Modifiers

Many users have had some difficulty with setting up modifiers for StumpWM keybindings. This is caused by a combination of how StumpWM handles modifiers and the default modifiers list for many users' X servers.

- My "Super" key doesn't work!

  This is most likely caused by having the Hyper and Super keys listed as the same modifier in the modifier list.

```
$ xmodmap
xmodmap:  up to 3 keys per modifier, (keycodes in parentheses):

shift      Shift_L (0x32),  Shift_R (0x3e)
lock       Caps_Lock (0x42)
control    Control_L (0x25),  Control_R (0x6d)
mod1       Alt_L (0x40),  Alt_R (0x71),  Meta_L (0x9c)
mod2       Num_Lock (0x4d)
mod3
mod4       Super_L (0x7f),  Hyper_L (0x80)
mod5       Mode_switch (0x5d),  ISO_Level3_Shift (0x7c)
```

  The problem is in the line beginning with "mod4". The way to set up the modifier list correctly is to have just the Super key as the mod4 modifier. The following xmodmap commands will do just that.

```
# clear out the mod4 modifier
$ xmodmap -e 'clear mod4'
$ xmodmap
xmodmap:  up to 3 keys per modifier, (keycodes in parentheses):

shift      Shift_L (0x32),  Shift_R (0x3e)
lock       Caps_Lock (0x42)
control    Control_L (0x25),  Control_R (0x6d)
mod1       Alt_L (0x40),  Alt_R (0x71),  Meta_L (0x9c)
mod2       Num_Lock (0x4d)
mod3
mod4
mod5       Mode_switch (0x5d),  ISO_Level3_Shift (0x7c)

# add Super as a mod4 modifier
$ xmodmap -e 'add mod4 = Super_L'
$ xmodmap
xmodmap:  up to 3 keys per modifier, (keycodes in parentheses):

shift      Shift_L (0x32),  Shift_R (0x3e)
lock       Caps_Lock (0x42)
control    Control_L (0x25),  Control_R (0x6d)
mod1       Alt_L (0x40),  Alt_R (0x71),  Meta_L (0x9c)
mod2       Num_Lock (0x4d)
mod3
mod4       Super_L (0x73),  Super_L (0x7f)
mod5       Mode_switch (0x5d),  ISO_Level3_Shift (0x7c)
```

You can automate this by storing the commands in a file and calling xmodmap when you start your X session.

```
$ cat ~/.Xmodmap
clear mod4
add mod4 = Super_L
```

If you use startx, modify your '~/.xsession' or '~/.xinitrc' file.

```
$ cat ~/.xsession
#!/bin/sh

xmodmap ~/.Xmodmap
exec /usr/bin/stumpwm
```

If you use a settings daemon from one of the major desktop environments (Gnome,KDE, or Unity) you may be able to set keyboard modifiers from their respective configuration GUIs. If not, xmodmap should always work if invoked at the right place.

- Handling Meta and Alt: when do I use M- and A-?

  If you have no Meta keys defined (see the output of the xmodmap command), then StumpWM will treat the M- prefix in keybindings to mean Alt. However, if there are Meta keys defined, then the M- prefix refers to them, and the A- prefix refers to Alt.

  Most users will simply use M- to refer to their Alt keys. However, users that define separate Meta and Alt keys will use M- to refer to the former, and A- to refer to the latter.

- How can I set up a Hyper key and use it with StumpWM?

  To set up a Hyper key, you need to do two things: bind a physical key to be a Hyper key, and add that key to the modifiers list.

  The following example shows how to bind the control key at the bottom-left of most keyboards to be Hyper. This is useful if you've made Caps Lock into a control key, and have no use for the bottom-left key.

```
$ xmodmap -e 'keycode 37 = Hyper_L'
$ xmodmap -e 'clear mod5'
$ xmodmap -e 'add mod5 = Hyper_L'
```

To use a different key for Hyper, replace the keycode “37” above. Use the xev program to see the keycode that any physical key has. Refer to the section above on setting up the Super key to see how to automate setting the Hyper key when you start X.

Now you can use H- as a prefix in StumpWM bindings.

```
(define-key *top-map* (kbd "H-RET") "fullscreen")
(define-key *top-map* (kbd "H-Left") "gprev")
(define-key *top-map* (kbd "H-Right") "gnext")
(define-key *top-map* (kbd "H-TAB") "other")
```

Since essentially no programs have Hyper bindings, you can safely bind commands to the **top-map**.