


awesome

- Objet : Gestionnaire de fenêtres awesome
- Niveau requis :
[débutant, avisé](#)
- Commentaires : *Guide pas à pas sur awesome couvrant son installation, sa présentation, sa configuration et sa personnalisation.*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là ! 😊](#) - être à l'aise avec la programmation (langage script) - rudiments de base Lua.
- Suivi :
[à-tester](#)
 - Création par  [aramis_qc](#) le 24/01/2011
 - Testé par le
- Commentaires sur le forum : [C'est ici^{1\)}](#)

Introduction

awesome est un gestionnaire de fenêtres de la famille des « tiling windows ».

En jargon informatique, un gestionnaire de fenêtres de cette famille présente les affichages sans chevauchement.

awesome est quand même plus qu'un simple gestionnaire de fenêtres, les nombreuses possibilités de configuration qu'il offre, la puissance et la modularité de sa configuration en font un outil ayant des possibilités infinies.

Le revers de la médaille, cela représente un niveau de complexité plutôt élevé.

awesome se configure à l'aide de scripts écrits en langage Lua.

Il est donc requis de posséder quelques rudiments de ce langage.

Ne soyez pas inquiet pour autant. Ce guide vous aidera à comprendre Lua. En effet, ce tutoriel s'adresse aux gens qui n'ont jamais utilisé un tel gestionnaire de fenêtres et qui n'ont pas nécessairement de grandes habiletés de bidouilleur.

Ce guide vous fait découvrir et apprendre awesome selon une approche par étape.

1. Vous commencez avec son installation.
2. Par la suite vous l'explorez et vous apprenez à modifier le thème par défaut.
3. Vous poursuivez avec l'ajout de quelques applications essentielles et vous vous attaquez à sa personnalisation.

En suivant ce tutoriel, vous débutez avec un environnement de travail très basique pour finalement vous retrouver avec un bureau qui sera le plus fonctionnel possible.

Installation

Les paquets

Sur votre installation de Debian, ou sur une [Installation minimale](#):

```
apt-get update && apt-get install awesome awesome-extra
```

Si vous partez d'une installation sans environnement graphique, pensez à également installer le serveur X correctement :



```
apt-get install xorg mesa-utils
```

Vous devriez également installer un gestionnaire de session comme lightdm :

```
apt-get install lightdm
```

Lancement

Pour lancer **awesome**, vous pouvez le faire depuis votre [gestionnaire de session](#), si vous en avez un, après l'avoir relancé ou avoir redémarré votre Debian.



Pour relancer lightdm :

```
service lightdm restart
```

Si vous préférez lancer awesome directement depuis le tty, vous devrez créer un fichier `~/.xsession` contenant le code suivant :

`~/.xsession`



```
#!/bin/sh
exec awesome
```

Connectez-vous ensuite en tant qu'utilisateur (non root) dans une console et tapez :

```
startx
```

Premiers contacts

Probablement que vous trouverez votre premier contact avec **awesome** quelque peu rude, vous y remédieriez un peu plus loin.



Commencez déjà à vous *familiariser* avec ce nouvel environnement de travail et le gestionnaire de fenêtres.

Sachez que vous utilisez plutôt le clavier que la souris pour vous déplacer et pour manipuler les objets.

awesome n'ignore pas pour autant la souris et la supporte fort bien.

Si vous placez le curseur de la souris dans le coin supérieur gauche de l'écran et que vous cliquez sur le logo **awesome**, un petit menu s'affiche.

Il offre trois options dont un sous-menu *Debian*.

- Ce dernier propose quelques applications pré-installées. Inutile de vous attarder sur ce sous-menu, il est appelé à disparaître.
- L'option **open terminal** permet de lancer un terminal graphique rudimentaire.
- L'option **awesome** vous permet (entre autres) de quitter l'environnement graphique (quit).

Maintenant que vous savez que la souris reste disponible, voyez un peu comment mettre à profit le clavier.

Le terminal

La combinaison des touches `⌘ Win` + `↵ Enter` ouvre un [terminal](#).

Le terminal installé par défaut est **xterm**. Ce n'est pas un terminal avec la plus agréable des physionomies, aussi, profitez du fait qu'il soit maintenant ouvert pour installer un terminal plus... *sexy*.

Identifiez-vous comme [administrateur](#) et installez (par exemple) **xfce4-terminal** ainsi :

```
apt-get install xfce4-terminal
```

Fermez le terminal au clavier :

- par la combinaison `Ctrl` + `D` ou
- par la combinaison `⌘ Win` + `⇧ Shift` + `C`

Relancez le terminal avec la combinaison : `⌘ Win` + `↵ Enter`

Cette fois-ci, c'est le terminal que vous venez d'installer qui s'exécute.

En effet, lors de l'installation de **xfce4-terminal**, apt-get a exécuté la commande **update-alternatives** pour activer par défaut ce terminal.

Il y a une autre technique pour lancer une application, c'est en tapant la commande dans la barre de statut :

1. appuyez sur les touches `⌘ Win` + `R`. Le champ **Run:** apparaît alors dans la barre de statut,
2. et lancez à nouveau le terminal en saisissant :

```
xfce4-terminal
```

La fonctionnalité dite [autocompletion](#) est disponible dans ce champ.

Par exemple, tapez simplement :

```
xfce4-
```

et appuyez sur la touche `Tab ↵` du clavier, le nom de la commande

```
xfce4-terminal
```

sera alors automatiquement complété.

Puisque vous avez ouvert le terminal, attrapez l'occasion *au vol* pour installer votre navigateur Web fétiche, vous apprécierez sûrement ainsi de consulter ce wiki via un navigateur Web graphique plutôt qu'en utilisant un navigateur Web textuel ([links2](#)) ou semi-graphique (**links2 -g**).

```
apt-get install iceweasel iceweasel-l10n-fr
```

awesome supporte plusieurs autres combinaisons de touches, le [manuel](#) disponible en *français* les décrit. Consultez-le en tapant :

```
man awesome
```

Les étiquettes (tags)

awesome diffère des gestionnaires de fenêtres traditionnels auxquels vous ont habitué les bureaux tels GNOME, KDE, XFCE et autres. Il n'y a pas dans awesome le concept des bureaux virtuels. Ce sont plutôt des étiquettes, **tags** dans le jargon technique.

Les tags sont ces petits carrés numérotés que vous retrouvez dans la barre de statut. Chaque tag peut présenter une ou plusieurs fenêtres d'application. Selon le mode d'affichage individuellement appliqué aux tags, les fenêtres se chevauchent (floating) ou se positionnent dans l'espace disponible (tiling).

Il y a 9 tags par défaut. Vous vous déplacez d'un tag à l'autre à l'aide de la combinaison `⊞ Win` + **numéro du tag** ou la combinaison `⊞ Win` + `→` | `←`. Vous verrez plus loin comment personnaliser ces tags.

Le fichier rc.lua

La configuration de awesome pivote autour du fichier **rc.lua**. L'installation du gestionnaire de fenêtres a placé ce fichier dans le dossier `/etc/xdg/awesome`. Pour personnaliser votre environnement, il est préférable d'en faire une copie dans votre dossier de configuration personnel de awesome.

```
cp /etc/xdg/awesome/rc.lua ~/.config/awesome/rc.lua
```



Le fichier de configuration est écrit en langage de script Lua. Les lignes de commentaire débutent par deux tirets continus.

Premières modifications

Ici vous apportez de légères modifications à l'apparence de votre environnement de travail. Vous verrez plus loin comment personnaliser le plus possible votre bureau.

Changement de thème

awesome propose trois thèmes :

1. default,
2. sky,
3. zenburn.

Pour changer le thème, éditez le fichier `~/.config/awesome/rc.lua`. Remplacez "default" par un des thèmes précédemment cités, sur la ligne :

Extrait de `rc.lua`

```
beautiful.init("/usr/share/awesome/themes/default/theme.lua")
```

Ce tutoriel s'en tient pour l'instant au thème par défaut.

Modification de l'arrière-plan

Pour mettre l'arrière-plan de votre choix, il faut modifier le fichier `theme.lua` et modifier la ligne commençant par "theme.wallpaper_cmd" comme suit (pour l'exemple, l'image se situe dans `/home/user/Images/` et se nomme `images.jpg`):

Extrait de `theme.lua`

```
theme.wallpaper_cmd = { awsetbg /home/user/Images/images.jpg }
```

Attribution d'un nom aux tags

Pour une gestion parfaite des tags et des layouts à adopter, voici un exemple de code pouvant être utilisé (remplacez les noms et numéros de layouts par ce que vous voulez bien sûr).

Dans le fichier `rc.lua`

extrait rc.lua

```
-- {{{ Tags
tags = {
  names = { "musique", "messagerie", "www", "dossiers", "terminal",
6, 7, 8, 9 },
  layout = { layouts[1], layouts[2], layouts[1], layouts[5],
layouts[6], layouts[12], layouts[9], layouts[3], layouts[7] }
}
for s = 1, screen.count() do
  tags[s] = awful.tag(tags.names, s, tags.layout)
end
-- }}}}
```

Un Tag > Un logiciel... ou plus ;)

Par défaut, chaque application lancée le sera dans le tag courant, l'intérêt étant de pouvoir attirer un tag à chaque logiciel (ou l'inverse) et naviguer facilement entre toutes ces fenêtres sans se prendre la tête :)

Tout ceci se fait de façon assez simple dans le fichier rc.lua, dans lequel il y a déjà des exemples pour plus de facilité :)

extrait rc.lua

```
{ rule = { class = "Chromium" },
properties = { floating = true, tag = tags[1][1], swichtotag = true }
},
```

Décortiquons un peu tout ceci :

class = → pour connaître la classe d'un logiciel, il faut utiliser la ligne de commande, plus précisément 'xprop' qui va renvoyer tout un tas d'informations dont la 'class'. Pour utiliser 'xprop', lancer la commande 'xprop' dans un terminal utilisateur et cliquer sur une application ; cette action renverra les informations utiles (dont la classe qui vous servira) dans le terminal.

"Chromium" → la 'class' du logiciel chromium (logique non??)

properties = { → propriétés à appliquer au logiciel, floating → le logiciel sera en mode floating et non en tile, tag = tags[1][1] → le premier [1] désigne l'écran sur lequel sera affecté l'application, le deuxième [1] désigne le tag d'affectation, il est donc possible de le changer en mettant une valeur de 1 à 9 selon ses préférences, swichtotag = true → si vous vous trouvez sur le tag 5 par exemple, le lancement de chromium va vous basculer directement sur le tag 1.

Si vous souhaitez faire de même pour un logiciel sans interface graphique, il faudra dans ce cas utiliser 'name' au lieu de 'class'

par exemple :

[extrait rc.lua](#)

```
lua
{ rule = { name = "weechat-curses" },
properties = { tag = tags[1][3], switchtotag = true } },
```

Raccourcis clavier

awesome se contrôlant principalement avec le clavier, il est très utile d'utiliser des raccourcis claviers. le fichier rc.lua par défaut en contient déjà beaucoup qu'il peut être pratique de modifier selon ses goûts.

Le raccourci clavier (défini par défaut) : **Win** + **Ctrl** + **R** est pratique car il permet de relancer awesome après avoir modifié le fichier rc.lua.

pour passer d'un tag à un autre avec les **←** | **→** du clavier il suffit d'utiliser cette partie du fichier rc.lua

[extrait rc.lua](#)

```
-- {{{ Key bindings
globalkeys =
{
  key({ modkey,           }, "Left",  awful.tag.viewprev      ),
  key({ modkey,           }, "Right", awful.tag.viewnext      ),
  key({ modkey,           }, "Escape", awful.tag.history.restore),
```

cette partie sert à gérer les layouts :

[extrait rc.lua](#)

```
-- Layout manipulation
key({ modkey, "Shift" }, "j", function ()
awful.client.swap.byidx( 1) end),
key({ modkey, "Shift" }, "k", function ()
awful.client.swap.byidx(-1) end),
key({ modkey, "Control" }, "j", function () awful.screen.focus( 1)
end),
key({ modkey, "Control" }, "k", function () awful.screen.focus(-1)
end),
key({ modkey,           }, "u", awful.client.urgent.jumpto),
key({ modkey,           }, "Tab",
  function ()
    awful.client.focus.history.previous()
    if client.focus then
      client.focus:raise()
    end
  end),
```

```

-- Standard program
key({ modkey,          }, "Return", function ()
awful.util.spawn(terminal) end),
key({ modkey, "Control" }, "r", awesome.restart),
key({ modkey, "Shift"   }, "q", awesome.quit),

key({ modkey,          }, "l",      function ()
awful.tag.incmwfact( 0.05) end),
key({ modkey,          }, "h",      function ()
awful.tag.incmwfact(-0.05) end),
key({ modkey, "Shift"   }, "h",      function ()
awful.tag.incmaster( 1)   end),
key({ modkey, "Shift"   }, "l",      function ()
awful.tag.incmaster(-1)  end),
key({ modkey, "Control" }, "h",      function () awful.tag.incncol(
1)          end),
key({ modkey, "Control" }, "l",      function ()
awful.tag.incncol(-1)    end),
key({ modkey,          }, "space", function ()
awful.layout.inc(layouts, 1) end),
key({ modkey, "Shift"   }, "space", function ()
awful.layout.inc(layouts, -1) end),

```

les prompts pour lancer des applications ou du code lua :

[extrait rc.lua](#)

```

-- Prompt
key({ modkey }, "F1",
function ()
awful.prompt.run({ prompt = "Run: " },
mypromptbox[mouse.screen],
awful.util.spawn, awful.completion.bash,
awful.util.getdir("cache") .. "/history")
end),

key({ modkey }, "F4",
function ()
awful.prompt.run({ prompt = "Run Lua code: " },
mypromptbox[mouse.screen],
awful.util.eval, awful.prompt.bash,
awful.util.getdir("cache") .. "/history_eval")
end),
}

```


Fenêtres

Par défaut, les fenêtres d'application ne comportent pas de boutons maximiser/diminuer/quitter étant donné qu'awesome est surtout conçu pour être géré au clavier, néanmoins il est très facile d'en ajouter.

Il suffit de décommenter la ligne suivante qui se trouve dans la section "Signals" du fichier rc.lua par défaut :

[Extrait de rc.lua](#)

```
awful.titlebar.add(c, { modkey = modkey })
```

Pour décommenter, il faut enlever les deux tirets qui se trouvent en début de ligne.

Signification des boutons par défaut :

- La flèche : fenêtre flottante. La fenêtre ignorera la configuration définie, elle pourra être redimensionnée librement.
- La fusée : fenêtre agrandie. La fenêtre ignorera également la configuration définie pour utiliser tout l'espace disponible.
- Le plus : fenêtre collante. La fenêtre sera visible sur tous les tags.
- L'étoile : fenêtre au dessus. La fenêtre ignorera l'ordre d'empilement normal et sera affichée au-dessus de tout le reste.
- La croix : bouton de fermeture. Ferme la fenêtre...

Démarrage

Pour lancer une application dès le démarrage, il existe plusieurs méthodes.

Lancer une application via awesome

Celle-ci fait partie des plus simples :) L'inconvénient majeur étant que si l'on décide de redémarrer awesome (sans fermer sa session) les applications seront relancées.

Pour ce faire, il suffit d'ajouter un bout de code lua tel que celui-ci dans le fichier rc.lua.

exemple :

[extrait rc.lua](#)

```
lua  
os.execute("chromium &")
```

et c'est tout, au démarrage d'awesome chromium sera lancé automatiquement selon la configuration faite dans la partie "Un Tag > Un logiciel...ou plus"

Référence : [wiki awesome \(fr\)](#)

Lancer une application via .xsession

La méthode la plus standard pour définir la liste des applications à exécuter au démarrage est de créer un fichier .xsession. Ce dernier doit contenir l'ensemble des commandes à lancer, sans oublier... le window manager.

Voici un exemple de .xsession :

[.xsession](#)

```
#!/bin/sh

setxkbmap fr
xmodmap $HOME/.xmodmaprc
xrdb $HOME/.Xdefaults
xcompmgr -FfCc &

exec awesome
```

Premier widget

1) Ajouter en haut du fichier rc.lua (après "require("naughty")" par exemple) le code suivant :

[extrait rc.lua](#)

```
-- Widgets library
require("wicked")
```

2) Ajouter dans la section Wibox (après "textclock widget" par exemple) le code suivant :

[extrait rc.lua](#)

```
-- graphe d'utilisation du processeur
cpugraphwidget = widget({
  type = 'graph',
  name = 'cpugraphwidget',
  align = 'right'
})

cpugraphwidget.height = 0.85
cpugraphwidget.width = 45
cpugraphwidget.bg = '#333333'
cpugraphwidget.border_color = '#0a0a0a'
cpugraphwidget.grow = 'left'
```

```
cpugraphwidget:plot_properties_set('cpu', {
  fg = '#AEC6D8',
  fg_center = '#285577',
  fg_end = '#285577',
  vertical_gradient = false
})

wicked.register(cpugraphwidget, wicked.widgets.cpu, '$1', 1, 'cpu')
```

3) Enfin ajouter le widget à la barre de statut (avec le nom du widget : cpugraphwidget) :

extrait rc.lua

```
-- Add widgets to the wibox - order matters
mywibox[s].widgets = {
  {
    mylauncher,
    mytaglist[s],
    mypromptbox[s],
    layout = awful.widget.layout.horizontal.leftright
  },
  mylayoutbox[s],
  mytextclock,
  cpugraphwidget,
  s == 1 and mysystray or nil,
  mytasklist[s],
  layout = awful.widget.layout.horizontal.rightright
}
```

Des exemples de widgets sont disponibles ici :

http://awesome.naquadah.org/wiki/Wicked/fr#Mettre_en_place_Wicked

Exemples de fichiers de configuration

Outre les exemples présents sur le wiki awesome, certains utilisateurs du wiki ont fourni leurs fichiers de configuration awesome :

- [captfnfab](#)

Liens vers le forum

- [\[résolu\] Awesome WM - Multi écran navigation dynamique](#)

Liens utiles

- <http://forum.malekal.com/awesome-window-manager-t21608.html>
- http://wiki.chezlefab.net/tuto_nix/awesome/accueil
- https://awesome.naquadah.org/wiki/Main_Page/fr

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:environnements:x11:awesome>



Last update: **24/03/2017 14:09**