



PeKwm est un gestionnaire de fenêtre pour X développé par Claes Nästén. PeKwm est extrêmement configurable, basé sur **awm++**, proposant la navigation par onglets dans les fenêtres, les keychains, un menu éditable, des propriétés automatiques et une foule de fichiers de configs différents... :D je crois même que c'est le wm qui utilise le plus de fichiers de conf !

si vous désirez tester pekwm en live, allez faire un tour du côté du [livarp](#) ;) la version 0.3 a une session pekwm préconfigurée...

screenshots



1



2



3



4

- 1 PeKwm/bmpanel par [ed1w2ard](#), thème gtk/pekwm: [Azul](#), bmpanel: XSOcaM (inclus dans [bmpanel2](#)),
- 2 PeKwm Audrey setup par [Izobalax](#), portage du [Audrey set](#) par [ptpbs](#) dispo [ici](#).
- 3-4 [PeKwm setup](#) par arpinux.

installation

PeKwm est disponible dans les dépôts de la plupart des distribution Linux. la dernière version est pekwm-0.1.16 publiée le 2012-09-21.

pour les distributions basées sur **Debian**:

```
# apt-get install pekwm pekwm-themes
```

mais vous pouvez également télécharger les **sources** depuis la [page de download](#) du site pekwm. puis appliquer la procédure de compilation classique: (il vous faudra cependant aller chercher quelques thèmes supplémentaires)

```
wget http://www.pekwm.org/projects/pekwm/files/pekwm-0.1.16.tar.bz2
tar xvjf pekwm-0.1.16.tar.bz2
cd pekwm-0.1.16/
./configure
make
sudo make install
```

lancement

si vous utilisez **gdm**, PeKwm devrait apparaitre dans votre choix de session.

si vous utilisez **startx**, il suffit de rajouter à la fin de votre `~/.xinitrc`:

```
#!/bin/bash
#####
# ~/.xinitrc by arpinux 2011 #
#####

## D-Bus ##
if which dbus-launch >/dev/null && test -z "$DBUS_SESSION_BUS_ADDRESS"; then
    eval "$(dbus-launch --sh-syntax --exit-with-session)"
fi

## trackpad ## tapbutton off by default ##
synclient VertTwoFingerScroll=1
synclient HorizTwoFingerScroll=1
synclient TapButton1=0
## dualscreen ## edit if needed ##
xrandr --output LVDS --mode 1024x768 --pos 0x0 --rotate normal --output VGA-0 --mode 1024x768 --pos 1024x0 --rotate normal
## panel ##
fbpanel &
## launch terminal daemon ##
urxvtd &
## mail-checker ##
sleep 4 && claws-mail --online &
## wallpaper ##
nitrogen --restore &
## launch WM ##
exec ck-launch-session pekwm
```

configuration

la configuration de PeKwm passe par l'édition de fichiers situés dans votre ~/.pekwm. il faut le créer

```
mkdir ~/.pekwm
```

puis copier les fichiers de configuration par défaut.
si vous avez installé pekwm depuis les sources (version 0.1.16):

```
cp /usr/local/etc/pekwm/* ~/.pekwm/
```

si vous avez installé le paquet debian squeeze (version 0.1.12):

```
cp /etc/pekwm/* ~/.pekwm/
```

lors de la première installation/lancement, vous n'aurez ni barre de tâche, ni panel, ni icônes sur le bureau... seulement un menu au clic droit sur le bureau. pour agrémenter un peu votre PeK, passons aux fichiers de configuration:

~/.pekwm/start

ce fichier joue le rôle du fichier autostart dans openbox: un script de lancement pour vos applications/préférences au démarrage de votre session. **notez** cependant que ce fichier sera **relu à chaque rechargement** de pekwm. il est donc conseillé de lancer vos applications depuis votre ~/.xinitrc.

vous pouvez néanmoins utiliser ce fichier pour lancer un script de fond d'écran aléatoire par exemple:

```
#!/bin/sh
# PekWM start file
# This file is a simple shell script; It gets run on pekwm startup, after
# the theme and all config has loaded if it is set executable
# (chmod +x start).
#
# This is different from ~/.xinitrc because a normal configuration of
# .xinitrc you'll run all commands, then launch the window manager last.
#
# It also gets re-run every time pekwm is restarted.
#
# As for it's usefulness, well, it's up to you. I actually set my background
# from my start file; since it runs after the theme gets loaded, this
# effectively overrides whatever's in the theme.
#
# There's probably a few other good uses for it, too. I mainly pushed for it
# because when I was doing fluxbox's docs, people used to complain that
# there
```

```
# wasn't one, and I wanted to avoid that for pekwm. ;) --eyez

# change aléatoirement le fond d'écran depuis un dossier #
# nécessite feh <# apt-get install feh> #
find [adresse_du_dossier] -type f -name '*.jpg' -o -name '*.png' | shuf -n 1
| xargs feh --no-xinerama --bg-scale &
```

~/.pekwm/config

ce fichier correspond au config.xml chez openbox, ou le wmfsr sur [wmfs](#), mais en plus simplifié, étant donné que certaines variables seront configurés dans des fichiers annexes. c'est dans ce fichier que vous allez indiquer le nombre de bureau, ainsi que leurs noms, les adresses de fichiers de configuration, les propriétés des fenêtres... en voici un exemplaire commenté:

```
## emplacement des fichiers annexes à utiliser##
Files {
    Keys = "~/.pekwm/keys"
    Mouse = "~/.pekwm/mouse"
    Menu = "~/.pekwm/menu"
    Start = "~/.pekwm/start"
    AutoProps = "~/.pekwm/autoproperties"
    Theme = "/usr/share/pekwm/themes/default"
    Icons = "~/.pekwm/icons/"
}

## comportement des fenêtres lors du déplacement/redimensionnement ##
MoveResize {
    EdgeAttract = "10"          ## distance d'adhérence au bords en pixel
    EdgeResist = "10"           ## résistance des bords d'écran
    WindowAttract = "5"        ## distance d'adhérence entre fenêtres en
pixel
    WindowResist = "5"          ## résistance des fenêtres
    OpaqueMove = "True"         ## dessiner les fenêtres lors du déplacement,
mettre "False" pour augmenter la réactivité
    OpaqueResize = "False"      ## dessiner les fenêtres lors du
redimensionnement
}

## propriétés des bureaux
Screen {
    Workspaces = "9"             ## nombre de bureaux
    WorkspacesPerRow = "3"       ## nombre de bureaux par
colonne: "0" pour mettre en ligne.
    WorkspaceNames = "Main;Web;E-mail;Music;Work 05;Work 06;Work 07;Work
08;Work 09"  ## nom des bureaux
    ShowFrameList = "True"       ## afficher la liste des
fenêtres
    ShowStatusWindow = "True"    ## afficher les informations
```

```

sur la fenêtre
    ShowStatusWindowCenteredOnRoot = "False"  ## afficher les informations
sur la fenêtre au centre de l'écran
    ShowClientID = "False"                    ## affiche l'ID de la fenêtre
    ShowWorkspaceIndicator = "500"            ## temps d'affichage de la
fenêtre d'information au changement de bureau
    PlaceNew = "True"                         ## on tente de placer le nouveau client, si
False, il apparaît au coin supérieur gauche
    FocusNew = "True"                         ## le nouveau client passe au premier plan

    TrimTitle = "..."                       ## ce qui remplace le titre de la
fenêtre quand la place manque
    FullscreenAbove = "True"                  ## le client en plein écran est au
premier plan
    FullscreenDetect = "True"                 ## détecter les écrans lors d'un appel
au fullscreen
    HonourRandr = "True"                     ## lire les informations de Xrandr
    HonourAspectRatio = "True"                ## préserver le ratio d'affichage des
fenêtres
    EdgeSize = "1 1 1 1"                     ## distance des bords de l'écran (haut
bas gauche droite)
    EdgeIndent = "False"                      ## les bords d'écran doivent-ils être
réservés
    PixmapCacheSize = "20"                    ## nombre d'image inutilisée à garder
dans le cache
    DoubleClickTime = "250"                   ## temps entre deux clics pour devenir
un double-clic en ms

## placement des fenêtres
## options du Model:
## Smart: tente de placer la fenêtre dans un espace inoccupé
## MouseCentered: place le centre de la fenêtre sous le pointeur
## MouseTopLeft: place le coin supérieur gauche de la fenêtre sous le
pointeur
## MouseNotUnder: place la fenêtre au coin supérieur gauche de l'écran
## CenteredOnParent: place la fenêtre au centre de leur fenêtre parent
Placement {
    Model = "CenteredOnParent Smart MouseNotUnder"
    Smart {
        Row = "True"                          ## utilise les lignes, si False, utilise
les colonnes
        TopToBottom = "True"                  ## placement du haut vers le bas
        LeftToRight = "True"                  ## placement de la gauche vers la droite
        OffsetX = "0"                         ## décalage horizontal du placement
        OffsetY = "0"                         ## décalage vertical du placement
    }
}

## définir si une fenêtre doit avoir un nom unique dans la liste des
fenêtres
## si oui, le nom sera suivi d'un numéro

```

```
UniqueNames {
    SetUnique = "False" ## utiliser l'option nom unique
    Pre = " #"          ## texte à placer devant le numéro de fenêtre
    Post = ""           ## texte à placer après le numéro de la fenêtre
}
}

## configuration du menu
Menu {
    DisplayIcons = "True" ## afficher les icones de menu

    Icons = "DEFAULT" {    ## taille des icones en pixel du menu "DEFAULT"
        Minimum = "16x16"
        Maximum = "16x16"
    }

    # pour avoir des icones de tailles différentes dans d'autres menus,
    # par exemple le menu wallpaper, utiliser cette syntaxe pour chaque
    # que vous voulez "libérer".

    # Icons = "Wallpaper" {
    #     Minimum = "64x64"
    #     Maximum = "64x64"
    # }

    # Definition du comportement de la souris sur les items de menu:
    # valeurs possibles: "ButtonPress ButtonRelease DoubleClick Motion"
    # pour avoir les sous-menus qui s'ouvrent au passage du pointeur,
    # commenter la ligne 'Enter', décommenter la ligne du dessous puis
    # relancer pekwm.

    Select = "Motion MotionPressed"      ## selection au survol
    Enter = "MotionPressed ButtonPress"   ## entrer au clic
    # Enter = "Motion"                    ## entrer au survol
    Exec = "ButtonRelease"                ## executer lors du relachement
}

## configuration du lanceur intégré
CmdDialog {
    HistoryUnique = "True"                ## eviter les répétitions dans
l'historique des commandes
    HistorySize = "1024"                  ## taille de l'historique
    HistoryFile = "~/pekwm/history"       ## fichier de stockage de l'historique
    HistorySaveInterval = "16"            ## intervalle de sauvegarde dans le
fichier history
}

## configuration du harbour pouvant inclure des dockapps
Harbour {
    OnTop = "True"                        ## placer au-dessus
    MaximizeOver = "False"                ## masquer derrière les fenêtres maximisées
}
```

```

Placement = "Right"      ## position
Orientation = "TopToBottom" ## orientation
Head = "0"               ## écran
## configuration des dockapps
DockApp {
    SideMin = "64"  ## taille minimale
    SideMax = "0"   ## taille maximale 0=illimitée
}
}

```

~/.pekwm/keys

la configuration des raccourcis clavier et des chaines de raccourcis. en effet, pekwm gère les key-chains, une sorte de sous-menu de raccourcis clavier qui vous permettent d'augmenter considérablement le nombre de raccourcis possibles.

si vous modifiez ce fichier, assurez-vous de ne pas obtenir de doublons, pour cela, je vous conseille de commenter/désactiver tous les raccourcis inutiles avant de modifier ce fichier.

```

INCLUDE = "vars"

Global {
# - - - - -
# configuration des raccourcis clavier.
# MOD1 pour Alt , Mod4 pour Super (touche logo)
# pour ajouter les votres, aller dans la section
# 'Chaines'-'commandes externes' puis éditer les raccourcis.
    # se deplacer parmis les fenêtres
    KeyPress = "Mod1 Tab" { Actions = "NextFrame EndRaise" }
## aller à la fenêtre suivante
    KeyPress = "Mod1 Shift Tab" { Actions = "PrevFrame EndRaise" }
## aller à la fenêtre précédente
    KeyPress = "Mod1 Ctrl Tab" { Actions = "NextFrameMRU EndRaise" }
## aller à la fenêtre suivante et lui laisser le focus
    KeyPress = "Mod1 Ctrl Shift Tab" { Actions = "PrevFrameMRU EndRaise" }
} ## aller à la fenêtre précédente et lui laisser le focus
    KeyPress = "Mod4 Tab" { Actions = "ActivateClientRel 1" }
## aller à la fenêtrés suivante dans un groupe de fenêtres
    KeyPress = "Mod4 Shift Tab" { Actions = "ActivateClientRel -1" }
## aller à la fenêtrés précédente dans un groupe de fenêtres
    KeyPress = "Mod4 Ctrl Right" { Actions = "MoveClientRel 1" }
## déplacer la fenêtre vers la droite dans un groupe de fenêtres
    KeyPress = "Mod4 Ctrl Left" { Actions = "MoveClientRel -1" }
## déplacer la fenêtre vers la gauche dans un groupe de fenêtres
    KeyPress = "Mod4 Left" { Actions = "FocusDirectional Left" }
## déplace le focus vers la gauche
    KeyPress = "Mod4 Right" { Actions = "FocusDirectional Right" }
## déplace le focus vers la droite

```

```
KeyPress = "Mod4 Up" { Actions = "FocusDirectional Up" }
## déplace le focus vers le haut
KeyPress = "Mod4 Down" { Actions = "FocusDirectional Down" }
## déplace le focus vers le bas
# se déplacer dans les bureaux
KeyPress = "Ctrl Mod1 Left" { Actions = "GotoWorkspace Left" }
## aller au bureau de gauche
KeyPress = "Ctrl Mod1 Right" { Actions = "GotoWorkspace Right" }
## aller au bureau de droite
KeyPress = "Ctrl Mod1 Up" { Actions = "GotoWorkspace Up" }
## aller au bureau du haut
KeyPress = "Ctrl Mod1 Down" { Actions = "GotoWorkspace Down" }
## aller au bureau du bas
KeyPress = "Mod4 1" { Actions = "GotoWorkspace 1" }
## aller au bureau numéro 1
KeyPress = "Mod4 2" { Actions = "GotoWorkspace 2" }
## ... 2 ...
KeyPress = "Mod4 3" { Actions = "GotoWorkspace 3" }
KeyPress = "Mod4 4" { Actions = "GotoWorkspace 4" }
KeyPress = "Mod4 5" { Actions = "GotoWorkspace 5" }
KeyPress = "Mod4 6" { Actions = "GotoWorkspace 6" }
KeyPress = "Mod4 7" { Actions = "GotoWorkspace 7" }
KeyPress = "Mod4 8" { Actions = "GotoWorkspace 8" }
KeyPress = "Mod4 9" { Actions = "GotoWorkspace 9" }
KeyPress = "Ctrl Mod1 Shift Left" { Actions = "SendToWorkspace Next;
GoToWorkspace Next" } ## envoyer et suivre la fenêtre sur le bureau de
gauche
KeyPress = "Ctrl Mod1 Shift Right" { Actions = "SendToWorkspace Prev;
GoToWorkspace Prev" } ## envoyer et suivre la fenêtre sur le bureau de
droite
KeyPress = "Ctrl Mod1 Shift Up" { Actions = "SendToWorkspace NextV;
GoToWorkspace NextV" } ## envoyer et suivre la fenêtre sur le bureau du
haut
KeyPress = "Ctrl Mod1 Shift Down" { Actions = "SendToWorkspace PrevV;
GoToWorkspace PrevV" } ## envoyer et suivre la fenêtre sur le bureau du bas
KeyPress = "Mod4 F1" { Actions = "SendToWorkspace 1" }
## envoyer la fenêtre sur le bureau numero 1
KeyPress = "Mod4 F2" { Actions = "SendToWorkspace 2" }
## ... 2 ...
KeyPress = "Mod4 F3" { Actions = "SendToWorkspace 3" }
KeyPress = "Mod4 F4" { Actions = "SendToWorkspace 4" }
KeyPress = "Mod4 F5" { Actions = "SendToWorkspace 5" }
KeyPress = "Mod4 F6" { Actions = "SendToWorkspace 6" }
KeyPress = "Mod4 F7" { Actions = "SendToWorkspace 7" }
KeyPress = "Mod4 F8" { Actions = "SendToWorkspace 8" }
KeyPress = "Mod4 F9" { Actions = "SendToWorkspace 9" }
# action de base sur les fenêtres
KeyPress = "Mod4 M" { Actions = "Toggle Maximized True True" } ##
maximise/restaure la fenêtre
KeyPress = "Mod4 G" { Actions = "Maxfill True True" } ##
active/desactive remplir l'espace disponible
```



```

    KeyPress = "Mod4 F" { Actions = "Toggle FullScreen" }      ##
active/desactive le plein écran
    KeyPress = "Mod4 Return" { Actions = "MoveResize" }        ##
active la fonction deplacer/redimensionner par le clavier
    KeyPress = "Mod4 Q" { Actions = "Close" }                   ##
ferme la fenêtre
    KeyPress = "Mod4 S" { Actions = "Toggle Shaded" }          ##
enroule/deroule la fenêtre
    KeyPress = "Mod4 I" { Actions = "Toggle Iconified" }        ##
minimise/restaure la fenêtre
    # Marquage
    KeyPress = "Mod4 Z" { Actions = "Toggle Marked" }          ##
marque/demarque la fenêtre
    KeyPress = "Mod4 A" { Actions = "AttachMarked" }            ##
groupe toutes les fenêtres marquée à la fenêtre courante
    # Tagging
    KeyPress = "Mod4 T" { Actions = "Toggle Tagged False" }     ##
??
    # Menus
    KeyPress = "Mod4 R" { Actions = "ShowMenu Root" }          ##
menu principal
    KeyPress = "Mod4 W" { Actions = "ShowMenu Window" }        ##
menu client
    KeyPress = "Mod4 L" { Actions = "ShowMenu Goto" }           ##
menu goto
    KeyPress = "Mod4 C" { Actions = "ShowMenu GotoClient" }     ##
menu goto client
    KeyPress = "Mod4 Shift I" { Actions = "ShowMenu Icon" }     ##
??
    KeyPress = "Mod4 X" { Actions = "HideAllMenus" }            ##
cache tous les menus
    # commandes externes
    KeyPress = "Mod4 E" { Actions = "Exec $TERM" }              ##
lance le terminal défini dans ~/.pekwm/vars
    # controle de PeKwm
    KeyPress = "Ctrl Mod1 Delete" { Actions = "Reload" }       ##
relancer PeKwm
    KeyPress = "Mod4 D" { Actions = "ShowCmdDialog" }           ##
lancer la commande rapide
    KeyPress = "Mod4 V" { Actions = "ShowSearchDialog" }       ##
lancer la recherche rapide
    KeyPress = "Mod4 H" { Actions = "Toggle HarbourHidden" }   ##
affiche/masque le harbour

# - - - - -
# Chaines: elles vous donnent accès à absolument tout.
# certaines entrées sont doublées afin de pouvoir être accessibles des deux
mains
    # déplacement sur les coins/bords de l'écran.
    Chain = "Ctrl Mod1 C" {
## la chaine commence par Ctrl+Alt+c

```

```
KeyPress = "Q" { Actions = "MoveToEdge TopLeft" }
## vers le coin haut/gauche
KeyPress = "Y" { Actions = "MoveToEdge TopCenterEdge" }
## vers le haut/centre
KeyPress = "W" { Actions = "MoveToEdge TopCenterEdge" }
## ..
KeyPress = "Shift Y" { Actions = "MoveToEdge TopEdge" }
## vers le haut
KeyPress = "Shift W" { Actions = "MoveToEdge TopEdge" }
## ..
KeyPress = "P" { Actions = "MoveToEdge TopRight" }
## vers le coin haut/droite
KeyPress = "E" { Actions = "MoveToEdge TopRight" }
## ..
KeyPress = "A" { Actions = "MoveToEdge LeftCenterEdge" }
## vers la gauche/centre
KeyPress = "Shift A" { Actions = "MoveToEdge LeftEdge" }
## vers la gauche
KeyPress = "L" { Actions = "MoveToEdge RightCenterEdge" }
## vers la droite/centre
KeyPress = "D" { Actions = "MoveToEdge RightCenterEdge" }
## ..
KeyPress = "Shift L" { Actions = "MoveToEdge RightEdge" }
## vers la droite
KeyPress = "Shift D" { Actions = "MoveToEdge RightEdge" }
## ..
KeyPress = "Z" { Actions = "MoveToEdge BottomLeft" }
## vers le coin bas/gauche
KeyPress = "B" { Actions = "MoveToEdge BottomCenterEdge" }
## vers le bas/centre
KeyPress = "X" { Actions = "MoveToEdge BottomCenterEdge" }
## ..
KeyPress = "Shift B" { Actions = "MoveToEdge BottomEdge" }
## vers le haut
KeyPress = "Shift X" { Actions = "MoveToEdge BottomEdge" }
## ..
KeyPress = "M" { Actions = "MoveToEdge BottomRight" }
## vers le coin bas/droite
KeyPress = "C" { Actions = "MoveToEdge BottomRight" }
## ..
KeyPress = "H" { Actions = "MoveToEdge Center" }
## au centre
KeyPress = "S" { Actions = "MoveToEdge Center" }
## ..
}
# Menus
Chain = "Ctrl Mod1 M" {
## la chaine commence par Ctrl+Alt+m
KeyPress = "R" { Actions = "ShowMenu Root" }
## affiche le menu principal
KeyPress = "W" { Actions = "ShowMenu Window" }
```

```
## affiche le menu client
    KeyPress = "I" { Actions = "ShowMenu Icon" }
## ??
    KeyPress = "G" { Actions = "ShowMenu Goto" }
## affiche le menu goto
    KeyPress = "C" { Actions = "ShowMenu GotoClient" }
## affiche le menu goto client
    KeyPress = "D" { Actions = "ShowMenu Decor" }
## affiche le menu décoration
    KeyPress = "A" { Actions = "ShowMenu AttachClientInFrame" }
## affiche le menu grouper le client avec la fenêtre
    KeyPress = "F" { Actions = "ShowMenu AttachFrameInFrame" }
## affiche le menu grouper le groupe
    KeyPress = "Shift A" { Actions = "ShowMenu AttachClient" }
## affiche le menu grouper le client
    KeyPress = "Shift F" { Actions = "ShowMenu AttachFrame" }
## affiche le menu grouper
    KeyPress = "X" { Actions = "HideAllMenus" }
## cache tous les menus
    }
    # grouper les clients
    Chain = "Ctrl Mod1 T" {
## cette chaine commence par Ctrl+Alt+t
    KeyPress = "T" { Actions = "Toggle Tagged False" }
##
    KeyPress = "B" { Actions = "Toggle Tagged True" }
##
    KeyPress = "C" { Actions = "Unset Tagged" }
##
    KeyPress = "G" { Actions = "Toggle GlobalGrouping" }
##
    KeyPress = "M" { Actions = "Toggle Marked" }
## marque/demarque
    KeyPress = "A" { Actions = "AttachMarked" }
## groupe les clients marqués
    KeyPress = "D" { Actions = "Detach" }
## détache
    KeyPress = "P" { Actions = "AttachClientInNextFrame" }
## attache le client dans le groupe suivant
    KeyPress = "O" { Actions = "AttachClientInPrevFrame" }
## attache le client dans le groupe précédent
    KeyPress = "I" { Actions = "AttachFrameInNextFrame" }
## attache le groupe au groupe suivant
    KeyPress = "U" { Actions = "AttachFrameInPrevFrame" }
## attache le groupe au groupe précédent
    }
    # aficher les décorations de fenêtres
    Chain = "Ctrl Mod1 D" {
    KeyPress = "B" { Actions = "Toggle DecorBorder" }
## affiche/masque les bordures
    KeyPress = "T" { Actions = "Toggle DecorTitlebar" }
```

```

## affiche/masque la barre de titre
    KeyPress = "D" { Actions = "Toggle DecorBorder; Toggle
DecorTitlebar" } ## affiche/masque les décorations
    }
    # actions avancées sur les fenêtres
    Chain = "Ctrl Mod1 A" {
        Chain = "G" {
            KeyPress = "G" { Actions = "MaxFill True True" }
## remplir l'espace horiz/vert libre
            KeyPress = "V" { Actions = "MaxFill False True" }
## remplir l'espace vertical libre
            KeyPress = "H" { Actions = "MaxFill True False" }
## remplir l'espace horizontal libre
        }
        Chain = "M" {
            KeyPress = "M" { Actions = "Toggle Maximized True
True" } ## maximiser la fenêtre
            KeyPress = "V" { Actions = "Toggle Maximized False
True" } ## maximiser verticalement
            KeyPress = "H" { Actions = "Toggle Maximized True
False" } ## maximiser horizontalement
        }
        Chain = "Q" {
            KeyPress = "Q" { Actions = "Close" } ##
ferme le client
            KeyPress = "F" { Actions = "CloseFrame" } ##
ferme la fenêtre (client ou groupe de client)
            KeyPress = "K" { Actions = "Kill" } ##
tue (kill) le client
        }
        KeyPress = "S" { Actions = "Toggle Shaded" } ##
enroule/deroule la fenêtre
        KeyPress = "A" { Actions = "Toggle Sticky" } ##
active/desactive afficher sur tous les bureaux
        KeyPress = "O" { Actions = "Toggle AlwaysOnTop" } ##
active/desactive toujours au-dessus
        KeyPress = "B" { Actions = "Toggle AlwaysBelow" } ##
active/desactive toujours en-dessous
        KeyPress = "I" { Actions = "Set Iconified" } ##
minimise
        KeyPress = "R" { Actions = "Raise" } ##
donne le focus
        KeyPress = "Shift R" { Actions = "Raise True" } ##
donne le focus et met au premier plan
        KeyPress = "L" { Actions = "Lower" } ##
enlève le focus
        KeyPress = "Shift L" { Actions = "Lower True" } ##
enlève le focus et place au dernier plan
        KeyPress = "X" { Actions = "ActivateOrRaise" } ##
donne le focus ou met au premier plan
        KeyPress = "Return" { Actions = "MoveResize" } ##

```

```
deplace/redimensionne
    KeyPress = "F" { Actions = "Toggle Fullscreen" }      ##
active/desactive le plein écran
    KeyPress = "Left" { Actions = "GrowDirection Left" }   ##
agrandir la fenêtre vers la gauche
    KeyPress = "Right" { Actions = "GrowDirection Right" } ##
agrandir la fenêtre vers la droite
    KeyPress = "Up" { Actions = "GrowDirection Up" }        ##
agrandir la fenêtre vers le haut
    KeyPress = "Down" { Actions = "GrowDirection Down" }    ##
agrandir la fenêtre vers le bas
    }
# se déplacer dans un groupe de clients
Chain = "Ctrl Mod1 F" {
    KeyPress = "P" { Actions = "NextFrame AlwaysRaise" }
## aller au groupe de fenêtres suivant et le mettre au premier plan
    KeyPress = "O" { Actions = "PrevFrame AlwaysRaise" }
## aller au groupe de fenêtres précédent et le mettre au premier plan

    KeyPress = "Shift P" { Actions = "NextFrameMRU EndRaise" }
## aller au groupe de fenêtres suivant et mettre le client au premier plan

    KeyPress = "Shift O" { Actions = "PrevFrameMRU EndRaise" }
## aller au groupe de fenêtres précédent et mettre le client au premier plan
    KeyPress = "I" { Actions = "ActivateClientRel 1" }
## activer le client à droite
    KeyPress = "U" { Actions = "ActivateClientRel -1" }
## activer le client à gauche
    KeyPress = "Shift I" { Actions = "MoveClientRel 1" }
## déplacer le client vers la droite
    KeyPress = "Shift U" { Actions = "MoveClientRel -1" }
## déplacer le client vers la gauche
    KeyPress = "Up" { Actions = "FocusDirectional Up" }
## déplacer le focus vers le haut
    KeyPress = "Down" { Actions = "FocusDirectional Down" }
## déplacer le focus vers le bas
    KeyPress = "Left" { Actions = "FocusDirectional Left" }
## déplacer le focus vers la gauche
    KeyPress = "Right" { Actions = "FocusDirectional Right" }
## déplacer le focus vers la droite
    KeyPress = "1" { Actions = "ActivateClientNum 1" }
## activer le client numéro 1
    KeyPress = "2" { Actions = "ActivateClientNum 2" }
## ... 2 ...
    KeyPress = "3" { Actions = "ActivateClientNum 3" }
##
    KeyPress = "4" { Actions = "ActivateClientNum 4" }
##
    KeyPress = "5" { Actions = "ActivateClientNum 5" }
##
    KeyPress = "6" { Actions = "ActivateClientNum 6" }
```

```

##
    KeyPress = "7" { Actions = "ActivateClientNum 7" }
##
    KeyPress = "8" { Actions = "ActivateClientNum 8" }
##
    KeyPress = "9" { Actions = "ActivateClientNum 9" }
##
    KeyPress = "0" { Actions = "ActivateClientNum 10" }
##
    KeyPress = "C" { Actions = "ShowCmdDialog GotoClientID " }
## ??
}
# bureaux
Chain = "Ctrl Mod1 W" {
    KeyPress = "Right" { Actions = "GoToWorkspace Right" } ##
    aller au bureau de droite
    KeyPress = "Left" { Actions = "GoToWorkspace Left" } ##
    aller au bureau de gauche
    KeyPress = "N" { Actions = "GoToWorkspace Next" } ##
    aller au bureau suivant
    KeyPress = "P" { Actions = "GoToWorkspace Prev" } ##
    aller au bureau précédent
    KeyPress = "1" { Actions = "GoToWorkspace 1" } ##
    aller au bureau numéro 1
    KeyPress = "2" { Actions = "GoToWorkspace 2" } ##
    ... 2 ...
    KeyPress = "3" { Actions = "GoToWorkspace 3" }
    KeyPress = "4" { Actions = "GoToWorkspace 4" }
    KeyPress = "5" { Actions = "GoToWorkspace 5" }
    KeyPress = "6" { Actions = "GoToWorkspace 6" }
    KeyPress = "7" { Actions = "GoToWorkspace 7" }
    KeyPress = "8" { Actions = "GoToWorkspace 8" }
    KeyPress = "9" { Actions = "GoToWorkspace 9" }
    KeyPress = "Up" { Actions = "SendToWorkspace Next;
GoToWorkspace Next" } ## envoyer et suivre la fenêtre au bureau suivant
    KeyPress = "Down" { Actions = "SendToWorkspace Prev;
GoToWorkspace Prev" } ## envoyer et suivre la fenêtre au bureau précédent
    KeyPress = "F1" { Actions = "SendToWorkspace 1" } ##
    envoyer au bureau numéro 1
    KeyPress = "F2" { Actions = "SendToWorkspace 2" } ##
    ... 2 ...
    KeyPress = "F3" { Actions = "SendToWorkspace 3" }
    KeyPress = "F4" { Actions = "SendToWorkspace 4" }
    KeyPress = "F5" { Actions = "SendToWorkspace 5" }
    KeyPress = "F6" { Actions = "SendToWorkspace 6" }
    KeyPress = "F7" { Actions = "SendToWorkspace 7" }
    KeyPress = "F8" { Actions = "SendToWorkspace 8" }
    KeyPress = "F9" { Actions = "SendToWorkspace 9" }
}
# commandes externes
Chain = "Ctrl Mod1 E" {

```

```

        KeyPress = "E" { Actions = "Exec $TERM" } ##
lance le terminal défini dans ~/.pekwm/vars
        KeyPress = "L" { Actions = "Exec xlock -mode blank &" } ##
verrouille l'écran en noir
        KeyPress = "S" { Actions = "Exec scrot &" } ##
capture d'écran
        KeyPress = "C" { Actions = "ShowCmdDialog" } ##
lance la commande rapide
    }
    # actions sur le gestionnaire de fenêtres
    Chain = "Ctrl Mod1 P" {
        KeyPress = "Delete" { Actions = "Reload" } ##
recharger PeKwm
        KeyPress = "Next" { Actions = "Restart" } ##
redémarrer PeKwm
        KeyPress = "End" { Actions = "Exit" } ##
quitter PeKwm
        KeyPress = "Prior" { Actions = "RestartOther twm" } ##
quitter PeKwm et basculer sur un autre WM
        KeyPress = "D" { Actions = "ShowCmdDialog" } ##
lancer la commande rapide
        KeyPress = "H" { Actions = "Toggle HarbourHidden" } ##
afficher/masquer le harbour
    }
    # Skipping
    Chain = "Ctrl Mod1 S" {
        Keypress = "M" { Actions = "Toggle Skip Menus" } ##
        Keypress = "F" { Actions = "Toggle Skip FocusToggle" } ##
        Keypress = "S" { Actions = "Toggle Skip Snap" } ##
active/desactive l'adherance
    }
}

# raccourcis actif lors de l'utilisation de 'MoveResize'
MoveResize {
    KeyPress = "Left" { Actions = "MoveHorizontal -10" } ##
deplace le client de 10px vers la gauche
    KeyPress = "Right" { Actions = "MoveHorizontal 10" } ##
deplace le client de 10px vers la droite
    KeyPress = "Up" { Actions = "MoveVertical -10" } ##
deplace le client de 10px vers le haut
    KeyPress = "Down" { Actions = "MoveVertical 10" } ##
deplace le client de 10px vers le bas
    Keypress = "Shift Left" { Actions = "MoveHorizontal -1" } ##
deplace le client de 1px vers la gauche
    Keypress = "Shift Right" { Actions = "MoveHorizontal 1" } ##
deplace le client de 1px vers la droite
    Keypress = "Shift Up" { Actions = "MoveVertical -1" } ##
deplace le client de 1px vers le bas
    Keypress = "Shift Down" { Actions = "MoveVertical 1" } ##
deplace le client de 1px vers le haut

```

```
        Keypress = "Mod4 Left" { Actions = "ResizeHorizontal -10" }      ##
redimensionne le client de 10px vers la gauche
        Keypress = "Mod4 Right" { Actions = "ResizeHorizontal 10" }      ##
redimensionne le client de 10px vers la droite
        Keypress = "Mod4 Up" { Actions = "ResizeVertical -10" }          ##
redimensionne le client de 10px vers le haut
        Keypress = "Mod4 Down" { Actions = "ResizeVertical 10" }         ##
redimensionne le client de 10px vers le bas
        Keypress = "Mod1 Left" { Actions = "ResizeHorizontal -10" }      ##
redimensionne le client de 10px vers la gauche
        Keypress = "Mod1 Right" { Actions = "ResizeHorizontal 10" }      ##
redimensionne le client de 10px vers la droite
        Keypress = "Mod1 Up" { Actions = "ResizeVertical -10" }          ##
redimensionne le client de 10px vers le haut
        Keypress = "Mod1 Down" { Actions = "ResizeVertical 10" }         ##
redimensionne le client de 10px vers le bas
        Keypress = "Shift Mod4 Left" { Actions = "ResizeHorizontal -1" } ##
redimensionne le client de 1px vers la gauche
        Keypress = "Shift Mod4 Right" { Actions = "ResizeHorizontal 1" } ##
redimensionne le client de 1px vers la droite
        Keypress = "Shift Mod4 Up" { Actions = "ResizeVertical -1" }      ##
redimensionne le client de 1px vers le haut
        Keypress = "Shift Mod4 Down" { Actions = "ResizeVertical 1" }     ##
redimensionne le client de 1px vers le bas
        Keypress = "Shift Mod1 Left" { Actions = "ResizeHorizontal -1" } ##
redimensionne le client de 1px vers la gauche
        Keypress = "Shift Mod1 Right" { Actions = "ResizeHorizontal 1" } ##
redimensionne le client de 1px vers la droite
        Keypress = "Shift Mod1 Up" { Actions = "ResizeVertical -1" }      ##
redimensionne le client de 1px vers le haut
        Keypress = "Shift Mod1 Down" { Actions = "ResizeVertical 1" }     ##
redimensionne le client de 1px vers le bas
        Keypress = "s" { Actions = "MoveSnap" }                          ##
deplace le client vers le bord le plus proche
        Keypress = "Escape" { Actions = "Cancel" }                      ##
annule la chaine
        Keypress = "q" { Actions = "Cancel" }                          ##
..
        Keypress = "Return" { Actions = "End" }                        ##
fin de la chaine
}

# raccourcis de la commande rapide
InputDialog {
        KeyPress = "Left" { Actions = "CursPrev" }                      ##
deplacer le pointeur vers la gauche
        KeyPress = "Right" { Actions = "CursNext" }                    ##
deplacer le pointeur vers la droite
        KeyPress = "Ctrl A" { Actions = "CursBegin" }                  ##
deplacer le pointeur au debut
        KeyPress = "Ctrl E" { Actions = "CursEnd" }                    ##
}
```



```

deplacer le pointeur à la fin
    KeyPress = "BackSpace" { Actions = "Erase;CompleteAbort" } ##
effacer, abandonner la commande
    KeyPress = "Ctrl K" { Actions = "ClearFromCursor" }      ##
effacer depuis le pointeur
    KeyPress = "Ctrl C" { Actions = "Clear" }                 ##
effacer la commande
    KeyPress = "Return" { Actions = "Exec" }                  ##
executer la commande
    KeyPress = "Escape" { Actions = "Close" }                 ## fermer
la commande rapide
    KeyPress = "Up" { Actions = "HistPrev" }                  ##
montrer la commande précédente
    KeyPress = "Down" { Actions = "HistNext" }                ##
montrer la commande suivante
    KeyPress = "Ctrl P" { Actions = "HistPrev" }              ##
montrer la commande précédente
    KeyPress = "Ctrl N" { Actions = "HistNext" }              ##
montrer la commande suivante
    KeyPress = "Ctrl B" { Actions = "CursPrev" }              ##
deplacer le pointeur vers la gauche
    KeyPress = "Ctrl F" { Actions = "CursNext" }              ##
deplacer le pointeur vers la droite
    KeyPress = "Tab" { Actions = "Complete" }                 ##
activer la completion automatique
    KeyPress = "Any Any" { Actions = "Insert" }               ##
insérer
}

# touches pour les menus
Menu {
    KeyPress = "Down" { Actions = "NextItem" }                ## item suivant
    KeyPress = "Up" { Actions = "PrevItem" }                  ## item précédent
    KeyPress = "Ctrl N" { Actions = "NextItem" }              ## item suivant
    KeyPress = "Ctrl P" { Actions = "PrevItem" }              ## item précédent
    KeyPress = "Left" { Actions = "LeaveSubmenu" }             ## sortir du le
sous-menu
    KeyPress = "Right" { Actions = "EnterSubmenu" }           ## entrer dans le
sous-menu
    KeyPress = "Return" { Actions = "Select" }                 ## sélectionner
    KeyPress = "space" { Actions = "Select" }                  ## sélectionner
    KeyPress = "Escape" { Actions = "Close" }                  ## fermer
    KeyPress = "Q" { Actions = "Close" }                       ## fermer
}

```

comme vous pouvez le constater, la syntaxe est plutôt simple à comprendre. il vous permet de contrôler l'intégralité de PeKwm depuis votre clavier.

⚠ pensez à bien vérifier votre fichier pour éviter la présence de doublons lors de la modification.

ce fichier est complété par le fichier "mouse".

~/.pekwm/mouse

configuration des actions de la souris sur les fenêtres, le bureau, les menus ...

```
## valeurs ##
## ButonPress:      un simple clic
## ButtonRelease:   la commande est activée au relachement de clic
## DoubleClick:     double-clic
## Motion:          maintient du clic + déplacement
## Enter:            définir le comportement lors d'une entrée dans une zone
d'action (bouton, zone de texte)
## Leave:           définir le comportement lors d'une sortie d'une zone
d'action
## EnterMoving:     définir le comportement lorsqu'une fenêtre est amenée au
bord de l'écran
##
## boutons ##
## 1: clic gauche
## 2: clic central
## 3: clic droit
## 4: scroll vers le haut
## 5: scroll vers le bas
##
## actions de base ##
## Focus: donne le focus
## Raise: met au premier plan
## ActivateClient: active le client dans un groupe
## Move: déplace
## Resize: redimensionne
## Close: ferme
##
#####
## debut de la configuration
#####
## actions sur la barre de titre des fenêtres classiques
FrameTitle {
    ButtonRelease = "1" { Actions = "Raise; Focus; ActivateClient" } ##
met au premier plan, donne le focus, active
    ButtonRelease = "Mod1 1" { Actions = "Focus; Raise" }          ##
donne le focus, met au premier plan
    ButtonRelease = "Mod4 1" { Actions = "Focus; Raise" }          ##
donne le focus, met au premier plan
    ButtonRelease = "2" { Actions = "ActivateClient" }             ##
active le client
    ButtonRelease = "Mod4 3" { Actions = "Close" }                 ##
fermer
    ButtonRelease = "3" { Actions = "ShowMenu Window" }           ##
}
```

```

afficher le menu client
    ButtonRelease = "4" { Actions = "ActivateClientRel 1" }      ##
active le client de droite dans un groupe
    ButtonRelease = "5" { Actions = "ActivateClientRel -1" }    ##
active le client de gauche dans un groupe
    ButtonRelease = "Mod1 4" { Actions = "SendToWorkspace Next;
GotoWorkspace Next" }      ## deplace et suit le client vers le bureau
suivant
    ButtonRelease = "Mod1 5" { Actions = "SendToWorkspace Prev;
GotoWorkspace Prev" }      ## deplace et suit le client vers le bureau
suivant
    ButtonRelease = "Mod1 Shift 4" { Actions = "SendToWorkspace PrevV;
GotoWorkspace PrevV" } ## deplace et suit le client vers le bureau vertical
précédent
    ButtonRelease = "Mod1 Shift 5" { Actions = "SendToWorkspace NextV;
GotoWorkspace NextV" } ## deplace et suit le client vers le bureau vertical
suivant
    ButtonRelease = "Ctrl 4" { Actions = "MoveClientRel 1" }      ##
deplace le client vers la droite dans un groupe
    ButtonRelease = "Ctrl 5" { Actions = "MoveClientRel -1" }    ##
deplace le client vers la gauche dans un groupe
    ButtonRelease = "Ctrl Mod1 1" { Actions = "Focus; Raise True" } ##
donne le focus, garde au premier plan
    DoubleClick = "2" { Actions = "Toggle Shaded" }              ##
enroule/deroule
    DoubleClick = "Mod1 2" { Actions = "Toggle Shaded" }          ##
enroule/deroule
    DoubleClick = "1" { Actions = "MaxFill True True" }           ##
maximise
    DoubleClick = "Mod1 1" { Actions = "Toggle Maximized True True" } ##
maximise/restaure
    Motion = "1" { Threshold = "4"; Actions = "Raise; Move" }      ##
met au premier plan, deplace
    Motion = "Mod1 1" { Threshold = "4"; Actions = "Raise; Move" } ##
..
    Motion = "Mod4 1" { Threshold = "4"; Actions = "Raise; Move" } ##
..
    Motion = "2" { Threshold = "4"; Actions = "GroupingDrag True" } ##
grouper
    Motion = "Mod1 3" { Actions = "Resize" }                      ##
redimensionne
    # commenter ou effacer la ligne suivante si vous désirez utiliser
clic-to-focus
    Enter = "Any Any" { Actions = "Focus" }                      ##
donne le focus au survol du pointeur
}

## actions sur la barre de titre d'autres fenêtres ( commande rapide, menu
..)
OtherTitle {
    ButtonRelease = "1" { Actions = "Raise; Focus" }              ##

```

```
donne le focus, met au premier plan
    ButtonRelease = "2" { Actions = "Focus" } ##
donne le focus
    ButtonRelease = "3" { Actions = "Close" } ##
ferme
    ButtonRelease = "Mod4 3" { Actions = "ShowMenu Window" } ##
affiche le menu client
    ButtonRelease = "Mod1 4" { Actions = "SendToWorkspace Next;
GotoWorkspace Next" } ## deplace et suit le client vers le bureau
suivant
    ButtonRelease = "Mod1 5" { Actions = "SendToWorkspace Prev;
GotoWorkspace Prev" } ## deplace et suit le client vers le bureau
précédent
    ButtonRelease = "Mod1 Shift 4" { Actions = "SendToWorkspace PrevV;
GotoWorkspace PrevV" } ## deplace et suit le client vers le bureau vertical
précédent
    ButtonRelease = "Mod1 Shift 5" { Actions = "SendToWorkspace NextV;
GotoWorkspace NextV" } ## deplace et suit le client vers le bureau vertical
suivant
    Motion = "1" { Threshold = "4"; Actions = "Raise; Move" } ##
met au premier plan, déplace
    Motion = "Mod1 1" { Threshold = "4"; Actions = "Raise; Move" } ##
..
    Motion = "Mod4 1" { Threshold = "4"; Actions = "Raise; Move" } ##
..
    # commenter ou effacer la ligne suivante si vous désirez utiliser
clic-to-focus
    Enter = "Any Any" { Actions = "Focus" } ##
donne le focus au survol du pointeur
}

## actions sur les bordures
Border {
    TopLeft {
        # Remove the following line if you want to use click to
focus.
        ## commenter ou effacer la ligne suivante si vous désirez
utiliser clic-to-focus
        Enter = "Any Any" { Actions = "Focus" }
        ButtonPress = "1" { Actions = "Focus; Resize TopLeft" } }
    Top {
        # Remove the following line if you want to use click to
focus.
        Enter = "Any Any" { Actions = "Focus" }
        ButtonPress = "1" { Actions = "Focus; Resize Top" } }
    TopRight {
        # Remove the following line if you want to use click to
focus.
        Enter = "Any Any" { Actions = "Focus" }
        ButtonPress = "1" { Actions = "Focus; Resize TopRight" } }
    Left {
```

```

        # Remove the following line if you want to use click to
focus.
        Enter = "Any Any" { Actions = "Focus" }
        ButtonPress = "1" { Actions = "Focus; Resize Left" } }
    Right {
        # Remove the following line if you want to use click to
focus.
        Enter = "Any Any" { Actions = "Focus" }
        ButtonPress = "1" { Actions = "Focus; Resize Right" } }
    BottomLeft {
        # Remove the following line if you want to use click to
focus.
        Enter = "Any Any" { Actions = "Focus" }
        ButtonPress = "1" { Actions = "Focus; Resize BottomLeft" } }
    Bottom {
        # Remove the following line if you want to use click to
focus.
        Enter = "Any Any" { Actions = "Focus" }
        ButtonPress = "1" { Actions = "Focus; Resize Bottom" } }
    BottomRight {
        # Remove the following line if you want to use click to
focus.
        Enter = "Any Any" { Actions = "Focus" }
        ButtonPress = "1" { Actions = "Focus; Resize BottomRight" }
    }
}

## actions sur les bords d'écran
ScreenEdge {
    ## bord inférieur de l'écran
    Down {
        Enter = "Mod1 Any" { Actions = "GoToWorkspace Down" }
    }
    ## va au bureau du bas
    ButtonRelease = "3" { Actions = "ShowMenu Root" }
    ## affiche le menu principal
    ButtonRelease = "2" { Actions = "ShowMenu Goto" }
    ## affiche le menu goto
    ButtonRelease = "1" { Actions = "GoToWorkspace Down" }
    ## va au bureau du bas
    ButtonRelease = "Mod4 2" { Actions = "ShowMenu GotoClient" }
    ## affiche le menu goto client
    ButtonRelease = "4" { Actions = "GoToWorkspace Up" }
    ## va au bureau du haut
    ButtonRelease = "5" { Actions = "GoToWorkspace Down" }
    ## va au bureau du bas
    ButtonRelease = "Mod1 4" { Actions = "GoToWorkspace PrevV" }
    ## va au bureau vertical précédent
    ButtonRelease = "Mod1 5" { Actions = "GoToWorkspace NextV" }
    ## va au bureau vertical suivant
    EnterMoving = "Any Any" { Actions = "WarpToWorkspace Down" }
    ## deplace et suit le client vers le bas

```

```
}
## bord supérieur de l'écran
Up {
    Enter = "Mod1 Any" { Actions = "GoToWorkspace Up" }
    ButtonRelease = "3" { Actions = "ShowMenu Root" }
    ButtonRelease = "2" { Actions = "ShowMenu Goto" }
    ButtonRelease = "1" { Actions = "GoToWorkspace Up" }
    ButtonRelease = "Mod4 2" { Actions = "ShowMenu GotoClient" }
    ButtonRelease = "4" { Actions = "GoToWorkspace Up" }
    ButtonRelease = "5" { Actions = "GoToWorkspace Down" }
    ButtonRelease = "Mod1 4" { Actions = "GoToWorkspace PrevV" }
    ButtonRelease = "Mod1 5" { Actions = "GoToWorkspace NextV" }
    EnterMoving = "Any Any" { Actions = "WarpToWorkspace Up" }
}
## bord gauche de l'écran
Left {
    Enter = "Mod1 Any" { Actions = "GoToWorkspace Left" }
    ButtonRelease = "3" { Actions = "ShowMenu Root" }
    ButtonRelease = "1" { Actions = "GoToWorkspace Left" }
    DoubleClick = "1" { Actions = "GoToWorkspace Left" }
    ButtonRelease = "2" { Actions = "ShowMenu Goto" }
    ButtonRelease = "Mod4 2" { Actions = "ShowMenu GotoClient" }
    ButtonRelease = "4" { Actions = "GoToWorkspace Right" }
    ButtonRelease = "5" { Actions = "GoToWorkspace Left" }
    ButtonRelease = "Mod1 4" { Actions = "GoToWorkspace Next" }
    ButtonRelease = "Mod1 5" { Actions = "GoToWorkspace Prev" }
    EnterMoving = "Any Any" { Actions = "WarpToWorkspace Left" }
}
## bord droit de l'écran
Right {
    Enter = "Mod1 Any" { Actions = "GoToWorkspace Right" }
    ButtonRelease = "3" { Actions = "ShowMenu Root" }
    ButtonRelease = "1" { Actions = "GoToWorkspace Right" }
    DoubleClick = "1" { Actions = "GoToWorkspace Right" }
    ButtonRelease = "2" { Actions = "ShowMenu Goto" }
    ButtonRelease = "Mod4 2" { Actions = "ShowMenu GotoClient" }
    ButtonRelease = "4" { Actions = "GoToWorkspace Right" }
    ButtonRelease = "5" { Actions = "GoToWorkspace Left" }
    ButtonRelease = "Mod1 4" { Actions = "GoToWorkspace Next" }
    ButtonRelease = "Mod1 5" { Actions = "GoToWorkspace Prev" }
    EnterMoving = "Any Any" { Actions = "WarpToWorkspace Right" }
}
}

## actions sur les clients (excepté la barre de titre)
Client {
    # commenter ou effacer la ligne suivante si vous désirez utiliser
    clic-to-focus
    ButtonPress = "1" { Actions = "Focus" }
    # décommenter la ligne suivante si vous désirez utiliser
```

```

clic-to-focus
    # ButtonPress = "1" { Actions = "Focus; Raise" }

    ButtonRelease = "Mod1 1" { Actions = "Focus; Raise" }
    ButtonRelease = "Mod4 1" { Actions = "Lower" }
    ButtonRelease = "Mod1 4" { Actions = "SendToWorkspace Next;
GotoWorkspace Next" }
    ButtonRelease = "Mod1 5" { Actions = "SendToWorkspace Prev;
GotoWorkspace Prev" }
    ButtonRelease = "Mod1 Shift 4" { Actions = "SendToWorkspace PrevV;
GotoWorkspace PrevV" }
    ButtonRelease = "Mod1 Shift 5" { Actions = "SendToWorkspace NextV;
GotoWorkspace NextV" }
    ButtonRelease = "Ctrl Mod1 1" { Actions = "Focus; Raise True" }
    Motion = "Mod1 1" { Threshold = "4"; Actions = "Focus; Raise; Move"
}
    Motion = "Mod4 1" { Threshold = "4"; Actions = "Focus; Raise; Move"
}

    Motion = "Mod1 2" { Threshold = "4"; Actions = "GroupingDrag True" }
    Motion = "Mod1 3" { Actions = "Resize" }
    # commenter ou effacer la ligne suivante si vous désirez utiliser
clic-to-focus.
    Enter = "Any Any" { Actions = "Focus" }
}

## actions sur le bureau
Root {
    ButtonRelease = "3" { Actions = "ShowMenu Root" }          ## menu
principal lors du clic-droit sur le bureau
    ButtonRelease = "2" { Actions = "ShowMenu Goto" }
    ButtonRelease = "Mod4 2" { Actions = "ShowMenu GotoClient" }
    # Horizontal movement
    ButtonRelease = "4" { Actions = "GoToWorkspace Right" }
    ButtonRelease = "5" { Actions = "GoToWorkspace Left" }
    ButtonRelease = "Mod1 4" { Actions = "GoToWorkspace Next" }
    ButtonRelease = "Mod1 5" { Actions = "GoToWorkspace Prev" }
    # Vertical movement
    ButtonRelease = "Shift 4" { Actions = "GoToWorkspace Up" }
    ButtonRelease = "Shift 5" { Actions = "GoToWorkspace Down" }
    ButtonRelease = "Mod1 Shift 4" { Actions = "GoToWorkspace NextV" }
    ButtonRelease = "Mod1 Shift 5" { Actions = "GoToWorkspace PrevV" }
    ButtonRelease = "1" { Actions = "HideAllMenus" }
}

## actions sur les menus (excepté la barre de titre)
Menu {
    Enter = "Any Any" { Actions = "Focus" }
    Motion = "Mod1 1" { Threshold = "4"; Actions = "Focus; Raise; Move"
}
}

```

```
## autres
Other {
    Enter = "Any Any" { Actions = "Focus" }
    ButtonRelease = "3" { Actions = "Close" }
    Motion = "1" { Actions = "Focus; Raise; Move" }
    Motion = "Mod1 1" { Threshold = "4"; Actions = "Focus; Raise; Move" }
}
}
```

comme les actions se répètent, je vous laisse la traduction partielle ;)

~/.pekwm/vars

vous avez sans doute noté au début de certains fichier:

```
INCLUDE = "vars"
```

c'est ici que ça se joue. le fichier **vars** contient vos commandes favorites. il se présente sous une forme minimale mais je vous propose un fichier un peu plus étoffé. à vous de l'adapter à vos besoins. **vars** gère les commandes argumentées:

```
$TERM="urxvtc"
$FILER="urxvtc -name filer -e ranger"
$GFILER="rox-filer"
$WEB="uzbl http://start.arpinux.org"
$GWEB="firefox"
$EDIT="urxvtc -name editor -e vim"
$GEDIT="geany"
$ZIK="urxvtc -name player -e mocp -T transparent-background"
$VOL="urxvtc -e alsamixer"
$JAB="urxvtc -e mcabber"
```

~/.pekwm/menu

le menu de base de PeKwm supporte le menu Debian, les sous-menus et les entrées dynamiques à la manière d'Openbox.

les menus dynamiques lisent le résultat d'une commande ou d'un script passé en argument. vous trouverez un exemple de menu modifié dans mon pack [devart-PeKwm](#).

```
# Menu config for pekwm

# Variables
INCLUDE = "vars"
```



```

RootMenu = "Pekwm" {
    Entry = "terminal" { Actions = "Exec $TERM &" }
    Entry = "execute.." { Actions = "ShowCmdDialog" }

    Separator {}

    Submenu = "aller à..." {
        SubMenu = "bureau" {
            # création du menu goto une fois par rechargement de
            # c'est la façon la plus rapide . par défaut
            COMMAND = "/usr/share/pekwm/scripts/pekwm_ws_menu.sh
            # création du menu goto à chaque ouverture du menu,
            # vous aurez besoin de cette option si vous modifiez
            # de session le nombre de bureaux.
            # ~/.pekwm/config. You will know if you want this.
            # Entry = "" { Actions = "Dynamic
            /usr/share/pekwm/scripts/pekwm_ws_menu.sh send dynamic" }
        }
        Entry = "client.." { Actions = "ShowMenu GotoClient True" }
    }
    Submenu = "Pekwm" {
        Submenu = "themes" {
            Entry { Actions = "Dynamic
            /usr/share/pekwm/scripts/pekwm_themeset.sh /usr/share/pekwm/themes" }
            Entry { Actions = "Dynamic
            /usr/share/pekwm/scripts/pekwm_themeset.sh ~/.pekwm/themes" }
        }
        Entry = "relancer" { Actions = "Reload" }
        Entry = "redemarrer" { Actions = "Restart" }
        Entry = "quitter" { Actions = "Exit" }
    }

    Separator {}

    INCLUDE = "debian-menu"
}

WindowMenu = "menu client" {
    Entry = "visible sur tous les bureaux o/n" { Actions = "Toggle
    Sticky" }
    Entry = "enroule/deroule" { Actions = "Toggle Shaded" }
    Entry = "minimise " { Actions = "Set Iconified" }
    Entry = "commande.." { Actions = "ShowCmdDialog" }

    Submenu = "maximisation" {
        Entry = "complète" { Actions = "Toggle Maximized True True"
    }

```

```

        Entry = "horizontale" { Actions = "Toggle Maximized True
False" }
        Entry = "verticale" { Actions = "Toggle Maximized False True"
}
    }
    Submenu = "remplir l'espace libre" {
        Entry = "complet" { Actions = "MaxFill True True" }
        Entry = "horizontal" { Actions = "MaxFill True False" }
        Entry = "vertical" { Actions = "MaxFill False True" }
    }
    Submenu = "placement" {
        Entry = "met au premier plan " { Actions = "Raise" }
        Entry = "enlève du premier plan" { Actions = "Lower" }
        Entry = "toujours au-dessus" { Actions = "Toggle AlwaysOnTop"
}
        Entry = "toujours en-dessous" { Actions = "Toggle
AlwaysBelow" }
    }
    Submenu = "décorations" {
        Entry = "active/desactive les décorations" { Actions =
"Toggle DecorBorder; Toggle DecorTitlebar" }
        Entry = "active/desactive les bordures" { Actions = "Toggle
DecorBorder" }
        Entry = "active/desactive la barre de titre" { Actions =
"Toggle DecorTitlebar" }
    }
    Submenu = "Skip" {
        Entry = "Menus" { Actions = "Toggle Skip Menus" }
        Entry = "Focus Toggle" { Actions = "Toggle Skip FocusToggle"
}
        Entry = "Snap" { Actions = "Toggle Skip Snap" }
    }
    SubMenu = "envoyer vers..." {
        # création du menu sendto une fois par rechargement de PeK
        # c'est la façon la plus rapide . par défaut.
        COMMAND = "/usr/share/pekwm/scripts/pekwm_ws_menu.sh send"
        # création du menu sendto à chaque ouverture du menu, c'est
la façon lente.
        # vous aurez besoin de cette option si vous modifiez souvent
en cours
        # de session le nombre de bureaux.
        # Entry = "" { Actions = "Dynamic
/usr/share/pekwm/scripts/pekwm_ws_menu.sh send dynamic" }
    }
    Separator {}
    Entry = "fermer" { Actions = "Close" }
    Submenu = "tuer le client" { Entry = "Kill application" { Actions =
"Kill" } }
    }

```

~/.pekwm/autoproperties

le fichier qui personnalise l'affichage de vos applications: ouverture en mode maximisé automatique, ouverture sur tous les bureaux... tout est possible grâce à ce fichier.

```
/*
  Autoproperties.
  ce fichier rassemble des exemples explicites afin de créer votre propre
  fichier
  de règles. il contient également les propriétés par défaut de PeKwm.
  il devrait être une bonne base pour éditer vos propres règles.

  Pour connaître les propriétés d'un fenêtre, vous aurez besoin du programme
  xprop.
  vous pourrez ainsi définir les informations appropriées à entrer dans les
  champs.

  une autre astuce est de vérifier l'emploi de l'entrée ApplyOn.
  les propriétés ne pourront pas s'appliquer sans cette entrée.

  vous pouvez donner plusieurs propriétés à une fenêtre, la plus pertinente
  sera
  prise en compte, les autres ignorées. (voir l'exemple gimp).

  Notez que les entrées par défaut sont commentées, ne pas commenter vos
  autoproperties propres. :)
*/
// Autoproperties. -----
Require {
    Templates = "True"
}

/*
  Grouper les terminaux
  */
# Property = "(term|rxvt),(erm|xvt)" {
#     ApplyOn = "New"
#     Group = "term" {
#         Size = "5"
#         FocusedFirst = "True"
#         Raise = "True"
#     }
# }

/*
  retirer les décorations de la barre personnalisée de firefox.
  */
```

```
Property = "^ (gecko|Gecko|firefox-bin),^Firefox-bin,,^Customize Toolbar\$" {
    ApplyOn = "Start New TransientOnly"
    Border = "False"
    Titlebar = "False"
}

/*
groupe "moz-dl" automatiquement créé au dela de 10 clients de
mozilla-download,
en utilisant la WM_CLASS et titrant la fenêtre du nom du groupe.
deplace cette fenêtre au coin supérieur gauche de l'écran et la fait passer
derrière les autres fenêtres.
*/
# Property = "^mozilla-bin,^Mozilla-bin,,^Saving" {
#     ApplyOn = "New Transient"
#     Group = "moz-dl" { Size = "10" }
#     FrameGeometry = "+0+0"
#     Layer = "Below"
# }

/*
groupe automatiquement au-dela de deux fenêtres de même WM_CLASS répondants
aux propriétés.
les ouvrir sur le bureau 2.
*/
# Property = "^Mozilla,^navigator:browser" {
#     ApplyOn = "Start New Workspace"
#     Workspace = "2"
# }

/*
grouper toutes les fenêtres répondants aux propriétés. lorsqu'une nouvelle
fenêtre est ouverte, qu'elle le soit en background. ne jamais rendre la
fenêtre
active du groupe. cette propriété s'applique à chaque démarrage de pek ou
lorsqu'une nouvelle fenêtre est ouverte.
*/
# Property = "^dillo,^Dillo" {
#     ApplyOn = "Start New"
#     Group = "dillo" { Size = "0"; Behind = "True" }
# }

/*
mettre les fenêtres répondants aux propriétés derrière les autres fenêtres.
les afficher sur tous les bureaux, ne pas les afficher dans les menus de
fenêtres goto.
appliquer cette propriété lors du lancement de pekwm et à chaque ouverture
de fenêtre, y compris les fenêtres secondaires (dans l'exemple, xmms et sa
```

```
playlist).
*/
# Property = ".*,^xmms" {
#     ApplyOn = "Start New Transient"
#     Layer = "Desktop"
#     Sticky = "True"
#     Skip = "Menus FocusToggle Snap"
# }

/*
retirer les fenêtres secondaires de gimp du menu, n'afficher que la
fenêtre principale. utiliser le WM_WINDOW_ROLE pour faire la distinction.

en premier, nous séparons les fenêtres pour ne pas les mélanger. cette
propriété
dit à pek d'ignorer la 'toolbox' dans les propriétés suivantes. sinon, la
'toolbox'
sera traité comme les autres fenêtres, ce que nous voulons éviter.
*/
# Property = "^gimp,^Gimp,gimp-toolbox" {
#     ApplyOn = "Start New"
# }
/*
la boite de dialogue de découpage reste au-dessus des autres, dans le
coin supérieur gauche mais n'est pas visible dans le menu goto.
*/
# Property = "^gimp,^Gimp,gimp-crop-tool" {
#     ApplyOn = "Start New"
#     Layer = "OnTop"
#     FrameGeometry = "+0+0"
#     Skip = "Menus"
# }
/*
le reste des fenêtres ne doit pas être affiché dans les menus non plus.
*/
# Property = ".gimp,^Gimp" {
#     ApplyOn = "Start New"
#     Skip = "Menus";
# }

/*
tout ceci devrait commencer à avoir du sens pour vous :).
*/
# Property = "^gkrellm,^Gkrellm" {
#     ApplyOn = "Start New"
#     Sticky = "True"
#     Skip = "Menus FocusToggle"
#     Layer = "Desktop"
# }
```

```
/*
    quelques applications standard, xclock xload et xbiff. en plus de ce
    que vous avez déjà vu, les fenêtres apparaissent sans décorations.
    on utilise aussi l'argument de 'geometry' au complet, définissant la
    taille de la fenêtre ainsi que sa position sur l'écran.
*/
# Property = "^xclock,^XClock" {
#     ApplyOn = "Start New"
#     ClientGeometry = "120x137+0-137"
#     Border = "False"; Titlebar = "False"
#     Sticky = "True"
#     Layer = "Desktop"
#     Skip = "Menus FocusToggle Snap"
# }
#
# Property = "^xload,^XLoad" {
#     ApplyOn = "Start New"
#     ClientGeometry = "560x137+120-137"
#     Border = "False"; Titlebar = "False"
#     Sticky = "True"
#     Layer = "Desktop"
#     Skip = "Menus FocusToggle Snap"
# }
#
# Property = "^xbiff,^XBiff" {
#     ApplyOn = "Start New"
#     ClientGeometry = "120x137-120-137"
#     Border = "False"; Titlebar = "False"
#     Sticky = "True"
#     Layer = "Desktop"
#     Skip = "Menus FocusToggle Snap"
# }

// fin des autoproprieties. -----

TypeRules {
    /*
        les propriétés appliquées à des types de fenêtres.
        la fenêtre de bureau dans nautilus doit recouvrir la "root window"
        mais se placer derrière toutes les autres, ne pas s'afficher dans
        le menu pek, et ne pas adhérer.
    */
    Property = "DESKTOP" {
        FrameGeometry = "0x0+0+0"
        Titlebar = "False"
        Border = "False"
        Sticky = "True"
        Skip = "FocusToggle Menus Snap"
        Layer = "Desktop"
        Focusable = "False"
    }
}
```

```

    }
    Property = "DOCK" {
        Titlebar = "False"
        Border = "False"
        Sticky = "True"
        Layer = "Dock"
        Skip = "FocusToggle Menus"
        Focusable = "False"
    }
    Property = "TOOLBAR" {
        Skip = "FocusToggle Menus Snap"
    }
    Property = "MENU" {
        Titlebar = "False"
        Border = "False"
        Skip = "FocusToggle Menus Snap"
    }
    Property = "UTILITY" {
    }
    Property = "SPLASH" {
        Titlebar = "False"
        Border = "False"
        Layer = "OnTop"
    }
    Property = "DIALOG" {
        Layer = "OnTop"
    }
    Property = "NORMAL" {
    }
}

/*
les propriétés des titres dans les barres de titres:
raccourcir les titres des navigateurs web.
*/
# TitleRules {
#     Property = "^dillo,^Dillo" {
#         Rule = "/Dillo: (.*)/\\1 - dillo/"
#     }
#     Property = "^firefox-bin,^Firefox-bin" {
#         Rule = "/(.*) - Mozilla Firefox/MF: \\1/"
#     }
# }

// End of titlerules. -----

/*
la configuration des applications du harbour.
obpager se positionne en dernier, la charge cpu en premier et wmd au

```

```

centre.
*/
# Harbour {
#     Property = "^obpager,^obpager" {
#         Position = "-1"
#     }
#     Property = ".*,.*cpuload" {
#         Position = "1"
#     }
#     Property = ".*,^wmnd" {
#         Position = "0"
#     }
# }

// End of harbour order rules. -----

/*
et maintenant, pour finir, si vous avez un thème qui le supporte,
la section "DecorRules {}".

ici nous indiquons au terminal d'utiliser la décoration TERM de votre theme
votre theme doit supporter cette option.
*/
# DecorRules {
#     Property = "^term,^xterm" {
#         Decor = "TERM"
#     }
# }

```

~/.pekwm/themes ,/scripts, /icons

- le dossier ~/.pekwm/**themes** regroupe les themes que vous avez téléchargé ou créé afin qu'ils puissent être détectés par pekwm
- le dossier ~/.pekwm/**scripts** contient les scripts utilisés pour les menus dynamique
- le dossier ~/.pekwm/**icons** contient certaines images apellées par les thèmes

vous trouverez des exemplaires de ces dossiers dans /usr/share/pekwm.

utilisation

voici quelques exemples expliqués d'utilisation de PeKwm.

grouper les fenêtres

PeKwm supporte les groupes de clients dans un même fenêtre. les clients sont organisé en onglets.

pour grouper les clients:

- avec la **souris**: clic-central maintenu sur la barre de titre puis on glisse vers le client ou groupe de clients auquel on souhaite rattacher le client sélectionné. cette commande est visible dans le fichier [mouse]Motion = "2" { Threshold = "4"; Actions = "GroupingDrag True" }
- avec le **clavier**: activer la chaine 'groupe' avec **[Ctrl+Alt+t]** puis
 - [p] pour attacher le client au prochain cadre
 - [o] pour attacher le client au précédent cadre
 - [i] pour attacher le groupe au prochain cadre
 - [u] pour attacher le groupe au précédent cadre
 - [d] pour detacher le client

les raccourcis clavier sont visibles dans le fichier [keys] section des chaines

```
# grouper les clients
Chain = "Ctrl Mod1 T" {
    KeyPress = "T" { Actions = "Toggle Tagged False" }
    KeyPress = "B" { Actions = "Toggle Tagged True" }
    KeyPress = "C" { Actions = "Unset Tagged" }
    KeyPress = "G" { Actions = "Toggle GlobalGrouping" }
    KeyPress = "M" { Actions = "Toggle Marked" }
    KeyPress = "A" { Actions = "AttachMarked" }
    KeyPress = "D" { Actions = "Detach" }
    KeyPress = "P" { Actions = "AttachClientInNextFrame" }
    KeyPress = "O" { Actions = "AttachClientInPrevFrame" }
    KeyPress = "I" { Actions = "AttachFrameInNextFrame" }
    KeyPress = "U" { Actions = "AttachFrameInPrevFrame" }
```

pour naviguer dans un groupe de clients:

- avec la **souris**:
 - scroll haut/bas sur la barre de titre fait naviguer entre les onglets
 - clic-gauche sur un des titres active le client concerné
- avec le **clavier**: activer la chaine la chaine "actions sur les goupes" avec **[Ctfl+Alt+f]** puis
 - up/down/left/right: deplace le focus dans la direction indiquée
 - 'x' deplace le focus sur le client numéro 'x'
 - [i] active le client de droite
 - [u] active le client de gauche
 - [l] deplace le client à droite
 - [U] deplace le client à gauche

les menus dynamiques

c'est l'équivalent des pipes-menus d'openbox: ça lit un script (sh, perl, curl, py..) et donne le résultat dans une entrée ou un sous-menu. un exemple simple: date et heure dans le menu:

le script:

```
#!/bin/bash
echo "Dynamic {"
echo "Submenu = \"`date +%I:%M`\" {"
echo "  Entry = \"`date +%a %d`\" { Actions = \"Exec roxterm --geometry 65x37+565+0 -e cal -y \" } "
echo "}"
```

ce script indique l'heure dans le menu et la date dans un sous-menu. si on clic, un terminal s'ouvre avec le calendrier. ce type de script est inclut dans le menu ou un sous-menu de PeKwm avec ce code:

```
Entry = "" { Actions = "Dynamic ~/.pekwm/scripts/pekwm_time.sh" }
```

personnalisation

l'apparence de PeKwm est géré par GTK et PeK lui-même.

pour les **thèmes gtk**

il suffit de les placer dans votre ~/.themes. puis de choisir une appli genre gtkchooser ou lxappearance pour définir votre thème gtk. si vous désirez passez directement par PeKwm pour gérer l'apparence gtk, il vous faudra installer le paquet [pekwm_menu_tools](#) (par compilation: dépend de libboost1.35-dev, libboost-filesystem1.35-dev, libgtk2.0-dev) puis utiliser le script **pekwm_gtk_menu** afin de pouvoir la lancer avec la section:

```
Submenu = "gtk themes" {
  Entry { Actions = "Dynamic ~/.pekwm/scripts/pekwm_gtk_menu -l" }
}
```

pour les **thèmes pekwm**,

il faut les placer dans /usr/share/pekwm/themes ou ~/.pekwm/themes où ils pourront être détectés par le script pekwm_themeset.sh. ce script est inclus dans pekwm et est lancé par la commande de menu

```
Submenu = "pek themes" {
  Entry { Actions = "Dynamic /usr/share/pekwm/scripts/pekwm_themeset.sh /usr/share/pekwm/themes" }
  Entry { Actions = "Dynamic /usr/share/pekwm/scripts/pekwm_themeset.sh ~/.pekwm/themes" }
```

pour les **icones**, la **police** ...

c'est là qu'une application comme lxappearance est utile .. à moins que vous ne désiriez gérer votre machine qu'en cli .. dans ce cas, ce wiki ne vous apprendra rien ;).

conclusion

je dirais que pekwm est le wm le plus complexe que j'ai eu à configurer si on veut explorer toutes ses options. cependant, pour obtenir un système fonctionnel et performant, il n'est pas nécessaire de tout éditer.

le système de gestion des clients en groupe est des plus pratique et on arrive assez vite à autant d'ergonomie que dans un tiling wm. bien sûr, les raccourcis clavier et le principe des chaînes de raccourcis, lorsqu'on l'a pris en main, devient très vite indispensable.. et le retour est dur sur gnome ou xfce ;).

contributeur: [arpinux](#)

sources: [wiki archlinux](#) + post sur [forum crunchbang](#) + [documentation officielle](#)

From:

<http://arpinux.org/x/> - **arpinux wiki**

Permanent link:

<http://arpinux.org/x/doku.php/wms:pekwm>

Last update: **2013/03/23 17:57**

