations relatives à une série récupérées grâce au scraper Ciné-Passion

Extensions XBMC intègre un système d'extensions pour étendre ses fonctionnalités. Parmi la pléthore d'extensions disponibles, citons notamment celle pour visionner les vidéos du site YouTube ou encore celles prenant en charge la télévision de rattrapage^[1] (voir figure 55); XBMC supporte ainsi les plates-formes françaises de *replay TV* suivantes :

- Pluzz (la *replay TV* de France Télévisions) ;
- M6 Replay ;
- W9 Replay ;
- Canal Replay.

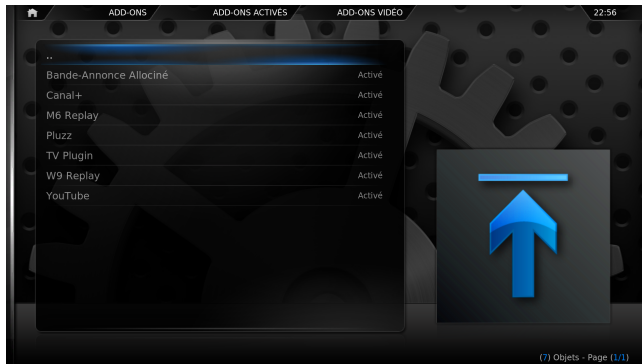


FIG. 55 Le gestionnaire d'extensions de XBMC

Aller plus loin

Nous vous proposons dans cette section quelques ressources Web pour mieux vous familiariser avec XBMC et étendre à l'infini ses possibilités.

Sites Internet

- <http://www.xbmc.org> : le site officiel de XBMC ;
- <http://www.xbmc.fr> : le site francophone.

Manuel d'utilisation

- http://xbmc.org/wiki/?title=XBMC_Online_Manual : le manuel de XBMC en anglais ;
- <http://manuel.xbmc.fr/> : le manuel en français.

Divers

- <http://passion-xbmc.org> : le site de la communauté des utilisateurs francophones de XBMC ;
- <http://www.xbmc.eu> : toutes les actualités relatives à XBMC ;
- <http://sourceforge.net/projects/xboxmediacenter/> : un projet proposant divers thèmes graphiques pour personnaliser XBMC ;
- <http://code.google.com/p/xbmc-addons/> : une collection de scripts et d'extensions pour XBMC.

Alexandre FRANDEMICHE

- [1] http://doc.fedora-fr.org/wiki/Carte_graphique_AMD_Radeon:_installation_du_pilote_propri%C3%A9taire
- [2] http://doc.fedora-fr.org/wiki/Carte_NVIDIA:_installation_des_pilotes
- [3] <http://rpmfusion.org/>
- [4] http://doc.fedora-fr.org/wiki/D%C3%A9p%C3%B4t_RPM_Fusion
- [5] <https://market.android.com/details?id=org.xbmc.android.remote>
- [6] http://wiki.passion-xbmc.org/Ajouter_des_sources
- [7] <http://thetvdb.com/>
- [8] <http://passion-xbmc.org/>
- [9] <http://allmusic.com/>
- [10] http://fr.wikipedia.org/wiki/T%C3%B3vision_de_rattrapage

Changer de gestionnaire de connexion

FEDORA, par défaut, propose l'environnement de bureau GNOME et en particulier son gestionnaire de connexion, GDM. D'autres gestionnaires existent cependant, tels que KDM, celui de KDE, ou encore Slim, celui de LXDE. Pour utiliser l'un ou l'autre de ces gestionnaires en lieu et place de GDM, il suffit de créer le fichier `/etc/sysconfig/desktop` et d'y définir le gestionnaire de connexion à uti-

liser ; par exemple, pour que KDM soit le nouveau gestionnaire de connexion par défaut à la prochaine fermeture de session :

```
$ su -lc "echo 'DISPLAYMANAGER=KDE' > /etc/sysconfig/desktop"
```

De même, pour Slim, on remplacera la chaîne `DISPLAYMANAGER=KDE` par `DISPLAYMANAGER=slim`.

Pour restaurer GDM, il suffira simplement de supprimer le fichier `/etc/sysconfig/desktop` :

```
$ su -lc "rm /etc/sysconfig/desktop"
```

Matthieu SAULNIER

Les événements à ne pas manquer

Nous vous proposons dans cet article de vous tenir au courant des rendez-vous incontournables du libre de cette année, avec des compte-rendus des événements passés et le programme de ceux à venir. Gardez donc votre agenda à portée de main en parcourant cet article pour ne pas manquer les prochaines rencontres auxquelles l'association Fedora-fr participera !

Fedora-fr.org change de look

Vous l'aurez sûrement remarqué, Fedora-fr.org dispose d'un nouveau thème (voir figure 56). Celui-ci s'inspire de la nouvelle charte graphique mise en place sur les différents sites du projet Fedora, en témoignent le choix des couleurs ou encore le placement du menu.

La page d'accueil a par ailleurs été complètement redéfinie puisqu'elle se présente désormais comme un portail d'accueil pour tous ceux qui souhaitent mieux connaître le projet Fedora et sa communauté francophone.

Un nouveau module de recherche a également été mis en place, plus rapide et plus efficace, rendant la navigation au sein du site toujours plus agréable et simple.

Notons enfin la mise en place d'un tableau de bord personnalisé, à l'adresse <http://mon.fedora-fr.org>, qui vous permettra de définir le contenu de votre propre page d'accueil parmi les derniers messages du forum¹, les derniers billets parus sur le planet Fedora² ou encore les messages Twitter³ et Facebook⁴ de la communauté.



FIG. 56 La nouvelle charte graphique du site Fedora-fr.org

Solutions Linux 2011



Vous y êtes passé ? Nous nous sommes alors sans doute rencontrés ! Fedora-fr a en effet fait acte de présence à cette douzième édition du salon Solution Linux⁵. L'événement a reflété les grands changements survenus dans le monde du logiciel libre de par la présence de Mageia⁶ (le fork communautaire de la distribution Mandriva), LibreOffice⁷ ou encore MeeGo⁸.

Contrairement aux deux précédentes éditions, le salon ne s'est pas tenu à la Porte de Versailles mais a fait son retour au CNIT, à la Défense, du 10 au 12 mai 2011. Près de 10 000 visiteurs étaient attendus pour pas moins de 250 exposants présents afin de les accueillir. Parmi les sujets des tables rondes, conférences et formations proposées au public, citons notamment le *cloud computing*⁹, les CMS¹⁰, le stockage des données, la sécurité, ou encore la virtualisation. Près d'une soixantaine d'associations étaient par ailleurs représentées dans le village associatif, dont Fedora-fr évidemment.

Rencontres Fedora 15

Les Rencontres Fedora sont l'occasion de rencontrer des contributeurs du projet Fedora et autres acteurs du monde libre autour de conférences et d'ateliers. Cette année, les Rencontres Fedora se dérouleront sur deux jours, les samedi 4 et dimanche 5 juin 2011, à la Cité des sciences et de l'industrie à Paris :

→ le samedi se déroulera dans le cadre des Premiers samedis du libre¹¹, un événement mensuel auquel diverses associations du monde du libre se retrouvent dans le cadre d'une *install party* ; des conférences seront également au programme :

14 h - 15 h présentation de GNOME 3, de Systemd et de toutes les autres nouveautés de Fedora 15,

15 h 30 - 18 h présentation de la virtualisation sous Fedora 15 avec BoxGrinder¹² ;

→ le dimanche sera plutôt orienté contribution : plusieurs contributeurs se regrouperont afin de travailler sur divers sujets, du développement à la traduction, en passant par le *packaging* RPM ; non seulement vous êtes les bienvenus à cette journée, mais vous pour-

rez en plus rejoindre l'un des ateliers et participer ainsi au projet Fedora.

Le programme complet de ces deux jours est disponible sur le wiki de l'association Fedora-fr^[1].

regroupe tous les événements du logiciel libre en France. Plus spécialement pour Fedora en France, vous avez également la possibilité de suivre la rubrique « actualités » du forum du portail Fedora-fr.org^[15].

Kévin RAYMOND

Rencontres mondiales du logiciel libre 2011



RMML
Rencontres Mondiales
du Logiciel Libre

Les Rencontres mondiales du logiciel libre^[2] proposent sur plusieurs jours des conférences, ateliers et stands dédiés aux logiciels libres. Organisées par une équipe de volontaires, leur accès est libre et gratuit pour tous. La douzième édition de ce rendez-vous annuel se tiendra cette année à Strasbourg du 9 au 14 juillet 2011.

Les deux premières journées seront dédiées au grand public, tandis que les jours suivants seront plutôt consacrés aux conférences et au village associatif. De nombreux thèmes seront abordés, notamment la santé, la culture, le droit du libre, l'administration des collectivités, la science, la recherche, les systèmes embarqués ou encore la sécurité.

Restez informés

Si vous ne savez pas quand se déroule le prochain événement dédié au libre dans votre région, faites un tour sur l'Agenda du libre^[3], qui



- [1] <http://forums.fedora-fr.org/>
- [2] <http://planet.fedora-fr.org/>
- [3] <http://twitter.com/fedorafr>
- [4] <http://www.facebook.com/#!/pages/Fedora-Fr/132102243492051>
- [5] <http://www.solutionslinux.fr>
- [6] <http://mageia.org/>
- [7] <http://www.libreoffice.org/>
- [8] <http://meego.com/>
- [9] http://fr.wikipedia.org/wiki/Cloud_computing
- [10] http://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_contenu
- [11] <http://premier-samedi.org/>
- [12] <http://boxgrinder.org/>
- [13] http://asso.fedora-fr.org/wiki/Rencontres_Fedora_15_%C3%A0_Paris
- [14] <http://2011.rmll.info>
- [15] <http://www.agendadulibre.org/>
- [16] <http://forums.fedora-fr.org/viewforum.php?id=38>

Réaliser des images du système

FAIRE un *snapshot* de son système, c'est un peu comme prendre une photo de celui-ci : on va figer sur la pellicule, capturer à un instant *t* toute l'organisation de vos données. Se limitera-t-on aux données personnelles ? Que nenni ! Vous pourrez faire des captures complètes de votre système Fedora. La pellicule ? C'est un fichier image ISO.

Nous utiliserons la commande `genisoimage` pour sauvegarder, par exemple, votre répertoire de téléchargement :



```
$ genisoimage -o
snapshot-test.iso
$HOME/Téléchargements
```

Vous pourrez par la suite monter l'image ISO `snapshot-test.iso` générée et vérifier que la photo est réussie.

Pour sauvegarder cette fois tout votre dossier utilisateur dans une image horodatée :



```
$ genisoimage -o
$USER-$(date +%F).iso
$HOME
```

Pour sauvegarder enfin la partition `/boot` :



```
$ su -lc "genisoimage -o
boot.iso /boot"
```

Il est ainsi aisé de créer des sauvegardes prêtes à être gravées sur DVD ou copiées sur tout autre média de stockage.

Matthieu SAULNIER

Utilisation de YUM : allons plus loin

Si vous avez lu le premier numéro de Muffin¹, voici la suite du dossier consacré à l'un des plus importants programmes sur Fedora : YUM. Nous avons en effet vu dans la première partie de ce dossier le principe de fonctionnement du gestionnaire de paquets et sa prise en main ; dans cet article, nous nous attacherons à présenter quelques options de YUM plus poussées qui devraient vous aider à mieux administrer votre système Fedora, au-delà des simples ajout et suppression de programmes.

Quelques subtiles options de YUM

La clause info

CETTE clause permet de récupérer un descriptif du paquet qui lui est passé en paramètre, qu'il soit installé sur le système ou non ; elle affiche également d'autres compléments relatifs à ce paquet tels que sa version, son architecture, sa taille, le dépôt le proposant, ou encore sa licence :

```
$ yum info arora
[...]
Nom           : arora
Architecture  : x86_64
Version       : 0.11.0
Révision      : 2.fc15
Taille        : 811 k
Dépôt         : fedora
Résumé        : A cross platform web browser
URL           : http://code.google.com/p/arora/
Licence       : GPLv2+
Description    : Arora is a simple, cross
                  : platform web browser based on
                  : the QtWebKit engine. Currently,
                  : Arora is still under
                  : development, but it already has
                  : support for browsing and other
                  : common features such as web
                  : history and bookmarks.
```

La clause repolist

Cette clause liste simplement tous les dépôts activés pour votre système :

```
$ yum repolist
id du dépôt  nom du dépôt  statut
fedora       Fedora 15 - x86_64  24 085
updates      Fedora 15 - x86_64 - Updates  872
repolist: 25 767
```

On notera qu'en plus des dépôts activés, la commande affiche, dans la colonne « statut », le nombre de paquets composant chacun d'entre eux. Pour afficher tous les dépôts définis dans votre système, y compris ceux désactivés (les dépôts de test par exemple), on rajoutera le mot-clé all à la commande :

```
$ yum repolist all
```

La clause provides

Si vous vous êtes déjà demandé quel paquet fournit un programme ou une bibliothèque donnée, la clause provides devrait répondre à vos attentes : celle-ci interroge en effet les dépôts pour déterminer le paquet proposant la ressource qui lui est passée en paramètre. Une fois le paquet identifié, ne reste plus qu'à l'installer. Notez que dans le cas où l'on souhaite trouver un paquet à partir d'un fichier, cette clause attend le chemin complet supposé de ce dernier (ou au moins une expression rationnelle) :

```
$ yum provides /bin/cp
$ yum provides */thunderbird
```

Notez cependant que pour les bibliothèques dynamiques en revanche (des fichiers *.so.*), le nom même du fichier suffit en revanche :

```
$ yum provides ld-linux.so.2
```

La clause check

Cette clause va vérifier l'intégrité de votre base de données RPM :

```
$ yum check
check all
```

Tel est le message que cette commande doit retourner, après quelques minutes d'exécution, si aucune dépendance brisée ni paquets dupliqués n'ont été détectés. Dans le cas contraire, il est vital de corriger les problèmes signalés pour le bon fonctionnement de votre système. La commande package-cleanup devrait vous y aider d'ailleurs.

La clause deplist

Cette clause renvoie, pour un paquet passé en paramètre, la liste de ses dépendances :

```
$ yum deplist wget
[...]
Recherche de dépendances :
paquet : wget.x86_64 1.12-3.fc15
dépendance : libssl.so.10()(64bit)
provider: openssl.x86_64 1.0.0d-1.fc15
dépendance : librt.so.1()(64bit)
provider: glibc.x86_64 2.13.90-9
dépendance : libc.so.6(GLIBC_2.8)(64bit)
provider: glibc.x86_64 2.13.90-9
dépendance : libdl.so.2()(64bit)
provider: glibc.x86_64 2.13.90-9
dépendance : /sbin/install-info
provider: info.x86_64 4.13a-15.fc15
dépendance : libz.so.1()(64bit)
provider: zlib.x86_64 1.2.5-3.fc15
dépendance : rtdld(GNU_HASH)
provider: glibc.x86_64 2.13.90-9
provider: glibc.i686 2.13.90-9
dépendance : libcrypto.so.10()(64bit)
provider: openssl.x86_64 1.0.0d-1.fc15
dépendance : librt.so.1(GLIBC_2.2.5)(64bit)
provider: glibc.x86_64 2.13.90-9
dépendance : /bin/sh
provider: bash.x86_64 4.2.10-2.fc15
```

Une commande à réserver à tous ceux qui s'intéressent à la façon dont YUM gère les dépendances d'un paquet...

La clause clean

YUM conserve les différentes informations relatives aux dépôts, et en particulier les paquets les composants, dans des fichiers de cache durant un certain temps (dans le dossier `/var/cache/yum/` pour information). De cette façon, YUM n'a pas à interroger systématiquement les dépôts en ligne à chaque utilisation, ce qui rallongerait considérablement ses traitements. Contrairement au gestionnaire de paquets APT, YUM est en mesure de mettre à jour son cache automatiquement, et ce aussi souvent que nécessaire.

Il peut néanmoins s'avérer nécessaire de purger ce cache manuellement, pour forcer YUM à récupérer de nouvelles données des dépôts : quoique ce cas demeure exceptionnel, cette opération s'imposera si, par exemple, YUM devait renvoyer des erreurs indiquant des incohérences dans sa base de données :

```
$ su -lc "yum clean all"
```

Ainsi, au prochain lancement de YUM, les informations des dépôts seront de nouveau générées.

La clause makecache

Cette clause est le pendant de la précédente dans la mesure où elle permet de lancer manuellement une mise à jour des méta-données de YUM relatives aux dépôts :

```
$ su -lc "yum makecache"
```

L'option --exclude

Si, pour diverses raisons, vous souhaitez exclure un paquet particulier d'une mise à jour, l'option `--exclude` est faite pour vous :

```
$ su -lc "yum update
--exclude=kernel,kernel-devel"
```

Cette commande va ainsi empêcher les paquets `kernel` et `kernel-devel` d'être mis à jour. Cette opération est cependant loin d'être anodine, sachant que les mises à jour visent avant tout à corriger des bogues voire des problèmes de sécurité ; aussi ne l'utilisez que si vous savez ce que vous faites...

L'option --skip-broken

Cette option est utile lorsque certains paquets que l'on souhaite installer ou mettre à jour présentent des dépendances non disponibles dans les dépôts. Un tel cas de figure — rare cependant — se produit par exemple lorsque le noyau est mis à jour, mais que des paquets `kmod-(module)` installés, construits pour une version spécifique du noyau, n'ont pas encore été mis à disposition dans les dépôts pour celui-ci. Dans pareille situation, YUM refusera de poursuivre la mise à jour ; les dépendances de ces paquets en erreur sont dites « brisées ». L'utilisation de cette option fera en sorte que YUM ne bloque pas une transaction à cause de ces derniers : ils ne seront tout simplement pas traités, contrairement aux autres paquets aux dépendances satisfaites :

```
$ su -lc "yum update --skip-broken"
```

Cette option est une alternative peut-être plus pratique à `--exclude` pour ce type de problèmes.

L'option --setopt

Les options de YUM ne se limitent cependant pas aux deux précédentes susmentionnées : YUM comporte en effet une foultitude de réglages « cachés », documentés dans la page de manuel de [yum.conf](#). Ces réglages sont passés à la commande `yum` au travers de l'option `--setopt=(option)`.

Parmi ces différents réglages, retenons les suivants.

→ Le réglage `throttle=(bande passante)` permet de modifier à la volée la bande passante rendue disponible au téléchargement des paquets :

```
$ su -lc "yum install pidgin
--setopt=throttle=3M"
```

Cette commande permet ainsi d'allouer temporairement à YUM une bande passante de 3 Mio par seconde.

→ YUM propose divers réglages également permettant de modifier la façon dont ses messages seront affichés, en y ajoutant notamment de la couleur :

```
$ yum list *gstreamer*
--setopt=color_list_available_install=green
--setopt=color_list_installed_reinstall=blue
```

Cette commande va ainsi lister les paquets contenant la chaîne « gstreamer » dans leur nom, en affichant ceux déjà installés sur le système en bleu et ceux disponibles en ligne en vert.

→ Un dernier réglage, introduit dans la dernière version de YUM (livrée avec Fedora 15), est particulièrement notable puisqu'il permet, à la désinstallation d'un paquet, de supprimer également toutes ses dépendances non requises par d'autres paquets en même temps :

```
$ su -lc "yum remove empathy
--setopt=clean_requirements_on_remove=1"
```

Cette commande désinstallera ainsi non seulement `Empathy`, mais aussi par exemple toutes les bibliothèques requises par ce dernier et non utiles à d'autres programmes. Un tel réglage permet ainsi de maintenir sa base de données de paquets sans le moindre effort, en ne conservant que les paquets réellement utiles au fonctionnement du système et des applications le composant.

Retour vers le passé

Nous nous intéresserons ici plus particulièrement à une clause relativement peu connue de YUM, mais ô combien utile : la clause `history`. Celle-ci vous permettra non seulement de suivre l'historique de vos opérations YUM, mais aussi d'annuler certaines d'entre elles le plus simplement du monde. Pour mieux cerner son fonctionnement, nous allons présenter dans la suite un exemple d'utilisation.

Supposons que nous souhaitons installer, par exemple afin de compiler un programme à partir de ses sources, le paquet `qt-devel` :

```
$ su -lc "yum install qt-devel"
```

En plus de ce paquet seront également installés une foultitude de paquets de développement. Une fois votre programme compilé, vous n'avez *a priori* plus besoin de tous ces paquets et souhaitez donc en toute logique les désinstaller. Or vous avez lancé depuis quantité de mises à jour et installé également d'autres paquets. Comment donc retrouver d'une part les paquets installés avec `qt-devel` et les supprimer sans devoir les retrouver un par un ? C'est là que la clause `history` nous sera d'un grand secours. On pourra ainsi lister les différentes opérations YUM exécutées sur le système avec la commande suivante :

```
$ su -lc "yum history list"
[...]
Id | Identifiant utilisateur | Date et heure | Action | Modifié
-----
159 | Casper <Casper> | 2011-05-09 23:49 | Install | 11
```

Nous identifions alors, à l'aide de l'horodatage, notre installation du paquet `qt-devel` et de ses dépendances. Nous pouvons alors demander l'annulation — ici la désinstallation — de cette transaction ainsi :

```
$ su -lc "yum history undo 159"
```

Le numéro 159 correspond à l'identifiant unique de notre transaction YUM listée dans les sorties de la commande `yum history list`. En plus de vous aider à maintenir un système propre et léger, cette clause s'avérera extrêmement utile pour annuler par exemple une mise à jour présentant des instabilités : on reviendra ainsi sans effort à l'état précédent du système pour mieux analyser le problème.

Un interpréteur de commandes dans YUM

YUM est doté d'un interpréteur de commandes au travers de la clause `shell`. Ce *shell* interactif vous permettra de combiner plusieurs opérations YUM en une seule commande, par exemple une mise à jour et l'installation d'un nouveau paquet :

```
$ su -lc "yum shell"
[...]
Configuration du shell Yum
> update
[...]
> install gnome-tweak-tool
[...]
> run
```

Ainsi qu'on peut le constater dans cette dernière commande, une fois les différentes opérations YUM saisies, ces dernières seront exécutées avec la commande `run`. L'interpréteur se dote par ailleurs d'une aide, accessible avec la commande `help` :

```
$ su -lc "yum shell"
[...]
Configuration du shell Yum
> help
```

Ce *shell* s'avère extrêmement puissant puisqu'il autorise également l'exécution de véritables scripts YUM : on peut ainsi aisément imaginer un script que vous pourriez lancer suite à l'installation d'une nouvelle version de Fedora et qui installerait vos programmes favoris et lancerait une mise à jour en une seule passe : considérons par exemple à cette fin le script suivant, supposé écrit dans un fichier `post-install.yum` :

```
update
install gparted pavucontrol bash-completion \
dconf-editor
run
```

Pour l'exécuter, il suffit simplement de passer à la commande `yum shell` le chemin du fichier :

```
$ su -lc "yum shell post-install.yum"
```

Notez cependant que vous serez invités, comme pour n'importe quelle transaction YUM par défaut, à confirmer chaque opération à chaque fois qu'il vous le sera demandé. Avec un tel script, que vous pourrez maintenir et ainsi enrichir, vous retrouverez aisément votre environnement logiciel à chaque nouvelle version de Fedora. ●

Matthieu SAULNIER

 [1] <http://mag.fedora-fr.org/w/images/e/ec/MUFFIN-01.pdf>

La gravure (trop) facile

TOUT le monde connaît le logiciel de gravure K3B. Mais saviez-vous que ce dernier n'est qu'une interface à la commande `wodim` ? Voici quelques commandes pour mieux appréhender `wodim` en console.

→ Identifier son graveur :

```
$ wodim --devices
wodim: Overview of accessible
drives (1 found) :
-----
0 dev='/dev/scd0' rwrw--
: 'HL-DT-ST' 'BDDVDRW
CT21N'
```

→ Graver des fichiers :

```
$ wodim speed=1
dev=/dev/scd0 -eject
(liste de fichiers)
```

→ Graver des répertoires :

```
$ wodim speed=1
dev=/dev/scd0 -eject
(répertoire)
```

→ Graver des images disque amorçables :

```
$ wodim speed=1
dev=/dev/scd0 -eject
-data (fichier ISO)
```

Vous aurez noté que la vitesse de gravure est déterminée par le paramètre `speed=N`,

qui lancera une gravure en vitesse `Nx` ; la vitesse peut jouer un rôle majeur dans la qualité de la gravure, décroissante avec la vitesse de gravure.

N'hésitez pas à adapter à vos besoins les quelques commandes ci-dessus ou à consulter la page de manuel de l'outil^[1]. ●

Matthieu SAULNIER

 [1] <http://manpages.ubuntu.com/manpages/natty/en/man1/wodim.1.html>

Nous rejoindre en cuisine

Muffin est le magazine de la communauté Fedora francophone ; libre et gratuit, il se veut le reflet de cette communauté en s'adressant à tous ses membres, qu'ils soient utilisateurs débutants ou contributeurs chevronnés. Sa réalisation est avant tout guidée par les mêmes motivations que celles du projet Fedora : le partage.

En tant que projet communautaire, Muffin a besoin de contributeurs pour vous proposer régulièrement de nouveaux numéros, en phase à la fois avec l'évolution du projet Fedora et avec notre communauté. Quels que soient vos talents, vous pouvez apporter votre contribution à Muffin en rejoignant notre équipe de rédaction.

Notre équipe de rédaction

LES contributeurs de Muffin sont répartis en six équipes aux rôles définis mais variés ; nul doute que vous trouverez votre place parmi l'une d'entre elles :

Direction les membres de cette équipe coordonnent et arbitrent les travaux des différentes équipes tout en gérant le planning de développement du magazine ;

Rédaction cette équipe regroupe des rédacteurs, tant réguliers qu'occasionnels, chargés de rédiger des articles sur les sujets qui leur tiennent à cœur ;

Actualité les membres de cette équipe sont avant tout des personnes intéressées par l'actualité du libre en général et de Fedora en particulier ; leur rôle est de proposer et produire des articles en phase avec les événements Fedora au moment de la parution du magazine ;

Relecture garante de la qualité technique et rédactionnelle du magazine, cette équipe veille à relire et corriger au besoin les articles soumis à validation, aussi bien dans la forme que dans le fond ;

Design cette équipe s'occupe de la composition du magazine ainsi que de sa mise en page ;

Marketing les membres de cette équipe sont chargés de la communication autour des nouveaux numéros de Muffin, cela pour leur plus large diffusion auprès de tous.

Comment participer ?

Pour contribuer à la réalisation de votre magazine, rien de plus simple ! Avant toute chose, vous vous inscrirez à la liste de diffusion du projet^[1] qui vous permettra de suivre les différents échanges relatifs à la conception du magazine. Vous pourrez alors faire part de votre motivation à l'ensemble de ses contributeurs en y envoyant un message de présentation.

Les travaux de rédaction sont rédigés sur un wiki^[2] ; vous vous verrez par la suite attribuer un compte par les administrateurs du site vous permettant de vous y connecter pour saisir vos contributions. Vous serez d'ailleurs invité à créer une page personnelle reprenant votre présentation. Vous voilà enfin membre de notre équipe, prêt à travailler au prochain numéro de Muffin^[3] !

Alexandre FRANDEMICHE



[1] <http://mailing-list.fedora-fr.org/mailman/listinfo/fedora-fr-news>

[2] <http://mag.fedora-fr.org/wiki/>

[3] <http://mag.fedora-fr.org/wiki/ReglesFonctionnement>

Rédacteur en chef Alexandre FRANDEMICHE

Conception graphique et mise en page Mohamed EL MORABITY, Kévin RAYMOND

Ont participé à ce numéro Mohamed EL MORABITY, Alexandre FRANDEMICHE, Kévin RAYMOND, Matthieu SAULNIER et Jonathan MERCIER

Relecture et corrections Fabien ARCHAMBAULT, Sébastien PASCHE, Éric QUERELLE



Les articles contenus dans ce magazine sont publiés sous la licence *Creative Commons Attribution-ShareAlike 3.0*. Cela signifie que vous êtes libres de les reproduire, distribuer et communiquer à la condition expresse de citer le nom des auteurs originaux ainsi que le nom du magazine (mais pas d'une manière qui suggérerait que les auteurs ou l'équipe de rédaction de Muffin vous soutiennent ou approuvent votre utilisation de l'œuvre).

Tous les produits, marques et logos cités sont la propriété exclusive de leur(s) détenteur(s).

Crédits d'images :

→ *Playing with light and CDs*, Vetto, ©1500 ;
→ *Keyboard*, DaveOnFlickr, ©1100 ;
→ *Jan:18*, Kim Love, ©1100 ;
→ *GNOME Lovelock background*, Emily DIRSH, Máirín DUFFY, Jef VAN SCHENDEL, ©1100 ;
→ *Muffin*, cavitt, ©1100 ;
→ *Lots of boxes*, Edward BAKER, ©1100 ;
→ *Remote Control*, Phil LONG, ©1100 ;
→ *Speed V2*, Gabry89, ©1100.

Les icônes d'illustration sont issues du projet Glyphish et publiées sous licence *Creative Commons Attribution 3.0 Unported*.

Ce magazine a été conçu avec $\LaTeX$, Inkscape et Gimp.

