spectrwm

spectr/scrotwm est un gestionnaire de fenêtres de type tiling inspiré par xmonad et dwm. spectrwm a été écrit par des hackers pour des hackers et tente de rester rapide, léger et compact.
spectrwm est en constant développement et demande encore quelques améliorations, mais ça ne l'empêche pas d'être un gestionnaire de fenêtres très rapide et hautement configurable.

spécificités

- support du multi écran grâce à xrandr & xinerama
- navigation à la souris ou au clavier
- status bar personnalisable
- fichier de configuration en mode texte
- rechargement de la configuration sans quitter la wm (on-the-fly-reloading)
- lanceur de menu (dmenu)
- disposition tile/bottomstack/fullscreen
- manipulation des clients master/stack/screen facilité
- support des clients libres
- définition possible de “régions” sur l'écran
- personnalisation des raccourcis et lanceurs
- utilisation des “quirks”
- support multi-OS(*BSD, Linux, OSX, Windows/cygwin)

installation

spectrwm est présent dans les dépôts Debian sous le nom **scrotwm**, ou sur les sources git.

- installation depuis les dépôts Squeeze:# apt-get install scrotwm
- installation depuis les dépôts Wheezy/Sid:# apt-get install spectrwm
- installation depuis les sources git (install buggée sur Debian Squeeze due à une absence de librairie libxcb-util):$ git clone git://opensource.conformal.com/spectrwm.git
  $ cd spectrwm/linux
  $ make
  # make install

lancement

scrot se lance simplement depuis votre ~/.xinitrc grâce à:

```
#launch window manager
exec scrotwm
```

configuration

la configuration de spectr/scrotwm passe par l'édition de son fichier de configuration ~/spectrwm.conf ou ~/.scrotwm.conf selon la version installée.
on commence par récupérer le fichier de conf donné en exemple:

```
$ cp /etc/scrotwm.conf ~/.scrotwm.conf
ou
$ cp ~/spectrwm/spectrwm.conf ~/.spectrwm.conf
```

vous pouvez aussi récupérer le fichier de raccourcis adapté à votre clavier (Wheezy/Sid/Sources):

```
$ cp /usr/share/spectrwm/examples/scrotwm_fr.conf ~/.scrotwm_fr.conf
ou
$ cp ~/spectrwm/spectrwm_fr.conf ~/.spectrwm_fr.conf
```

la syntaxe du fichier de conf est du type <keyword> = <setting>. ex

```
color_focus = red
```

les lignes précédées d'un ”#” sont des commentaires. si vous désirez afficher un ”#”, vous devez inscrire “\#”.

options de configuration

note : les options marquées d'un “*” ne sont disponibles que si vous installez depuis les sources. reportez-vous au manuel de votre version pour connaître les options disponibles.

option	description
*autorun	lance une application au démarrage. format ws[<idx>]:application, ex: ws[2]:xterm lance xterm dans l'espace de travail 2.
bar_action	script externe à lancer pour un affichage dans la status bar.
*bar_at_bottom	place la status bar en bas de l'écran.
bar_border[x]	couleur de la bordure de la status bar sur l'écran x.
*bar_border_unfocus[x]	couleur de la bordure de la status bar de la région en arrière plan sur l'écran x.
*bar_border_width	épaisseur de la bordure de la status bar. pas de bordure si 0.
bar_color[x]	couleur de fond de la status bar sur l'écran x.
bar_enabled	afficher la barre.
*bar_enabled_ws[x]	afficher la barre sur l'espace de travail x; par defaut 1.

bar_font	police utilisée pour la status bar. format Xft ou X Logical Font Description(XLFD). vous pouvez spécifier plusieurs polices séparées par ",". si une des polices est en XFT, elle sera utilisée par défaut. notez que dmenu n'utilise pas les polices XFT. ex: bar_font = -*-profont-medium-*-*-*-11-*-*-*-*-*-*-*,Terminus:pixelsize=14,-*-clean-medium-*-*-*-12-*-*-*-*-*-*-*.
bar_font_color[x]	couleur de la police de la status bar pour l'écran x.
*bar_format	définit le format de la status bar en écrasant toutes les autres options. le format utilise la syntaxe strftime(3). séquences disponibles en fin de tableau.
*bar_justify	alignement du texte de la status bar. valeurs: left, center, et right.
bind[x]	associer une touche au programme x. voir la section raccourcis plus bas.
*border_width	épaisseur de la bordure des clients. pas de bordures si 0.
clock_enabled	affiche l'heure dans la status bar.
color_focus	couleur de la bordure du client ayant le focus.
color_unfocus	couleur de la bordure des clients en arrière plan.
dialog_ratio	redimensionnement des fenêtres type dialog. ex: 0.6 = 60% de l'écran.
*disable_border	n'affiche pas la bordure si la status bar est masquée et qu'il n'y a qu'un seul client.
*focus_close	dirige le focus à la fermeture d'un client. valeurs: first, next, previous (défaut) et last.
*focus_close_wrap	donne le focus au dernier clients lors de la fermeture d'un client.
*focus_default	client ayant le focus par défaut. valeurs: first et last (défaut).
*focus_mode	mode du focus. valeurs: default, follow et manual.
*keyboard_mapping	charge une disposition de clavier. voir la section KEYBOARD MAPPING FILES plus bas.
*layout	sélection de la disposition par défaut. valeurs: vertical, vertical_flip, horizontal, horizontal_flip et fullscreen.
modkey	touche de modification. Mod1=ALT, Mod4=windows.
program[p]	définir une action "p" à lancer. voir la section PROGRAMS plus bas.
quirk[c:n]	ajoute une règle "quirk" pour le client de class 'c' et nommé 'n'. voir la section QUIRKS plus bas.
region	définir des régions sur l'écran, en écrasant toute auto-détection. format: screen[<idx>]:WIDTHxHEIGHT+X+Y, ex: screen[1]:800×1200+0+0. vous pouvez aussi couvrir deux écrans: screen[1]:2048×768+0+0.
*region_padding	espace libre entre les régions.
*spawn_position	placement des nouveaux clients. valeurs: first, next, previous et last (default).
stack_enabled	afficher la disposition dans la status bar.
term_width	définir une taille moinimale pour xterm (le seul supporté pour le moment).
*tile_gap	espace libre entre les clients.
title_class_enabled	afficher la class du client dans la status bar
title_name_enabled	afficher le titre du client dans la status bar.
*urgent_enabled	notification des clients urgents.
*verbose_layout	précise l'affiche de la disposition dans la status bar.
*window_name_enabled	affiche le nom du client, limité à 64 caractères.
*workspace_limit	définir le nombre d'espace de travfail disponible. Minimum=1, maximum=22, défaut=10.

séquences de la barre

uniquement depuis les sources.

Character sequence	Replaced with
+<	Pad with a space
+A	résultat d'un script externe
+C	'Window class' du client
+D	nom de l'espace de travail
+F	indicateur de client libre
+I	index des espaces de travail
+N	numéro de l'écran
+P	'Window class' et titre du client séparés par ":"

+S	disposition
+T	titre du client
+U	indicateur d'application urgente
+V	version du programme
+W	nom du client
++	un '+'

toutes les séquences sont limitées en nombre de caractères, par exemple +64A.

keyboard mapping files

uniquement sur Wheezy/Sources
Keyboard mapping files sont des fichiers de configuration reprenant les raccourcis claviers adaptés à votre disposition. ces fichiers peuvent être chargés dans spectrwm.conf avec l'option 'keyboard_mapping". ces fichiers sont situés dans les sources ou, si vous avez installé spectrwm depuis Deban Wheezy/Sid, dans /usr/share/doc/spectrwm/examples/.

```
spectrwm_cz.conf     Czech Republic keyboard layout
spectrwm_es.conf     Spanish keyboard layout
spectrwm_fr.conf     French keyboard layout
spectrwm_fr_ch.conf  Swiss French keyboard layout
spectrwm_se.conf     Swedish keyboard layout
spectrwm_us.conf     United States keyboard layout
```

quirks aka règles

spectrwm peut définir des "quirks" ou règles à certains clients selon leur class et leur titre. voici les règles par défaut:

```
Firefox-bin:firefox-bin                 TRANSSZ
Firefox:Dialog                          FLOAT
Gimp:gimp                               FLOAT + ANYWHERE
MPlayer:xv                              FLOAT + FULLSCREEN + FOCUSPREV
OpenOffice.org 2.4:VCLSalFrame          FLOAT
OpenOffice.org 3.1:VCLSalFrame          FLOAT
pcb:pcb                                 FLOAT
xine:Xine Window                        FLOAT + ANYWHERE
xine:xine Panel                         FLOAT + ANYWHERE
xine:xine Video Fullscreen Window       FULLSCREEN + FLOAT
Xitk:Xitk Combo                         FLOAT + ANYWHERE
Xitk:Xine Window                        FLOAT + ANYWHERE
XTerm:xterm                             XTERM_FONTADJ
```

syntaxe des règles :

- FLOAT : client libre.

- TRANSSZ : ajuste la taille du client en fonction du **ratio** (à définir dans le fichier de configuration).
- ANYWHERE : autorise le client à se positionner où il veut, non centré.
- XTERM_FONTADJ : ajuste la police xterm lors du redimensionnement (seulement avec xterm).
- FULLSCREEN : remplir la totalité de la 'région' sans bordures.
- FOCUSPREV : lors de la fermeture du client, force le focus sur le dernier client ayant eu le focus.

vous pouvez ajouter ou enlever des règles avec la syntaxe:

```
quirk[<class>:<name>] = <quirk> [+ <quirk> ...]
```

- <class> et <name> définissent le client auquel s'appliqué la règle (depuis xprop),
- <quirk> est l'une des règles décrites ci-dessus.

par exemple:

```
quirk[MPlayer:xv] = FLOAT + FULLSCREEN + FOCUSPREV
quirk[pcb:pcb] = NONE  # enlève la règle par défaut
```

spectrwm attribue également des règles automatiques pour certains clients en se basant sur les propriétés _NET_WM_WINDOW_TYPE*:

```
_NET_WM_WINDOW_TYPE_DOCK              FLOAT + ANYWHERE
_NET_WM_WINDOW_TYPE_TOOLBAR           FLOAT + ANYWHERE
_NET_WM_WINDOW_TYPE_UTILITY           FLOAT + ANYWHERE
_NET_WM_WINDOW_TYPE_SPLASH            FLOAT
_NET_WM_WINDOW_TYPE_DIALOG            FLOAT
```

note les règles spécifiés dans le ~/.spectrwm.conf écrasent les règles par défaut.

normes EWMH

spectrwm respecte partiellement les normes EWMH (Extended Window Manager Hints). vous pouvez utiliser wmctrl(1) et xdotool(1) pour envoyer des messages sur certains clients.
identification du client avec _NET_ACTIVE_WINDOW, ce qui permet d'afficher le nom de ce client grâce à xprop(1) et grep(1):

```
$ WINDOWID=`xprop -root _NET_ACTIVE_WINDOW | grep -o "0x.*"`
$ xprop -id $WINDOWID WM_NAME | grep -o "\".*\""
```

un client peut **obtenir le focus** en envoyant un message de type _NET_ACTIVE_WINDOW. par exemple, avec wmctrl(1) pour lancer le message (avec ID client = 0x4a0000b):

```
$ wmctrl -i -a 0x4a0000b
```

fermeture d'un client avec un message de type _NET_CLOSE_WINDOW :

```
$ wmctrl -i -c 0x4a0000b
```

libère ou tile le client en ajoutant/enlevant _NET_WM_STATE_ABOVE du _NET_WM_STATE. on envoi un message du type _NET_WM_STATE. code pour libérer/attacher un client:

```
$ wmctrl -i -r 0x4a0000b -b toggle,_NET_WM_STATE_ABOVE
```

déplacement/redimensionnement d'un client libre:

```
$ wmctrl -i -r 0x4a0000b -e 0,100,50,640,480
```

déplace le client (100,50) et le redimensionne (640×480). les messages de type _NET_MOVERESIZE_WINDOW sont ignorés par les clients attachés.

raccourcis clavier

programs

spectrwm vous permet de spécifier des séquences d'actions à associer à des combinaisons de touches.
les programmes par défaut:

```
term             xterm
screenshot_all   screenshot.sh full
screenshot_wind  screenshot.sh window
lock             xlock
initscr          initscreen.sh
menu             dmenu_run -fn $bar_font -nb $bar_color -nf $bar_font_color
-sb $bar_border -sf $bar_color
```

synatxe des programmes:

```
program[<name>] = <progpath> [<arg> [... <arg>]]
```

avec:

- <name> identifier le prgramme, prenez garde à ne pas donner un nom existant dans les fonctions de spectrwm
- <progpath> le programme à lancer
- <arg> =0 ou les arguments à passer au programmer

les variables suivantes seront reconnues dans les programmes à lancer:

- $bar_border
- $bar_color
- $bar_font
- $bar_font_color
- $color_focus

- $color_unfocus
- $region_index
- $workspace_index

Exemple:

```
program[ff] = /usr/local/bin/firefox http://spectrwm.org/
bind[ff] = Mod+Shift+b # Mod+Shift+B lance firefox
```

pour effacer le programme précédent:

```
bind[] = Mod+Shift+b
program[ff] =
```

bindings

M=Alt, S=Shift, M1=bouton1, M2=bouton2, M3=bouton3

action de la souris	description
M1	donne le focus
M-M1	déplace le client
M-M3	redimensionne le client
M-S-M3	redimensionne le client depuis le centre

action du clavier	program/fonction	description
M-S-<Return>	term	ouvre le programme définit par 'term'
M-p	menu	lance la commande de 'menu'
M-S-q	quit	quitte spectrwm
M-q	restart	relance spectrwm
M-<Space>	cycle_layout	change la disposition
M-S-<\>	flip_layout	échange la zone master avec le stack
M-S-<Space>	stack_reset	rétablit la proportion stack/master
M-h	master_shrink	réduit la taille du master
M-l	master_grow	agrandit la taille du master
M-,	master_add	ajoute un client dans le master
M-.	master_del	enlève un client du master
M-S-,	stack_inc	ajoute un client dans le stack
M-S-.	stack_dec	enlève un client du stack
M-<Return>	swap_main	échange le client avec le master
M-j, M-<TAB>	focus_next	donne le focus au client suivant
M-k, M-S-<TAB>	focus_prev	donne le focus au client précédent
M-m	focus_main	donne le focus au master
M-S-j	swap_next	échange le client avec le client suivant
M-S-k	swap_prev	échange le client avec le client précédent
M-b	bar_toggle	affiche/masque la barre
M-S-b	bar_toggle_ws	affiche/masque la bar sur l'espace de travail courant
M-x	wind_del	ferme le client

M-S-x	wind_kill	tue le client
M-<1-9,0,F1-F12>	ws_<1-22>	rejoint l'espace de travail <1-22>
M-S-<1-9,0,F1-F12>	mvws_<1-22>	envoi le client sur l'espace de travail <1-22>
M-<Keypad 1-9>	rg_<1-9>	rejoint la region <1-9>
M-S-<Keypad 1-9>	mvrg_<1-9>	envoi le client dans la region <1-9>
M-<Right>	ws_next	rejoint l'espace de travail occupé suivant
M-<Left>	ws_prev	rejoint l'espace de travail occupé précédent
M-<Up>	ws_next_all	rejoint l'espace de travail suivant
M-<Down>	ws_prev_all	rejoint l'espace de travail précédent
M-a	ws_prior	rejoint le dernier espace de travail visité
M-S-<Right>	rg_next	rejoint la region suivante
M-S-<Left>	rg_prev	rejoint la region précédente
M-s	screenshot_all	capture d'écran
M-S-s	screenshot_wind	capture du client
M-S-v	version	affiche la version dans la barre
M-t	float_toggle	libère/attache le client
M-S-<Delete>	lock	vérouille l'écran
M-S-i	initscr	script défini par 'initscr'
M-w	iconify	minimise
M-S-w	uniconify	restaure
M-S-r	always_raise	permettre aux clients attaché de passer au-dessus des clients libres
M-v	button2	simule un clic central
M–	width_shrink	réduit la largeur du client libre
M-=	width_grow	augmente la largeur du client libre
M-S–	height_shrink	réduit la hauteur du client libre
M-S-=	height_grow	augmente la hauteur du client libre
M-[	move_left	déplace le client libre sur la gauche
M-]	move_right	déplace le client libre sur la droite
M-S-[	move_up	déplace le client libre vers le haut
M-S-]	move_down	déplace le client libre vers le bas
M-S-/	name_workspace	nomme l'espace de travail courant
M-/	search_workspace	recherche un espace de travail
M-f	search_win	recherche un client dans l'espace de travail courant

syntaxe des raccourcis:

```
bind[<action>] = <keys>
```

- <action> une des actions décrites ci-dessus (ou vide pour effacer une règle)
- <keys> une touche de modification (MOD, Mod1, Shift, etc.) et une ou plusieurs touches (b, space, etc.), séparés par un ”+”.

exemple:

```
bind[reset] = Mod4+q # associer la touche 'windows + q' à l'action 'reset'
bind[] = Mod1+q # effacer la règle précédente
```

pour spécifier la touche de modification par défaut, utiliser 'MOD'. plusieurs raccourcis peuvent être associés à la même action.
pour associer les caractères spéciaux, utiliser **xev** afin d'identifier la touche.