# echinuswm

**Echinus** a été au départ un simple fork de dwm, ayant pour but de simplifier la syntaxe de configuration en ajoutant un ~echinusrc au format ~/.Xdefaults. Mais il est maintenant un gestionnaire de fenêtres à part entière. codé par Alexander Polakov, il propose la gestion des fenêtres en tiling/floating dans la ligné de dwm. le **tiling** organise les fenêtres ou **clients** en deux zones; le **master** et le **stack** ou **tile**. le client “principal” occupe le master tandis que les autre clients s'empilent dans le stack selon une disposition ou **layout** prédéfinie. echinus ne dispose pas de panel, menu ou autre gadgets (f)utiles (il faudra les installer séparement).

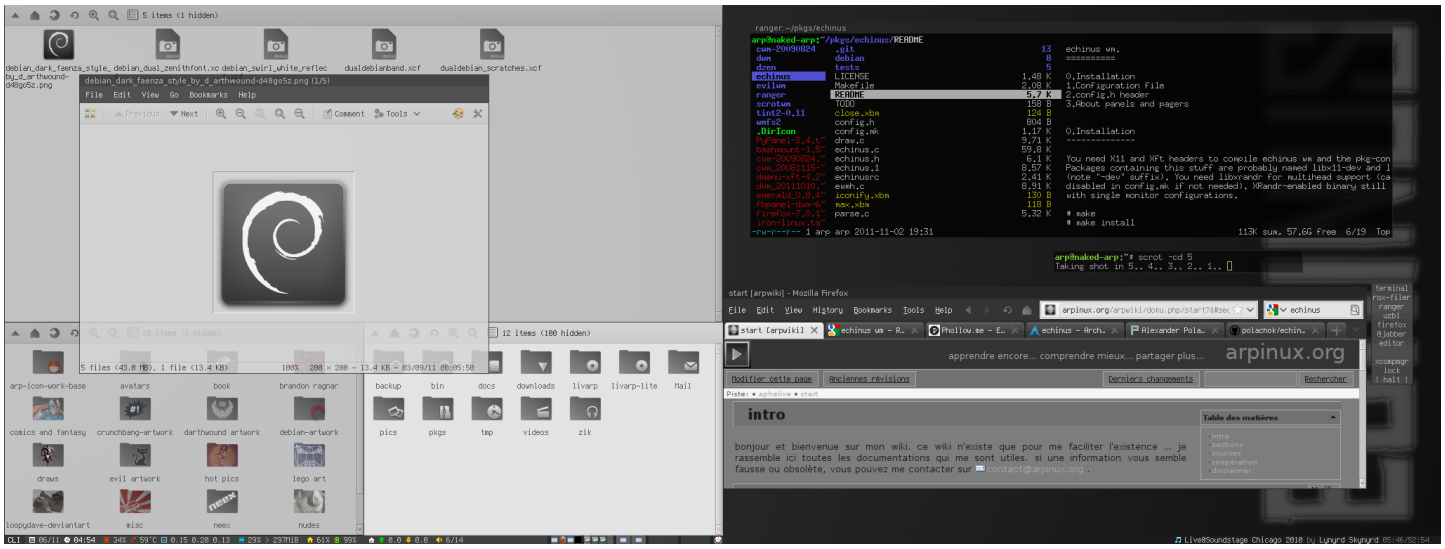
voici ses principales spécificités:

- syntaxe du fichier de config très simple au format ~/.Xresources, ~/.Xdefaults
- support des normes EWMH
- barre de titres personnalisables
- support des polices Xft
- support du multi-écrans avec XRandr

si vous désirez tester echinus en live, allez faire un tour du côté du livarp ;) la version 0.3 a une session echinuswm préconfigurée et documentée...

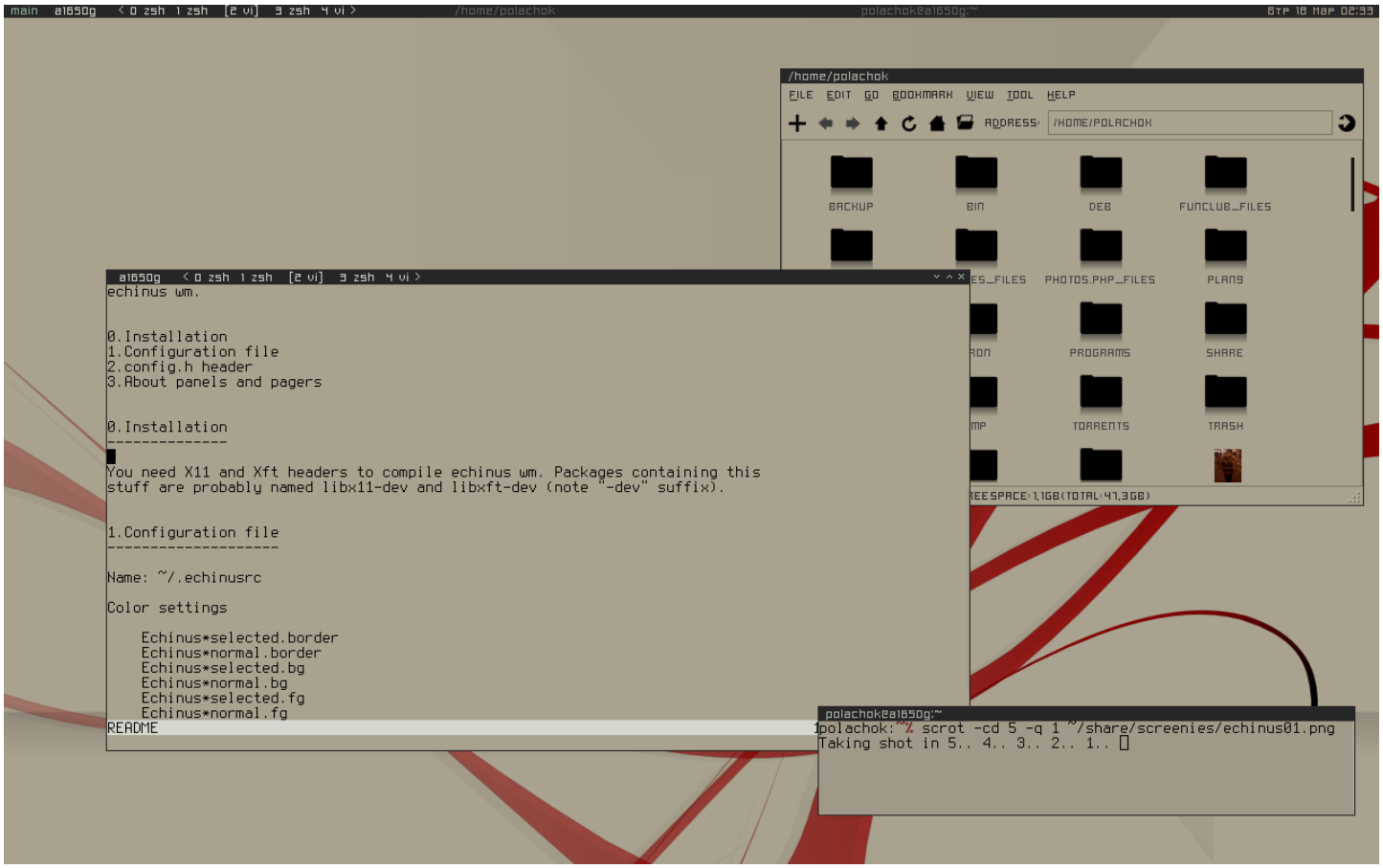
## Sommaire

# screenshots

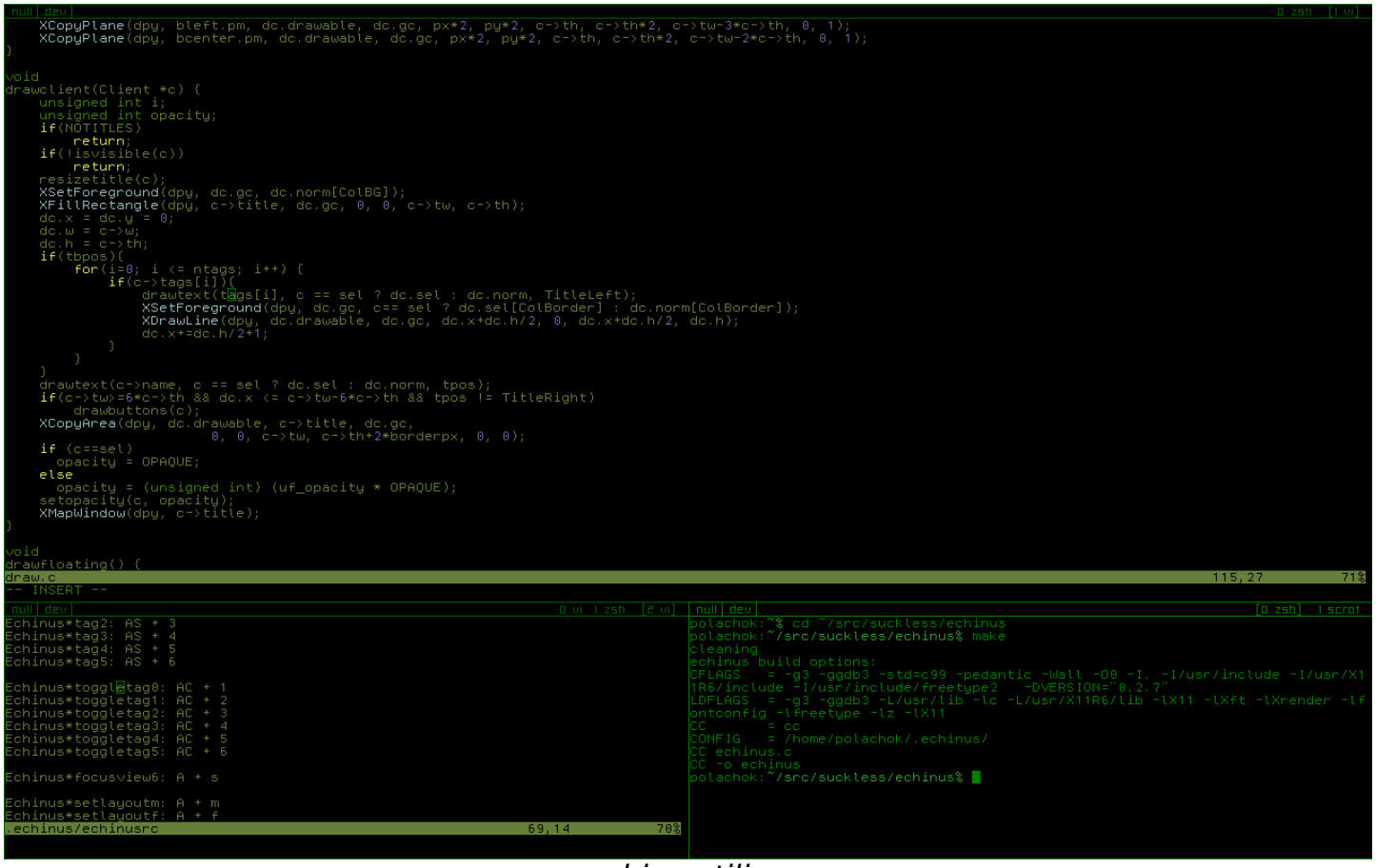


*livarp-echinus*

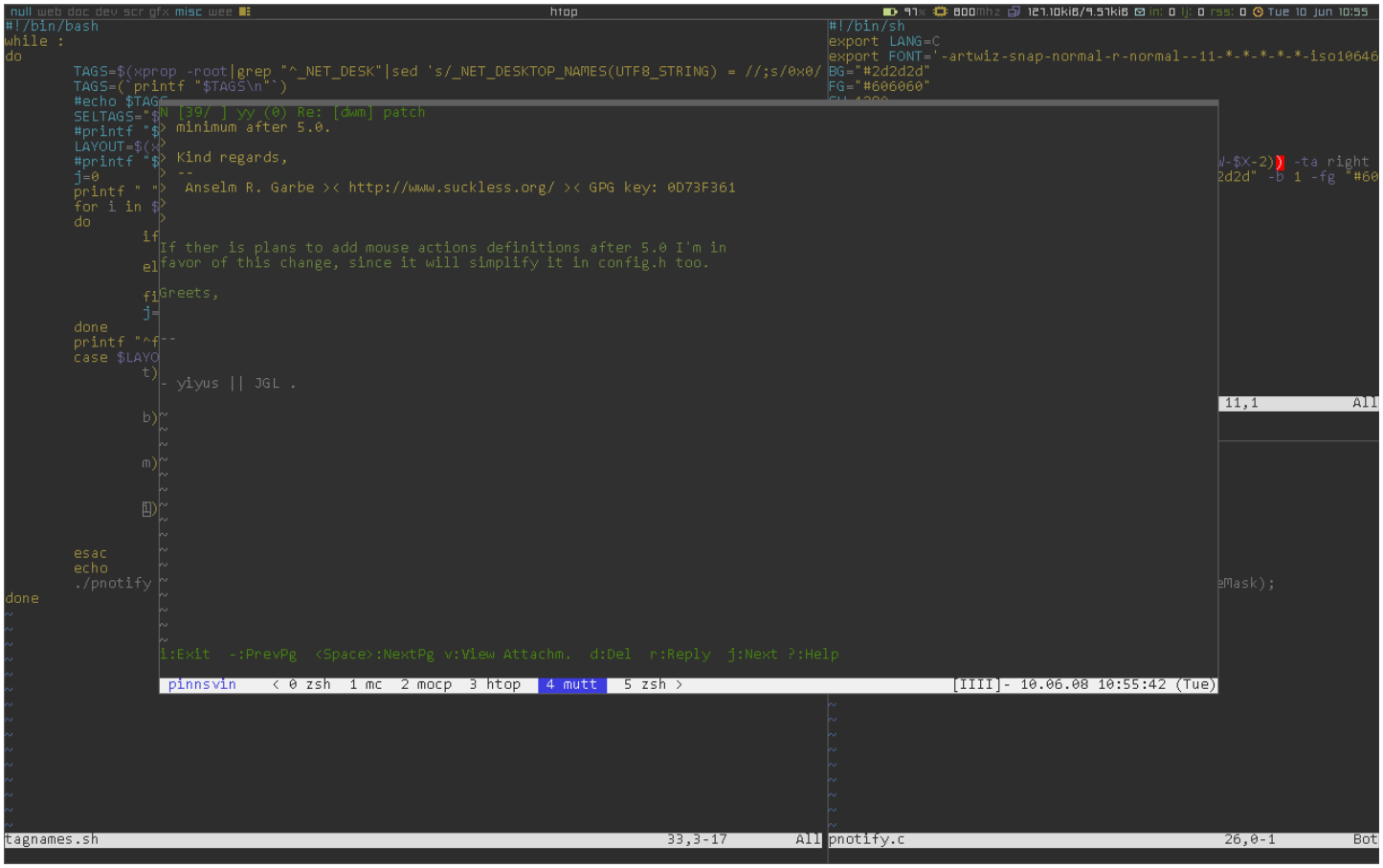
histoire de voir à quoi echinus peut ressembler...




*echinus floating*



*echinus tiling*



*echinus mixte*

# installation

**echinus** est disponible dans les dépôts 'contrib' de plusieurs distributions. cependant, le dev est plutôt actif, je vous conseille donc l'installation par les dépôts **git**.

- installation des dépendances:

```
# apt-get install build-essential libx11-dev libxft-dev libxrandr-dev pkg-
config
```

- puis on récupère les sources depuis git:

```
$ git clone git://github.com/polachok/echinus.git
```

- on installe depuis les sources:

```
$ cd echinus/
$ make
# make install
```

- on installe le fichier de configuration:

```
$ mkdir ~/.echinus
$ cp echinusrc ~/.echinus/echinusrc
```

*echinus installe ses fichiers dans /usr/local/**

# configuration

**echinus** se configure depuis son ~/.echinus/echinusrc. ce fichier est au format ~/.Xdefaults. je vous en livre un exemplaire commenté

```
! ----------------------
! file: ~/.echinus/echinusrc
! fichier de configuration de echinuswm
! vim:fenc=utf-8:nu:ai:si:et:ts=4:sw=4:ft=xdefaults:
! ----------------------------------------------------

! couleur de la bordure de la fenêtre ayant le focus
Echinus*selected.border: #262626
! couleur des boutons de la fenêtre ayant le focus
Echinus*selected.button: #d3d7cf
! couleur de la barre de titre de la fenêtre ayant le focus
Echinus*selected.bg: #262626
! couleur du texte de la barre de titre de la fenêtre ayant le focus
Echinus*selected.fg: #d3d7cf

! couleur de la bordure
Echinus*normal.border: #262626
! couleur des boutons
Echinus*normal.button: #262626
! couleur de la barre de titre
Echinus*normal.bg: #262626
! couleur du texte de la barre de titre
Echinus*normal.fg: #b0b4ac

! taille de la bordure en px
Echinus*border: 1
```

```
! chemin des boutons - à placer dans votre ~/.echinus/
Echinus*button.iconify.pixmap: iconify.xbm
Echinus*button.maximize.pixmap: max.xbm
Echinus*button.close.pixmap: close.xbm

! comportement du pointeur:
! options: 0:click to focus, 1:focus-follow pour les clients floating, 2:focus-
follow, 3:focus-follow-raise
Echinus*sloppy: 0
! opacité des fenêtres. la fenêtre ayant le focus est opaque (1)
! vous aurez besoin d'un gestionnaire de composite tel que xcompmgr
Echinus*opacity: 0.8
! barre de titre sur les fenêtres libres (0/1)
Echinus*decoratetiled: 0
! chevaucher les batards... panels, widgets, docks (0/1)
Echinus*hidebastards: 0
! place occupée par le master en mode tiling (0.5=moitié de l'écran)
Echinus*mwfact: 0.6
! nombre de clients dans la zone master
Echinus*nmaster: 1

! hauteur de la barre de titre en px
Echinus*title: 12
! police par défaut
Echinus*font: fixed-9
! touche de modification: A:Alt, W:Super, S:Shift, C:Control
Echinus*modkey: A

! layout par défaut:
! options: i:ifloating, f:floating, t:tiled, b:bottomstack, m:maximized
! ifloating vs floating: f=place le client au coin supérieur gauche de l'écran i=
placement intelligent
Echinus*deflayout: i

! nombre total de tags (bureaux virtuels)
Echinus*tags.number: 7
! nom des tags
Echinus*tags.name0: main
Echinus*tags.name1: web
Echinus*tags.name2: doc
Echinus*tags.name3: dev
Echinus*tags.name4: scr
Echinus*tags.name5: gfx
Echinus*tags.name6: misc

! layout par défaut des tags
Echinus*tags.layout1: m
Echinus*tags.layout3: b

! masquer/afficher les panels, pager, widgets
Echinus*togglestruts: A + b
! passer d'un écran à l'autre
Echinus*togglemonitor: A + grave
! donner le focus au client précédent/suivant
Echinus*focusnext: A + j
Echinus*focusprev: A + k
! voir le tag précédent
```

```
Echinus*viewprevtag: A + Tab
! voir le tag de gauche/droite
Echinus*viewlefttag: AS + Left
Echinus*viewrighttag: AS + Right
! quitter echinus
Echinus*quit: CA + q
! recharger echinus: relis le fichier ~/.echinus/echinusrc
Echinus*restart: AS + q = echinus
! fermer le client
Echinus*killclient: AS + c
! libère/tile le client
Echinus*togglefloating: A + space
! échange le client avec le client du master
Echinus*zoom: A + Return

! voir le tag 'n': aller au bureau 'n'
Echinus*view0: A + F1
Echinus*view1: A + F2
Echinus*view2: A + F3
Echinus*view3: A + F4
Echinus*view4: A + F5
Echinus*view5: A + F6

! tag 'n' le client: déplace le client sur le bureau 'n'
Echinus*tag0: AS + 1
Echinus*tag1: AS + 2
Echinus*tag2: AS + 3
Echinus*tag3: AS + 4
Echinus*tag4: AS + 5
Echinus*tag5: AS + 6

! ??
Echinus*toggletag0: CAS + 1
Echinus*toggletag1: CAS + 2
Echinus*toggletag2: CAS + 3
Echinus*toggletag3: CAS + 4
Echinus*toggletag4: CAS + 5
Echinus*toggletag5: CAS + 6

! échange le tag courant avec le tag 'n'
Echinus*toggleview0: CA + 1
Echinus*toggleview1: CA + 1
Echinus*toggleview2: CA + 1
Echinus*toggleview3: CA + 1
Echinus*toggleview4: CA + 1
Echinus*toggleview5: CA + 1

! échange le tag courant avec le tag 'n' et y affiche le client
Echinus*focusview6: A + s

! definir le layout
Echinus*setlayoutm: A + m
Echinus*setlayoutf: A + f
Echinus*setlayouti: A + i
Echinus*setlayoutt: A + r
Echinus*setlayoutb: A + w

! deplacement/redimensionnement des clients selon x,y,w et h
```

```
! x et y: adresse du coin supérieur gauche du client
! w et h: largeur et hauteur du client
! deplace le client de 5px vers la droite/gauche/haut/bas
Echinus*moveright: A + d = 5
Echinus*moveleft: A + a = -5
Echinus*moveup: A + w = 0 -5
Echinus*movedown: A + s = 0 5

! reduit le client de 5px
Echinus*resizedecx: AS + a = 0 0 -5 0
Echinus*resizedecy: AS + s = 0 0 0 -5
! agrandit le client de 5px
Echinus*resizeincx: AS + d = 0 0 5 0
Echinus*resizeincy: AS + w = 0 0 0 5

! raccourcis clavier(64maxi). format: <ASCW> + <key>
! A:Alt, S:Shift, C:Control, W:Super
Echinus*spawn0: A + t = xterm
Echinus*spawn1: AS + t = xterm -title root-terminal -e su

! reduit/augmente la taille du master
Echinus*decmwfact: A + h = -0.05
Echinus*incmwfact: A + l = +0.05
! enlève/ajoute un client dans le master
Echinus*decnmaster: AS + j = -1
Echinus*incnmaster: AS + k = +1

! commande exécutée lors d'un clic droit sur le bureau
Echinus*command: xterm

! règles scpéciales
! format: Echinus*rule#: <Window class|Window title> <tag> <isfloating>
<hastitle>
Echinus*rule0: Firefox.* web 0 1
Echinus*rule4: Mplayer.* NULL 1 1
Echinus*rule5: Gimp.* gfx 1 1
```

**echinus** permet le rechargement de ce fichier en cours de session avec la combinaison de touches Alt + Shift +q:

```
Echinus*restart: AS + q = echinus
```

# utilisation

une fois les raccourcis clavier de bases enregistrés, l'utilisation d'echinus est assez intuitive.

## lancement

- depuis ~/.xinitrc:

```
## launch WM ##
exec ck-launch-session /usr/local/bin/echinus
```

- avec gdm, il faudra créer un fichier de session echinus.desktop à placer dans /usr/local/share/xsessions:

```
[Desktop Entry]
Encoding=UTF-8
```

```
Name=echinus
Comment=tiling window manager
Exec=ck-launch-session /usr/local/bin/echinus
```

# outils

echinus est livré tout nu... pas de panel, pas de menu, pas de widgets.. c'est bien. mais si vous désirez rajouter quelques applications, je vous propose 9menu pour avoir un menu sur le bureau, conky dans dzen2 pour les infos, et tint2 panel pour le pager et le systray. si vous désirez un menu plus évolué, je vous conseille de visiter le wiki sur les menus indépendants.

## 9menu

9menu est un menu ultra minimal qui se lance depuis la ligne de commande ou lit un fichier de configuration. la syntaxe est simple:

```
9menu  [  -bg background-color ] [ -display displayname ] [ -file name ] [ -fg
foreground-color ] [ -font fname ] [ -geometry geom ] [ -iconic ] [ -label name ]
[ -path ] [ -popdown ] [ -popup ] [ -shell prog ] [ -teleport ] [ -version ] [
-warp  ]  menuitem[:command]
```

- bg: couleur de fond
- display: écran à utiliser
- file: nom de fichier à utiliser
- fg: couleur du texte
- font: police utilisée
- geometry: taille et placement du menu
- iconic: démarrage iconifié
- label: nom de la fenêtre de menu
- path: Fix Me!
- popdown: lorsqu'une entrée est slectionné, le menu s'iconifie
- popup: lorsqu'une entrée est slectionné, le menu quitte
- shell: définir un autre shell que /bin/sh
- teleport: place le menu sous la fenêtre
- version: affiche la version
- warp: après sélection, place la souris dans sa position initiale

dans notre exemple, nous utiliserons 9menu directement depuis la commande passée depuis un script. en premier, installons 9menu, disponible dans la plupart des dépôts des distributions courantes. pour les debianistes:

```
# apt-get install 9menu
```

puis éditons le script de lancement de 9menu:

```
#!/bin/sh
# ~/bin/echimenu.sh
exec 9menu -bg grey30 -fg grey80 -font -*-terminus-medium-*-*-*-12-*-*-*-*-*-*-*
-popup -label 'menu' 'terminal:urxvtc' 'rox-filer:rox' 'ranger:xterm -title
ranger -e ranger' 'uzbl:uzbl' 'firefox:firefox' '@jabber:xterm -title jabber -e
mcabber' 'editor:xterm -title editor -e vim' '' 'xcompmgr:xcompmgr_reset'
'lock:slock'  '! halt !:sudo halt'
```

ce script lance un menu ultra minimal qui ressemble à ceci:


il vous reste maintenant à associer ce script au clic-droit sur le bureau et définir ses propriétés; dans votre ~/.echinus/echinusrc:

```
Echinus*command: echimenu.sh
...
Echinus*rule3: 9menu.* NULL 1 0
```

pour plus d'infos sur 9menu, je vous conseille de visiter notre section dédiée.

## conky dans dzen2


dzen2 est une barre d'information hautement configurable qui va lire les informations délivrées par conky comme pour la barre de statut de dwm. l'avantage de dzen2 est qu'il accepte les formats de couleurs, polices, peut afficher des images (xbm,xpm), graphiques ou des barres de progressions, et qu'il peut s'afficher partout sur votre écran. pour plus d'informations sur l'installation, la configuration et les ooptions de dzen2, je vous laisse visiter sa page dédiée sur ce wiki. pour un affichage simple de conky, une installation standard suffit, pour les débianistes:

```
# apt-get install dzen2
```

le fichier de configuration du conky présenté dans le screenshot ressemble à celui utilisé pour dwm mais ajoute les options pour dzen:

```
out_to_x no
out_to_console yes
update_interval 1.0
total_run_times 0
use_spacer none

TEXT
 ^i(/home/arp/.arp_setups/dzicons/cal.xbm) ^fg(\#ccc)${time %d/%m}^fg()
^i(/home/arp/.arp_setups/dzicons/clock.xbm) ${time %I:%M}\
  ^fg(\#ff4500)^i(/home/arp/.arp_setups/dzicons/cpu.xbm) ^fg(\#ccc)${cpu}%\
 ^fg(\#ee2c2c)^i(/home/arp/.arp_setups/dzicons/temp.xbm) ^fg(\#ccc)${ibm_temps
0}°C\
 ^fg(\#87ceeb)^i(/home/arp/.arp_setups/dzicons/monitor.xbm) ^fg(\#ccc)${loadavg}\
  ^fg(\#00bfff)^i(/home/arp/.arp_setups/dzicons/mem.xbm) ^fg(\#ccc)$memperc% >
$mem\
  ^fg(\#ffd700)^i(/home/arp/.arp_setups/dzicons/home.xbm) ^fg(\#ccc)$
{fs_used_perc /}%\
```

```
${if_mounted /media/arp500} ^fg(\#000)^i(/home/arp/.arp_setups/dzicons/usb.xbm)
^fg(\#ccc)${fs_used_perc /media/arp500/}%${endif}${if_mounted /media/lacie300}
^fg(\#1e90ff)^i(/home/arp/.arp_setups/dzicons/usb.xbm) ^fg(\#ccc)$
{fs_used_perc /media/lacie300/}%${endif}\
 ^fg()${if_match ${battery_percent BAT0}
>=26}^fg(\#7cfc00)^i(/home/arp/.arp_setups/dzicons/bat_full_01.xbm)${endif}\
${if_match ${battery_percent BAT0} <=25}
^fg(\#CC0000)^i(/home/arp/.arp_setups/dzicons/bat_low_01.xbm)^fg()${endif}
^fg(\#ccc)${battery_percent}%\
 ^fg()${if_up eth0}  ^i(/home/arp/.arp_setups/dzicons/net_wired.xbm)\
 ^fg(\#00cd00)^i(/home/arp/.arp_setups/dzicons/net_up_02.xbm) ^fg(\#ccc)$
{upspeedf eth0}\
 ^fg(\#ffa500)^i(/home/arp/.arp_setups/dzicons/net_down_02.xbm) ^fg(\#ccc)$
{downspeedf eth0}\
^fg() ${endif}\
${if_match "$ibm_volume" == "mute" }^fg(\#000000)
^i(/home/arp/.arp_setups/dzicons/spkr_04.xbm)^fg(\#ccc) mute ${else}^fg(\#FFDF1D)
^i(/home/arp/.arp_setups/dzicons/spkr_01.xbm)^fg(\#ccc) ${ibm_volume}/14${endif}
```

**note** l'emploi du "\" en fin de ligne permet l'annulation du saut de ligne dans conky mais aide à une meilleure lisiblilité du fichier.

comme vous pouvez le constater, les options passées à conky sont "encadrées" par les options passées à dzen2:

- ^fg(): pour la couleur du texte/de l'image
- ^i() : pour l'adresse de l'image à afficher

les images doivent être au format *xbm (facilement éditables avec the Gimp). une archive contenant 67 icones est disponible ici.

ce fichier conky est appelé par dzen2 grâce à un script (que vous prendrez soin de rejouter à votre fichier ~autostart selon votre configuration):

```
#!/bin/sh
RC="$HOME/.conkyrc_dzen"
FG="white"
BG="#404240"
ALIGN="left"
WIDTH="744"
HEIGHT="12"
FONT="-*-terminus-medium-*-*-*-12-*-*-*-*-*-*-*"
XPOS="30"
YPOS="756"

conky -d -c $RC | dzen2 -fg $FG -bg $BG -ta $ALIGN -w $WIDTH -h $HEIGHT -x $XPOS
-y $YPOS -fn $FONT -dock &
exit 0
```

*sources du script: minull conky par xeNULL.*

## tint2

nous utiliserons tint2 comme taskbar/pager et pour le systray. dans notre exemple, tint2 et dzen2 sont placés tous les deux en bas de l'écran, de façon à former un "panel" homogène. tint2 sert aussi

à laisser dzen2 toujours visible.


le tint2rc associé:

```
#---------------------------------------------
# TINT2 CONFIG FILE
#---------------------------------------------

#---------------------------------------------
# BACKGROUND AND BORDER
#---------------------------------------------
#panel
rounded = 1
border_width = 0
background_color = #000000 60
border_color = #ffffff 18
#taskbar
rounded = 1
border_width = 0
background_color = #ffffff 10
border_color = #ffffff 50
#task-active
rounded = 1
border_width = 0
background_color = #ffffff 40
border_color = #ffffff 50
#task
rounded = 1
border_width = 0
background_color = #ffffff 18
border_color = #ffffff 70

#---------------------------------------------
# PANEL
#---------------------------------------------
panel_monitor = 1
panel_position = bottom right
panel_size = 245 12
panel_margin = 0 0
panel_padding = 0 0 4
font_shadow = 0
panel_background_id = 0
wm_menu = 0
panel_dock = 0
panel_layer = top

#---------------------------------------------
# TASKBAR
#---------------------------------------------
taskbar_mode = multi_desktop
#taskbar_mode = single_desktop
taskbar_padding = 0 0 2
taskbar_background_id = 2
#taskbar_active_background_id = 0

#---------------------------------------------
```

```
# TASKS
#-----------------------------------------
task_icon = 1
task_text = 0
task_maximum_size = 140 35
task_centered = 1
task_padding = 2 1
task_font = Terminus 6
task_font_color = #ffffff 70
task_background_id = 4
task_icon_asb = 100 0 0
# replace STATUS by 'urgent', 'active' or 'iconfied'
#task_STATUS_background_id = 2
#task_STATUS_font_color = #ffffff 85
#task_STATUS_icon_asb = 100 0 0
# example:
task_active_background_id = 3
task_active_font_color = #ffffff 85
task_active_icon_asb = 100 0 0
urgent_nb_of_blink = 8

#-----------------------------------------
# SYSTRAYBAR
#-----------------------------------------
systray = 1
systray_padding = 2 1 4
systray_background_id = 2
systray_sort = left2right
systray_icon_size = 0
systray_icon_asb = 100 0 0

#-----------------------------------------
# CLOCK
#-----------------------------------------
#time1_format = %d/%b - %H:%M
#time1_font = sans 8
#time2_format = %A %d %B
#time2_font = sans 6
clock_font_color = #ffffff 76
clock_padding = 1 0
clock_background_id = 0
#clock_lclick_command = xclock
clock_rclick_command = orage
#clock_tooltip = %A %d %B
#time1_timezone = :US/Hawaii
#time2_timezone = :Europe/Berlin
#clock_tooltip_timezone = :/usr/share/zoneinfo/Europe/Paris

#-----------------------------------------
# BATTERY
#-----------------------------------------
battery = 0
battery_hide = 98
battery_low_status = 10
battery_low_cmd = notify-send "battery low"
bat1_font = sans 8
bat2_font = sans 6
battery_font_color = #ffffff 76
```

```
battery_padding = 1 0
battery_background_id = 0

#--------------------------------------------
# TOOLTIP
#--------------------------------------------
tooltip = 1
tooltip_padding = 2 2
tooltip_show_timeout = 0.7
tooltip_hide_timeout = 0.3
tooltip_background_id = 1
tooltip_font_color = #ffffff 80
tooltip_font = sans 10

#--------------------------------------------
# MOUSE ACTION ON TASK
#--------------------------------------------
mouse_middle = none
mouse_right = none
mouse_scroll_up = toggle
mouse_scroll_down = iconify

#--------------------------------------------
# AUTOHIDE OPTIONS
#--------------------------------------------
autohide = 0
autohide_show_timeout = 0.3
autohide_hide_timeout = 2
autohide_height = 0
strut_policy = minimum
```

# conclusion

echinus est un excellent tiling wm qui gère aussi parfaitement le mode “libre” floating. il est rapide, léger, facilement configurable grâce à son fichier au format ~/.Xdefaults et gère le multi-écrans.
un gestionnaire de fenêtre à essayer si vous désirez vous lancer dans le tiling en toute simplicité.

---

contributeur: arpinux

sources: README des sources + post sur phollow.fr