Introduction à Ion3

Julien Barnier

Version 0.2
11 août 2005

Table des matières

1 Introduction

Ion est un gestionnaire de fenêtres développé par Tuomo Valkonen. La définition qu'en donne son auteur est : *« Ion is a tiling tabbed window manager designed with keyboard users in mind »*, soit, pour tenter une traduction approximative, *« Ion est un gestionnaire de fenêtres en mosaïque et à onglets, conçu pour ceux qui préfèrent utiliser le clavier »*.

Il a pour principal caractéristique une très grande légèreté, une esthétique sommaire mais une utilisation très efficace. Le site officiel de Ion se trouve à l'adresse http://iki.fi/tuomov/ion/.

La dernière version stable est la version 2, dite Ion2, et la version en cours de développement, mais parfaitement utilisable, est baptisée Ion3. Ce document se base sur cette dernière version.

Enfin, il faut noter que ce document a été rédigé par un débutant, pour des débutants. Il est donc fort probable que des erreurs, lacune et autres imprécisions se soient glissées dans ces lignes. L'auteur se fera un plaisir de les corriger pour peu qu'on les lui signale !

2 Installation

Nous ne détaillerons pas ici les procédures d'installation de Ion. Le nombre de dépendances étant limité, sa compilation à partir des sources ne devrait pas poser trop de problèmes.

Pour les heureux debianistes, des paquets tout prêts seront installés par un simple :

```
apt-get install ion3 ion3-doc ion3-scripts
```

Et pour les heureux gentooistes, un petit coup de `emerge` devrait suffire :

```
emerge ion
```

Enfin, une dernière méthode d'installation est de recompiler depuis les sources récupérées directement sur le dépôt `darcs` du projet. Pour cela, vous devez avoir installé `darcs`, puis lancer la série de commandes suivantes :

```
$darcs get --partial http://iki.fi/tuomov/repos/ion-3/
$cd ion-3
$sh predist.sh -snapshot
```

Cela devrait vous générer un paquet source contenant tout ce qu'il faut pour compiler un Ion3 tout frais tout chaud.

3 Configuration

La configuration de Ion s'effectue entièrement à l'aide de fichiers texte. Ces fichiers sont en fait des bouts de code en Lua, un langage de script léger. Les fichiers de configuration du système sont en général dans `/etc/X11/ion3` ou dans `/usr/local/etc/ion3`.

Plutôt que de travailler sur les fichiers s'appliquant à l'ensemble des utilisateurs de votre système, il est préférable de placer tout ça dans un répertoire `~/.ion3`. La première chose à faire est donc de copier les fichiers de configuration généraux (fichiers `cfg_*.lua`) dans ce répertoire de votre dossier personnel.

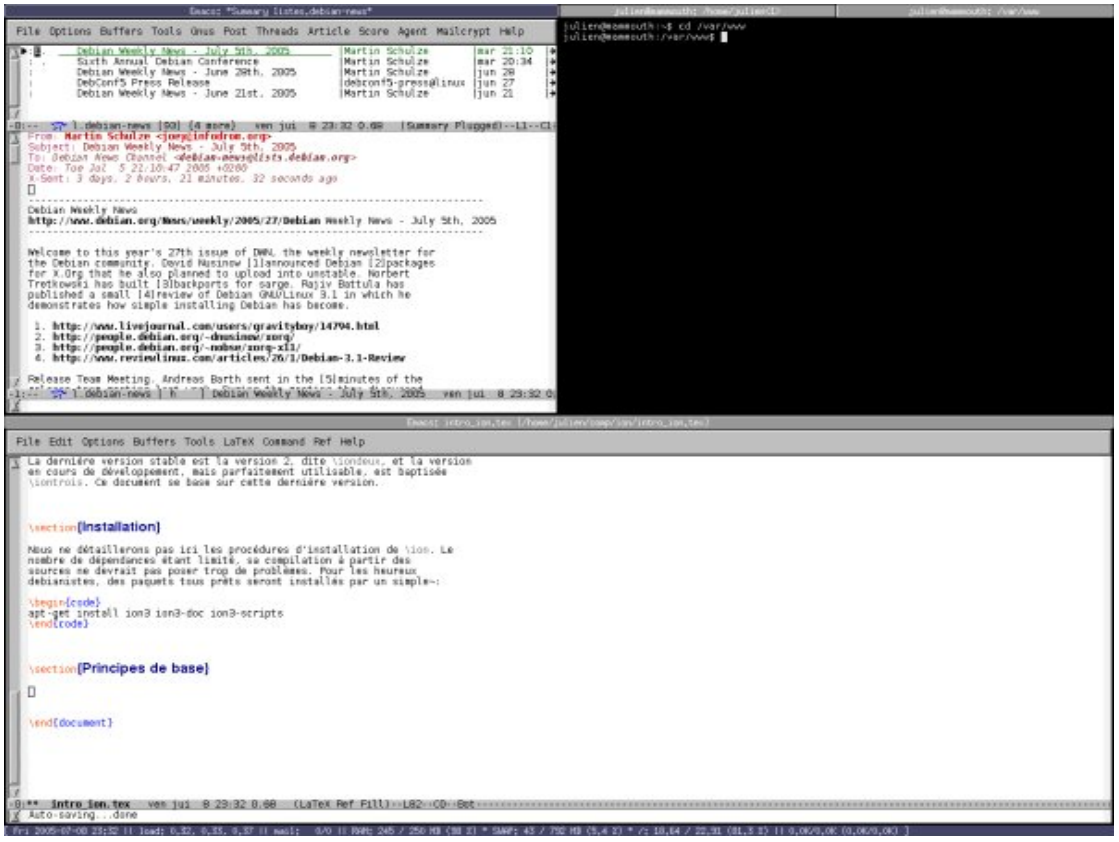

FIG. 1 – Capture d'écran d'un bureau

4 Principes de base

Pour comprendre le fonctionnement d'Ion, il faut savoir que celui-ci s'organise autour d'un petit nombre de concepts. Il est donc nécessaire de commencer par un (tout petit) peu de théorie. Mais pour avoir une illustration de ce à quoi ces concepts renvoient, vous pouvez vous référer à la capture d'écran, figure 1.

Écran *(screen)* Un écran correspond à un écran physique. Cette notion n'a de sens que pour ceux qui utilisent plusieurs moniteurs en *Xinerama* (ce dont nous ne parlerons pas ici).

Bureau (*workspace*) Un bureau ou espace de travail correspond à la même notion que dans les autres gestionnaires de fenêtres. Il s'agit d'un « bureau virtuel »qui contient lui-même un certain nombre d'objets : soit des cadres soit des *client windows* en mode plein écran. Il existe différents types de bureaux, en particulier les « mosaïque »(*tiled*) (où les cadres ne se recouvrent pas et remplissent tout l'écran, comme dans la figure 1) et les « flottants »(*floating*), dans lesquels les cadres peuvent se chevaucher et sont librement positionnables.

Cadre (*frame*) Un cadre est un espace rectangulaire délimitant une partie de l'écran. Sur la figure 1, le bureau est divisé en trois cadres.

Fenêtre (*client window*) Les fenêtres clientes sont des fenêtres faisant tourner une application. Il peut y avoir plusieurs fenêtres dans un seul cadre, c'est le cas pour le cadre supérieur droit de la figure 1. Chaque fenêtre est alors signalée par un onglet (*tab*) en haut du cadre.

Barre d'état (*statusbar*) La barre d'état de Ion est une ligne de texte fixe située en bas de l'écran et contenant différentes informations de type date, charge système, mails reçus, trafic réseau, etc. (tout ceci est bien entendu configurable).

Fenêtre transitoire (*transient window*) Une fenêtre transitoire est une fenêtre dépendant d'une application et prévue pour n'apparaître que brièvement. Les boîtes de dialogues devraient être gérées comme des fenêtres transitoires.

La principale particularité de Ion réside dans les « bureaux en mosaïque »(*tiled workspaces*). Dans ce cas, soit les applications lancées sur ce bureau utilisent toutes la totalité de l'écran, soit elles prennent la forme d'une « mosaïque »de cadres pouvant eux-mêmes contenir d'autres cadres ou une ou plusieurs fenêtres d'applications.

Ainsi, si vous lancez un programme classique comme Firefox, celui-ci s'affichera au premier abord en plein écran, sans même de décorations de fenêtres à part l'onglet en haut de l'écran.

Si une application souhaite afficher une boîte de dialogue, c'est à dire (normalement) une fenêtre transitoire, celle-ci apparaît à l'intérieur du même cadre, en général en bas de celui-ci. C'est un peu déconcertant au début mais on s'y fait très vite (si vous ne comprenez pas la dernière phrase, faites un `Fichier > Ouvrir` dans n'importe quelle application et vous y verrez plus clair).

Il peut arriver que les applications ne gèrent pas ces fenêtres transitoires de manière correcte, et cela peut générer des problèmes d'affichage. Nous en reparlerons dans la section 8.

5 Raccourcis claviers

Pour vivre et même survivre dans Ion, il est indispensable d'imprimer immédiatement la page de manuel (`man ion3`, vous avez de la chance, elle devrait s'ouvrir automatiquement au premier démarrage), de parcourir la liste des raccourcis claviers et de les garder sous la main. La première chose à savoir est de comprendre ce que signifie le symbole `Mod1`. En règle générale, il s'agit de la touche `Alt`, mais la commande `xmodmap` peut vous aider à l'identifier.

Une astuce qui peut être utile, et qui limite notamment les collisions entre les raccourcis claviers de Ion et ceux des applications, et de remplacer la touche `Alt` par la touche *Windows*, juste à côté, vous savez, celle que vous n'utilisez jamais. Pour cela, c'est très simple, il vous suffit d'éditer le fichier `~/.ion3/cfg_ion.lua` et de le modifier pour avoir la ligne suivante :

```
MOD1="Mod4+"
```

Une autre possibilité est de carrément redéfinir la définition des `Mod<n>`. Ceci peut se faire à l'aide de la commande `xmodmap` (`man xmodmap`).

Enfin, il est bien évidemment possible de redéfinir tous les raccourcis claviers et d'en rajouter à volonté. Tout ceci se fait dans les fichiers de configuration, et notamment dans `~/.ion3/cfg_bindings.lua`. Consultez *Configuring and extending Ion3 with Lua* pour plus d'informations. Dans tout ce qui suit, nous ne mentionnerons que les raccourcis claviers par défaut ?

6 Visite guidée

Plutôt que de lister les raccourcis claviers, ce qui n'apporterait rien de plus que la très complète page de manuel, nous ne reculons devant rien et nous vous proposons plutôt une visite « interactive »de Ion.

6.1 Démarrer Ion

La première chose à faire est d'arriver à le lancer. Pour cela, il faut fermer votre session X et éditer le fichier `.xsession` pour qu'il contienne la ligne suivante :

```
ion3
```

C'est également dans ce fichier que vous pourrez mettre les applications à lancer automatiquement au démarrage de Ion, du type `xscreensaver --no-splash` ou bien différents *dockapps* de WindowMaker.

Ensuite, faites un simple `startx` pour lancer tout ça. Vous devriez vous retrouver sur un bel écran vide, avec juste une ligne en bas qui vous donne l'heure, la charge et le nombre de mails en attente, ainsi qu'un joli onglet *<empty frame>* tout en haut.

Une autre méthode consiste à intégrer Ion au sein de son *desktop manager* préféré (`kdm`, `gdm`, `xdm`...). N'en utilisant pas, nous attendrons qu'une âme charitable passe par ici et veuille bien nous indiquer la marche à suivre...

6.2 Lancer des applications

Bon, et maintenant, je fais quoi ? Pas de panique ! Voici quelques touches utiles qui vous permettront de lancer un ou deux programmes :

`F1` vous permettra d'afficher une page de manuel au choix. Remarquez la ligne de dialogue qui s'affiche en bas d'écran pour vous permettre de saisir le nom.

`Mod1+F1` vous affiche directement la page de manuel d'Ion. Pratique quand on a oublié les raccourcis clavier.

`F2` vous lance directement un terminal. Le choix du programme utilisé est celui défini dans `/etc/alternatives` (un petit `update-alternatives -config x-terminal-emulator` devrait vous permettre de personnaliser tout ça).

`F3` vous affiche un invite de commande vous permettant de lancer ce que vous voulez.

`Mod1+F3` vous permet de saisir directement du code Lua.

`F4` vous permet d'établir immédiatement une connexion SSH vers une autre machine dans un terminal.

`F5` vous permet d'éditer un fichier.

`F6` vous permet de visualiser le contenu d'un fichier.

Les commandes utilisées pour ces deux dernières fonctions sont définies dans votre fichier `mailcap`, qui fait les associations entre les types MIME et les applications utilisées pour les ouvrir. Ainsi, si vous voulez utiliser Emacs pour voir et éditer les fichiers avec `F5` et `F6`, rajoutez la ligne suivante à la fin du fichier `~/.mailcap`[1] :

```
application/*; emacs %s; edit= emacs %s
```

Tout cela est bien beau, me direz-vous, mais y'a quand même bien un menu quelque part, non ? Non ? et bien si ! Pour cela il vous suffit de presser la merveilleuse touche `F12`, et vous verrez apparaître un beau et classique menu avec tout ce qu'il faut pour lancer des programmes, fermer ou redémarrer `Ion`, voir tester les différents « styles »fournis en standard (ne vous attendez pas cependant à des fenêtres transparentes sous des menus déroulants animés avec ombre portée...). De plus les heureux debianistes constateront que le menu comporte une entrée *Debian* avec le menu du système à l'intérieur.

Là encore, les menus sont entièrement configurables, et il est tout à fait possible, voir conseillé, d'en créer de nouveaux tout en les associant à des raccourcis claviers personnalisés. Voir le *Guided tour to Ion2* pour un exemple simple ou *Configuring and extending Ion3 with Lua* pour plus d'informations.

6.3 Utiliser les cadres

6.3.1 Naviguer dans les onglets

Que se passe-t-il si vous lancez plusieurs applications ? Elles s'ouvrent toutes sur le même écran (le même bureau, puisque pour l'instant il n'y en a qu'un) et occupent à chaque fois la totalité de l'écran (qui est composé d'un unique cadre). Plusieurs onglets apparaissent alors au-dessus et permettent de passer d'une application (d'une fenêtre) à une autre à l'aide de la souris. De même, un clic droit sur un onglet vous offre un menu contextuel permettant diverses actions. Mais là encore, les raccourcis clavier vous feront gagner du temps :

`Mod1+K 1, Mod1+K 2, Mod1+K 3, Mod1+K 4...` vous permet de passer directement au n-ième onglet du cadre.

`Mod1+K P` passe à l'onglet précédent.

`Mod1+K N` passe à l'onglet suivant.

`Mod1+K M` ouvre le menu contextuel de l'onglet.

`Mod1+K C` force la fermeture de l'onglet visible.

`Mod1+C` ferme l'onglet visible.

Note : sur mon clavier français à moi que j'ai, la combinaison `Mod1+chiffre` fonctionne avec les touches de la rangée supérieure du clavier sans passer par la touche `Shift`. Ainsi, pour faire `Mod1+1` je fais en fait `Mod1+&` et pour faire `Mod1+2` je fais en fait `Mod1+é`. Je ne sais pas si c'est une mauvaise configuration de ma part, mais en tous cas c'est fort pratique [2].

6.3.2 Créer et supprimer des cadres

Vous savez donc maintenant comment faire pour passer d'un onglet à l'autre quand vous n'avez qu'un seul cadre visible. Mais comment faire pour créer de nouveaux cadres et les manipuler, histoire d'avoir plus d'une application visible simultanément sur le bureau ?

1. On me signale dans mon oreillette que cette fonctionnalité nécessite la présence du programme `run-mailcap` sur le système. Merci Tuomo.
2. On me signale à nouveau dans mon oreillette que oui, c'est fait exprès. Merci Tuomo.

La création de cadres se fait en scindant l'écran, soit horizontalement soit verticalement. `Mod1+S` effectuera une division verticale, tandis que `Mod1+K S` fera la même chose horizontalement. Vous pouvez ainsi diviser les cadres encore et encore selon vos besoins.

Pour supprimer une division, un petit `Mod1+K X` pourra vous être utile, il permet de supprimer le cadre actuel. S'il contenait des onglets, ceux-ci seront rassemblés avec ceux déjà existants. À noter qu'un cadre est également supprimé lorsque vous fermez son dernier onglet avec `Mod1+C` ou le menu contextuel.

6.3.3 Manipuler les cadres

Pour passer d'un cadre à l'autre, vous pouvez soit cliquer dedans, évidemment, soit utiliser les raccourcis claviers qui vont bien :

`Mod1+P` pour passer au cadre du dessus ;

`Mod1+N` pour passer au cadre du dessous ;

`Mod1+Tab` pour passer au cadre de droite ;

`Mod1+K Tab` pour passer au cadre de gauche.

Autre opération vitale, savoir redimensionner les cadres pour les adapter aux fenêtres qu'ils contiennent. Pour cela, encore, vous avez le choix entre la souris et le clavier :

`Mod1+Button1` **ou** `Mod1+Button2` permet de redimensionner les cadres à la souris ;

`Mod1+R` permet de passer en mode redimensionnement (*resize mode*) pendant une brève période.

Une fois dans le mode redimensionnement, les raccourcis suivants sont disponibles :

`Gauche, Droite, Haut, Bas, F, B, P, N` agrandit le cadre dans la direction choisie ;

`Shift+Gauche, Shift+Droite, Shift+Haut, Shift+Bas, Shift+F, Shift+B,...` rétrécit le cadre à partir du côté indiqué ;

`Escape` annule le mode redimensionnement ;

`Return` quitte le mode redimensionnement. À noter que la sortie du mode se fait automatiquement au bout d'un certain laps de temps paramétrable.

Bien, vous avez désormais tous pleins de cadres sur votre écran. Certains sont vides, d'autres contiennent une seule fenêtre, certains ont plusieurs onglets. Comment je fais pour organiser mes fenêtres entre tous ces cadres ?

Si vous activez un cadre et que vous lancez une application, celle-ci va s'exécuter à l'intérieur du cadre sélectionné. Mais vous pouvez bien évidemment passer une fenêtre d'un cadre à un autre. Là encore, les méthodes souris et claviers sont disponibles :

`Button1` un simple *drag and drop* sur un onglet vous permet de faire passer sa fenêtre d'un cadre à un autre ;

`Mod1+A` ouvre une fenêtre de requête et vous permet de spécifier le nom d'une fenêtre (pensez à `Tab` pour la saisie) qui sera déplacée vers le cadre actif ;

`Mod1+T` vous permet de marquer (*tag*) une fenêtre. Un petit signe distinctif apparaît en haut à droite de son onglet. La même touche permet de supprimer le marquage ;

`Mod1+K A` vous permet de déplacez toutes les fenêtres marquées vers le cadre actif.

D'autres commandes sont disponibles pour manipuler les cadres, comme vous le verrez dans la page de manuel de Ion. Parmi celles-ci, deux sont particulièrement utiles :

`Mod1+Return` vous permet de basculer la fenêtre active en mode plein écran. Si vous vous sentez un peu à l'étroit dans votre cadre, vous pouvez ainsi à tout moment maximiser l'affichage de la fenêtre puis revenir à l'affichage précédent avec la même combinaison de touches ;

`Mod1+K K` permet de retourner vers le dernier objet actif, que celui-ci soit sur le même bureau ou non. C'est très pratique si vous avez à basculer régulièrement entre deux fenêtres.

6.4 Créer des bureaux

Ion supporte évidemment les bureaux virtuels. Avoir plusieurs bureaux vous permet par exemple d'avoir différentes répartitions de cadre.

Il existe trois types de bureaux :

`WIonWS` il s'agit du type de bureaux que nous venons de décrire, avec des cadres « en mosaïque » ;

`WFloatWS` il s'agit des bureaux flottants dont nous parlerons dans la section 6.5 ;

`WPaneWS` il s'agit d'un troisième type de bureaux avec des cadres « élastiques ». N'ayant pas bien compris le fonctionnement de ce type de bureau, nous n'en parlerons pas dans ce document pour le moment.

Pour créer un bureau, deux solutions s'offrent à vous :

`F9` vous permet de créer un nouveau bureau. Ion commence par vous demander un nom pour ce bureau (facultatif), puis son type (voir ci-dessus) ;

`Mod1+F9` vous permet de créer immédiatement un nouveau bureau avec le type par défaut, sans que vous ayez à saisir une quelconque information.

Lorsque vous avez plusieurs bureaux, se déplacer d'un bureau à l'autre est très simple et se fait au clavier :

`Mod1+1, Mod1+2, Mod1+3, ...` vous permet de passer directement au n-ème bureau ;

`F9` si vous indiquez le nom d'un bureau déjà existant (pensez à `Tab`), cette commande vous permet de passer sur ce bureau plutôt que d'en créer un nouveau.

Pour supprimer un bureau, c'est très simple, il suffit de fermer toutes les fenêtres que ce bureau contient et de faire un `Mod1+C` sur le bureau désormais vide.

Pour déplacer des fenêtres d'un bureau à l'autre, il vous suffit d'utiliser la méthode décrite dans la section 6.3.3, à l'aide de `Mod1+A` ou de `Mod1+T` et `Mod1+K A`.

Enfin, à noter que lorsque vous basculez une fenêtre en mode plein écran à l'aide de `Mod1+Return`, Ion crée en fait un nouveau bureau placé en fin de liste avec cette fenêtre maximisée.

6.5 Utiliser un bureau flottant

Arrivés ici, vous devez commencer à avoir une petite idée de l'utilisation et peut-être de l'intérêt d'un gestionnaire de fenêtres gérant des « cadres en mosaïque ». Ce modèle n'est cependant pas optimal pour la totalité des applications. Utiliser The Gimp de cette manière, par exemple, peut s'avérer un peu ardu [3].

La particularité de Ion est de proposer également des bureaux flottants, avec des cadres mobiles pouvant se recouvrir, à la manière de la plupart des gestionnaires de fenêtres actuels.

Nous avons vu dans la section 6.4 comment créer un bureau de ce type. Nous nous attarderons moins sur l'utilisation des bureaux flottants, mais voici quand même quelques indications potentiellement utiles.

Lorsque vous lancez une application dans un bureau flottant, celle-ci s'affiche dans un cadre de manière « traditionnelle », sans prendre toute la place disponible. Vous pouvez déplacer ce cadre en cliquant sur son onglet. Vous pouvez également le « replier »en double-cliquant sur l'onglet en question.

Une des particularités de Ion est la possibilité, comme dans le type de bureau précédent, de regrouper des fenêtres dans un même cadre flottant. Ceci se fait en cliquant sur un onglet avec le bouton du milieu et en faisant un *drag and drop* sur un autre cadre, ou bien à l'aide des commandes de déplacement de fenêtres déjà décrites (`Mod1+A, Mod1+T, Mod1+K A`...

Voici quelques raccourcis bien utiles pour les bureaux flottants :

`Mod1+Tab`, `Mod1+K Tab` permet de passer d'un cadre flottant à l'autre ;

`Mod1+Button1` permet de déplacer un cadre sans avoir à cliquer spécifiquement sur un onglet ;

`Mod1+Button3` permet de redimensionner un cadre ;

`Mod1+P`, `Mod1+N` permet de faire passer le cadre actif au premier plan / en arrière plan.

À noter que Ion considère que le cadre actif est celui sous lequel se trouve la souris, sans que vous ayez à cliquer dessus pour l'activer. Un cadre peut être actif sans être au premier plan. Ceci peut parfois surprendre dans les premiers temps d'utilisation.

3. Ah, on me signale qu'apparemment certains auraient découvert que même Gimp est plus facile à utiliser dans un bureau en mosaïque. Disons que je n'ai pas encore franchi le temps d'adaptation nécessaire.

7 Astuces diverses

Les « astuces »présentées ci-dessous consistent en fait en l'utilisation de modules de Ion (en fait des scripts en Lua) fournis avec le programme. Pour cela, il va falloir éditer les fichiers de configuration. Comme indiqué dans la section 3, plutôt que de bidouiller les fichiers s'appliquant à l'ensemble des utilisateurs de votre système, il est préférable de placer tout ça dans un répertoire `~/.ion3`.

La première chose à faire est donc de copier les fichiers de configuration généraux (sous Debian, il s'agit des fichiers `cfg_*.lua` situés dans le répertoire `/etc/X11/ion3`) dans ce répertoire de votre dossier personnel.

7.1 Personnaliser la barre d'état

La barre d'état est cette petite ligne de texte en bas de l'écran qui contient par défaut un certain nombre d'informations sur le système : par défaut date et heure, charge, mails en attente.

La barre d'état est gérée par le module `mod_statusbar`. Pour activer ce module (ce qui est le cas par défaut), vous devez avoir la ligne suivante dans votre `cfg_ion.lua` :

```
dopath("mod_statusbar")
```

Le contenu de cette zone est bien évidemment totalement paramétrable. La personnalisation se fait dans le fichier `cfg_statusbar.lua`. On peut ainsi choisir où placer cette zone sur l'écran et surtout modifier son contenu à l'aide de la variable `template`.

Pour ajouter des informations à la barre d'état, le plus simple est d'utiliser les scripts Lua fournis spécialement à cet effet. Ceux-ci sont nommés `statusd_*.lua` et sont situés, du moins sous Debian, dans le répertoire `/usr/share/ion3`. Pour utiliser l'un de ces scripts, il vous suffit d'indiquer la chaîne `%nom_du_module` dans la variable `template`.

Ainsi, si vous souhaitez afficher des informations sur la charge réseau de votre système, fournies par le script `statusd_netmon.lua`, il vous suffit d'ajouter `%netmon` à l'endroit voulu dans la variable template. À noter qu'en préfixant le nom du script par `<`, `>` ou `|`, vous pouvez modifier l'alignement de la chaîne de caractères générée, et qu'en utilisant un espace précédé du caractère `%` vous introduisez un « espace élastique »qui peut faciliter l'affichage.

Juste pour information, ma barre d'état actuelle est la suivante :

```
template="[ %date || load:% %>load || mail:% %>mail_new/%>mail_total || %>sysmon || %>netmon ]"
```

7.2 Utiliser des *dockapps* WindowMaker

La barre d'état, c'est joli et pratique, mais on regrette parfois ses bons vieux *dockapps* qui faisaient notre bonheur sous WindowMaker. Et bien figurez-vous que Ion gère très bien la plupart des *dockapps* en question.

Si vous souhaitez en profiter, il vous faut d'abord activer le module `mod_dock` dans votre `cfg_ion.lua` à l'aide de la ligne suivante :

```
dopath("mod_dock")
```

Vous pouvez également configurer plus finement ce mode à l'aide du fichier `cfg_dock.lua`. Vous pouvez ainsi personnaliser la position d'affichage des dockapps à l'écran à l'aide de la variable `pos` :

```
pos="tr"
```

Vous pouvez choisir entre des *dockapps* « flottants », qui se superposent aux cadres existants quand ils sont affichés, ou bien des *dockapps* « embarqués », toujours visibles et qui s'affichent à la manière de la barre d'état par défaut. Ceci se règle à l'aide de la variable `mode` :

```
mode="floating"
```

Attention : les *dockapps* embarqués sont incompatibles avec la barre d'état. Donc soit vous choisissez de ne pas utiliser la barre d'état, soit vos *dockapps* seront forcément flottants.

Enfin, vous pouvez également configurer le raccourci clavier à utiliser pour afficher/masquer vos *dockapps*. La touche par défaut est `Mod1+D`, mais vous pouvez changer ça à la ligne suivante :

```
kpress(MOD1.."D", "mod_dock.set_floating_shown_on(_, 'toggle')")
```

Bon, tout cela est bien beau, mais je fais comment, moi, pour lancer mes *dockapps* ? Et bien c'est très simple : il vous suffit d'exécuter le programme correspondant et celui-ci sera automatiquement ajouté à la liste. Et si vous souhaitez un lancement automatique au démarrage, il vous suffit de les ajouter dans votre fichier `.xsession`.

7.3 Utiliser un *frame sp*

Un *frame sp*, encore dénommé *scratchpad*, est un cadre pouvant être affiché/masqué à tout moment à l'aide d'un raccourci clavier, quel que soit le bureau sur lequel vous vous trouvez. Ça peut être très pratique pour avoir toujours une console sous la main, par exemple.

Par défaut le *scratchpad* est un cadre de taille plutôt réduite s'affichant au milieu de l'écran, mais vous pouvez bien entendu régler tout ça.

Pour activer cette fonctionnalité, fournie par le module `mod_sp`, vous devez avoir la ligne suivante dans votre `cfg_ion.lua` :

```
dopath("mod_sp")
```

Le raccourci par défaut pour basculer l'affichage du *scratchpad* est `Mod1+Espace`. Vous pouvez modifier cela en éditant la ligne suivante du fichier `cfg_sp.lua` :

```
kpress(MOD1.."space", "mod_sp.set_shown_on(_, 'toggle')")
```

7.4 Inclure un bureau flottant dans un cadre en mosaïque

Et oui, tout est possible avec Ion. Vous pouvez donc inclure l'équivalent d'un bureau flottant à l'intérieur d'un cadre d'un bureau en mosaïque. Pour cela, il suffit d'exécuter le code Lua suivant (à l'aide de `Mod1+F3`, faut-il le rappeler) :

```
_:attach_new({type="WFloatWS"})
```

Et vous pouvez même inclure dans un cadre de ce bureau flottant l'équivalent d'un autre bureau en mosaïque. Qui lui-même, etc., etc. La capture d'écran suivante, tirée du site de Matthieu Moy, donne une idée de ce que ça peut donner :

http://www-verimag.imag.fr/~moy/ion/ion-screenshot.png

7.5 Associer une application à un cadre (*winprops)*

Vous l'aurez sans doute remarqué, lorsque vous quittez Ion, celui-ci sauvegarde la disposition de vos bureaux et de vos cadres, que vous retrouvez telle quelle au lancement suivant. Ceci est très pratique, surtout si on l'associe au fait qu'Ion est capable d'associer une application et un cadre de manière à ce que celle-ci s'exécute toujours dans le même cadre.

Les utilisations potentielles sont multiples. Vous pouvez par exemple prévoir un cadre de petite taille sur votre premier bureau et prévoir que celui-ci servira à l'exécution d'un lecteur de musique ou d'un logiciel de messagerie instantanée.

Dans l'exemple qui suit, nous faisons en sorte que toutes les fenêtres d'Emacs s'affichent dans le même cadre. La marche à suivre est la suivante[4] :

- Tout d'abord, il faut déterminer l'instance et la classe de la fenêtre de l'application concernée. Pour cela, il faut lancer la commande suivante dans un terminal et cliquer dans la fenêtre choisie :

  ```
  xprop WM_CLASS
  ```

- Vous devriez obtenir un résultat de ce type :

  ```
  WM_CLASS(STRING) = "emacs", "Emacs"
  ```

- La seconde chose à faire est de nommer explicitement le cadre qui vous intéresse. Pour cela, exécutez le code Lua suivant à l'aide de `Mod1+F3` et indiquez le nom choisi (`emacsframe`) dans notre cas :

  ```
  mod_query.query_renameframe(_)
  ```

- Enfin, éditez votre fichier de configuration `cfg_ion.lua` et ajoutez les lignes suivantes :

  ```
  defwinprop {
    class = "Emacs",
    instance = "emacs",
    target = "emacsframe",
  }
  ```

4. Cet exemple est directement adapté de *A guided tour to Ion2.*

Ceci n'est qu'un tout petit aperçu de ce qu'il est possible de faire avec les *winprops*. Consulter *Configuring and extending Ion3 with Lua* pour plus d'informations.

8 Problèmes

L'utilisation de Ion peut parfois poser quelques problèmes avec certaines applications. Le gestionnaire de fenêtres n'est pas en cause, c'est plutôt la manière dont l'interface graphique de ces applications a été conçue qui serait à revoir, apparemment.

Parmi les applications qui peuvent poser problème, Adobe Reader semble assez bien placé. J'ai également remarqué que tsclient avait tendance à m'ouvrir des fenêtres d'erreur en arrière-plan que je ne repérais pas dans les premiers temps.

La plupart de ces problèmes sont liés à la manière dont ces applications gèrent les fenêtres transitoires (les boîtes de dialogue) : plutôt que de laisser le gestionnaire de fenêtres s'en occuper, elles préfèrent garder la main dessus, contrairement aux standards devant s'appliquer sous X[5].

Il existe des techniques assez sophistiquées pour résoudre ces difficultés. Pour faire très simple, en général il suffit de lancer l'application problématique en mode « bureau flottant », beaucoup plus proche du mode « traditionnel »de fonctionnement des gestionnaires de fenêtres, pour que tout rentre dans l'ordre ou du moins pour que l'application soit parfaitement utilisable. Pour des solutions plus compliquées mais plus satisfaisantes, consultez le document *Configuring and extending Ion3 with Lua*.

9 Ressources

- Site officiel de Ion
  http://iki.fi/tuomov/ion/
- FAQ de Ion
  http://modeemi.fi/~tuomov/ion/faq.html
- Document *Configuring and extending Ion3 with Lua*
  http://modeemi.fi/~tuomov/ion/doc-3/ionconf/
- Collection de scripts pour Ion3
  http://modeemi.fi/~tuomov/repos/ion-scripts-3/
- *A Guided Tour Of Ion2*
  http://modeemi.fi/~tuomov/ion/misc/guidedtour.html
- *The Ion Window Manager* de Matthieu Moy
  http://www-verimag.imag.fr/~moy/ion/
- *My Ion Page* de Thomas Nemeth
  http://tnemeth.free.fr/ion/
- *XSteve's Ion page*
  http://www.xsteve.at/prg/ion/

5. Ces standards, qui portent le doux nom d'ICCCM (*Inter-Client Communications Conventions Manual*) peuvent être consultés à l'adresse : http://tronche.com/gui/x/icccm/

10 À propos de ce document

Ce document est disponible aux formats HTML, PDF et code source LaTeX. L'adresse de référence est la suivante :

http://blog.nozav.org/docs/

Contributeur : Sylvain Abélard – sylvain.abelard(arobase)gmail.com.

N'hésitez pas à nous contacter pour tout commentaire ou suggestion.