

Der Window Manager Improved 2 (*wmii*) [1] ist ein Fenstermanager, der sehr minimalistisch und gleichzeitig hoch konfigurierbar ist. Er eignet sich besonders für Nutzer, die verstärkt die Shell nutzen und weniger bis nie die Maus. Um *wmii* besser nutzen zu können, wird die Konfiguration hier näher vorgestellt.

wmii liegt in vielen Distributionen schon im Repository vor. Auf die Installation wird in diesem Artikel nicht näher eingegangen, da sich schon der Artikel „Der Windowmanager *wmii*“ in freiesMagazin 07/2007 [2] damit beschäftigt. Die Versionen 2, 3.5 sowie 3.6 werden aber immer noch über die gleichen Tasten gesteuert, womit der Artikel von 2007 auch heute noch Gültigkeit besitzt. Es existiert zwar schon eine neue Version (3.9b1), allerdings liegt diese nur in einem Betastatus vor und kann nur über das Versionsverwaltungsprogramm Mercurial [3] bezogen werden.

Individuelle Konfiguration für jeden Benutzer

Da am Anfang die Konfigurationsdatei **wmiirc**, die man benötigt, um eigene Einstellungen vorzunehmen, in **/etc/X11** oder direkt in **/etc** liegt, kopiert man sie sich einfach nach der Installation von *wmii* ins Homeverzeichnis:

```
$ cp -r /etc/X11/wmii-3.5/* ~/.wmii~  
-3.5
```

oder

```
$ cp -r /etc/wmii-3.5/* ~/.wmii~  
-3.5/rc.wmii
```

Danach sollte man noch die Besitzrechte anpassen:

```
$ chown -R $USER:'id -ng' ~/.wmii~  
-3.5
```

Nun kann man den Fenstermanager individuell gestalten. Die Datei **wmiirc** liegt im Folgenden im versteckten Ordner **.wmii-3.5** im Homeverzeichnis des Benutzers (**/home/BENUTZERNAME**).

Farben, Schriften und Hintergrund

Farben

Um das Standardblau z. B. durch ein dunkleres Rot zu ersetzen, öffnet man die eben kopierte **wmiirc** mit dem Lieblingseditor und findet im Block „Color tuples“ Folgendes:

```
# Colors tuples: "<text> <background> <border>"  
WMII_NORMCOLORS='#888888 #222222 #333333'  
WMII_FOCUSCOLORS='#ffffff #285577 #4c7899'
```

Der Ausdruck **#888888** ist dabei eine hexadezimale Farbdefinition [4] und steht in diesem Fall für ein Rot-/Grün-/Blauanteil von 136, was einem Grauton entspricht. **WMII_NORMCOLORS** steht dabei für die Farbe, in der *wmii* die Fenster darstellt. **text** bedeutet, in welcher Farbe der Text dargestellt wird; **background**, wie der Hintergrund

gezeichnet wird und **border**, wie die einzelnen Trennlinien der Fenster dargestellt werden. **WMII_FOCUSCOLORS** wird benutzt, wenn ein Fenster den Fokus der Maus hat.

Dies ersetzt man nun nach Belieben. Die Farben können z. B. durch Gimp (ein Bild-Manipulationsprogramm), das auch in den meisten Distributionen in den Paketquellen enthalten ist, bestimmt werden. Dazu klickt man einfach nach dem Start von Gimp auf eine der zwei Farbflächen in der Toolbox.

Auch den Standardhintergrund kann man, wenn man noch kein Bild hat, bestimmen:

```
WMII_BACKGROUND='#333333'
```

Schriften

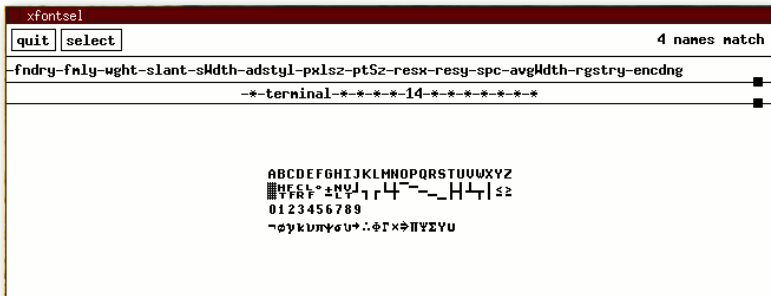
Die Schrift bestimmt man durch die Variable **WMII_FONT**. Das könnte beispielsweise einfach nur

```
WMII_FONT='fixed'
```

oder auch

```
WMII_FONT='-*-fixed-medium-r~  
-*-*12-*-*-*-*-*'
```

sein. Bestimmen kann man die Schrift mit dem Programm **xfontsel**, das man mit dem gleichen Namen über die Paketverwaltung installieren kann.



Das Programm *xfontsel* zum Bestimmen von Schriften. 🔍

Um eine Schriftart, wie z. B. `--terminal--*--*--*--14--*--*--*--*--*` festzulegen, klickt man einmal mit der linken Maustaste auf „flmy“ und wählt dann bei gedrückter Maustaste „terminal“ aus – sofern diese Schriftart installiert ist. Nach gleicher Methode wählt man die Schriftgröße „14“ aus „pxlsz“ aus. Weitere Informationen über *xfontsel* erhält man über **man xfontsel**.

Hintergrundbild

Um ein Hintergrundbild zu definieren, kann man z. B. den kleinen Bildbetrachter **feh**, den man durch mit diesem Namen aus den Paketquellen installieren kann, benutzen. Nach der Installation benutzt man einen der folgenden Befehle:

```
$ feh --bg-scale /voller/pfad/zum/␣
bild.endung
```

oder auch

```
$ feh --bg-center /voller/pfad/zum/␣
bild.endung
```

Mit dem Parameter `--bg-scale` wird das Hintergrundbild skaliert, also an die Auflösung des Monitors angepasst, während `--bg-center` das Bild zentriert. Dabei speichert **feh** das Bild in der versteckten Datei **.fehbg** im Homeverzeichnis ab. Um das Hintergrundbild bei jedem Starten der grafischen Oberfläche direkt anzeigen zu lassen, kommentiert man die folgende Zeile durch eine Raute # aus

```
# xsetroot -solid "$WMII_BACKGROUND" &
```

und fügt dafür diese in **wmiirc** ein:

```
eval 'cat ~/.fehbg' &
```

Zur näheren Beschreibung kann man die Manpage von **feh** konsultieren:

```
$ man feh
```

Es gibt natürlich noch andere Programme, um den Hintergrund zu ändern. Diese sind im Wiki von ubuntuusers.de [5] oder in den FAQ von suckless.org [6] beschrieben.

Programmeinteilung

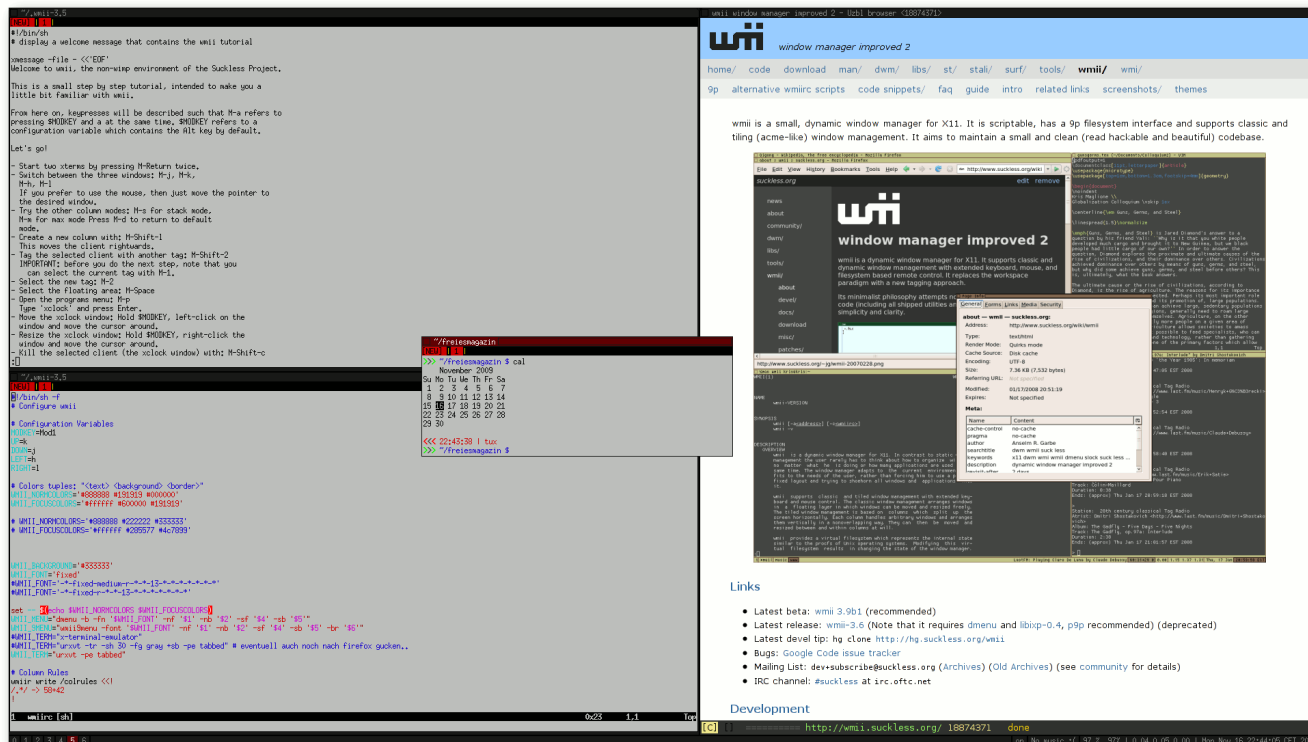
Startet man **wmii** und danach seine Lieblingsprogramme, sammeln sich diese – sofern man sie nicht auf einen anderen Desktop gelegt hat – auf Desktop 1. Möchte man verschiedene Programme auf einem individuellem Desktop (auch „tag“ genannt) sammeln, gibt es dazu die „Tagging Rules“ in der **wmiirc**.

Diese sehen z. B. so aus:

```
# Tagging Rules
wmii write /tagrules <<!
/MPlayer.* / -> ~
/Shiretoko.* / -> 3
/Thunderbird.* / -> 2
/Gimp.* / -> ~+gimp
/Pidgin.* / -> 4+chat
./.* / -> sel
./.* / -> 1
!
```

Zur Erklärung: **wmii write /tagrules «!** bedeutet, dass **wmii** (das ist das Dateisystem-Programm, um **wmii** zu steuern; auf **wmii** wird später noch näher eingegangen) die Taggingrules bis zum nächsten Ausrufezeichen einliest. **MPlayer** (ein Mediaplayer) wird immer auf dem gerade aktivem Desktop im „Floating Mode“ gestartet, d. h. er „schwebt“ wie bei anderen Desktopumgebungen. **Shiretoko** (ein alternativer Name für **Mozilla Firefox**) wird immer auf Desktop 3 platziert, **Thunderbird** auf 2. **Gimp** wird im Floatingmode auf dem Desktop mit dem Namen **gimp** gestartet. **Pidgin** auf Desktop 4 und zusätzlich Desktop **chat**. Dabei läuft auf beiden dieselbe Instanz.

Alles andere wird auf dem gerade aktiven Desktop gestartet. Existiert kein Desktop (wie am Anfang direkt nach dem Start von **wmii**, erkennbar durch den Namen **nil**), wird alles auf Desktop 1 gestartet. Der Floatingmode (dargestellt in den Tagging Rules mit einer Tilde ~) bezeichnet dabei ein Fenster, das über den anderen „schwebt“.



wmii im Tiling und Floating Mode. 🔍

Verknüpfen kann man die Regeln mit einem Plus +. Möchte man also, dass MPlayer immer auf Tag 1 startet, aber zusätzlich im Floatingmode, schreibt man 1+~. Der Ausdruck ~ allein würde MPlayer immer auf dem gerade aktivem Desktop starten. Ändern kann man die Belegung mit **Modkey** + **Shift** + **t** (siehe dazu auch den Artikel aus **freiesMagazin** 07/2007 [1]).

Sehr sinnvoll kann es sein, Programme per Shellskript zu starten. Das funktioniert natürlich auch mit Programmen in der Shell. Weechat (ein IRC-Client für die Shell) [7] will man z. B. auf

Desktop 4 starten. Dazu trägt man einfach **/weechat.* / -> 4** in die Taggingrules ein und startet dann mit einem Terminalemulator, z. B. urxvt (rxvt-unicode) [8], weechat:

```
$ urxvt -T "weechat" -e sh -c "weechat-curses" >/dev/null 2>&1 &
```

Der Parameter **-T** setzt den Titel des Emulators auf „weechat“ und **-e** startet eine Shell mit dem Befehl (**-c**) „weechat-curses“ im Hintergrund. Die Ausgabe (stdin und stderr [9]) wird nach **/dev/null** [10] weitergeleitet.

Tastaturkürzel

In der Funktion „eventstuff()“ kann man sich eigene Tastaturkürzel anlegen. Unter anderem folgende:

```
Key Print
    screenshot scr

Key $MODKEY-Print
    screenshot win

Key $MODKEY-q
    #mocp -r
    echo quit | nc localhost 54311

Key $MODKEY-w
    #mocp -P
    echo pause | nc localhost 54311

Key $MODKEY-e
    #mocp -U
    echo stop | nc localhost 54311

Key $MODKEY-r
    #mocp -f
    echo skip | nc localhost 54311

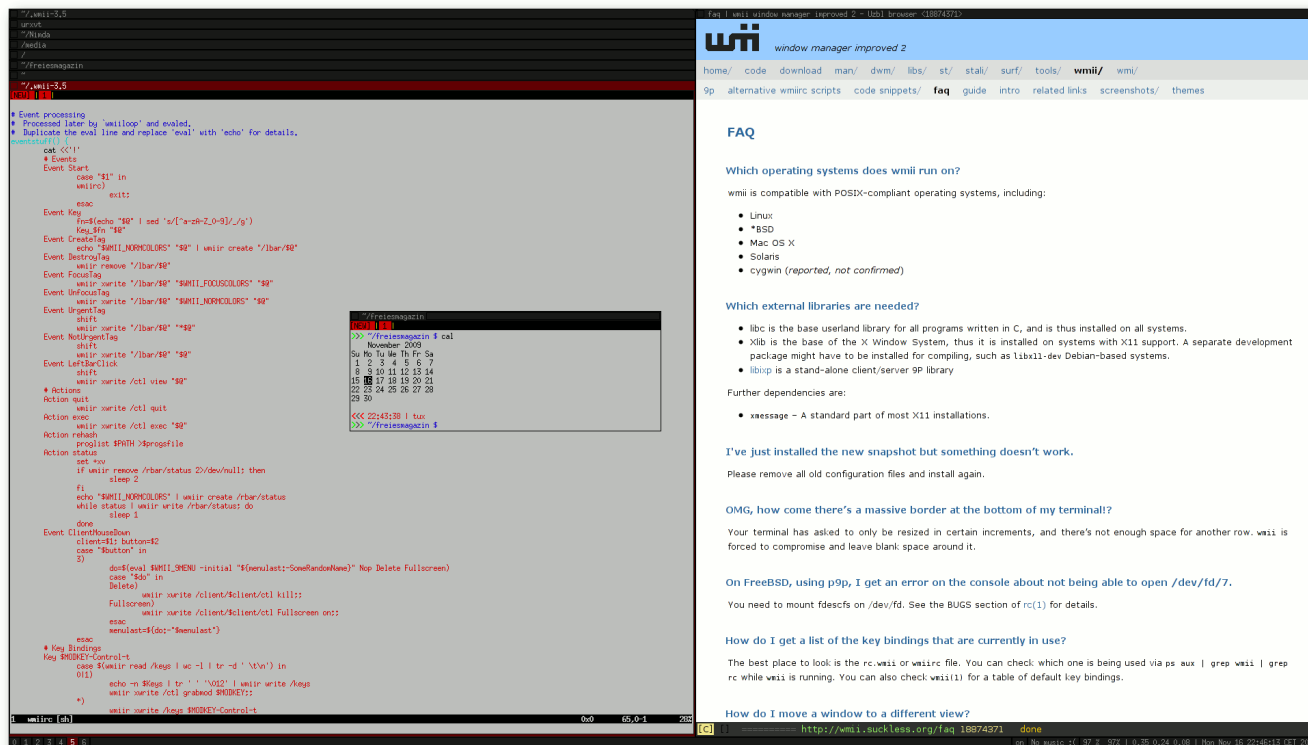
Key $MODKEY-Shift-m
    amixer set Master 0%

Key $MODKEY-Shift-n
    amixer set Master 100%

Key $MODKEY-g
    mocp -G
```

Achtung: Die Definitionen müssen hier zwingend mit Tabulatoren **Tab** eingerückt werden.

screenshot ist ein Programm, das im Wiki von ubuntuusers.de [11] beschrieben ist. Mit mocp kann man Musik hören und somit mocp als Daemon leicht steuern. Die Konfiguration von Music on Console kann man sich im Artikel „MOC’n’Roll: audiophil mit der Konsole“, **freiesMagazin** 02/2008 [12], durchlesen. amixer



wmii im Stacking und Floating Mode. 🔍

ist ein Programm, das die Lautstärke via Shell ändern kann. Eine Oberfläche in der Shell bietet das Programm als **man amixer** bzw. **man alsamixer**. Zuletzt kann man mit

```
$ echo <befehl> | nc <host> <port>
```

das Programm „shell-fm“ steuern, wenn es als Daemon gestartet wurde. shell-fm ist ein Programm, mit dem man Last.fm [13] über die Shell steuern kann. Informationen findet man dazu in der Manpage: **man shell-fm**.

Der Befehl wird dann an das Programm „nc“ (netcat), das auch über die Paketverwaltung bezogen werden kann, weitergeleitet.

Pidgin – Message Notifications

Hier noch eine letzte Programmanbindung, nämlich an Pidgin. Trifft eine neue Meldung ein, bekommt man normalerweise nichts davon mit, da man gerade auf einem anderen Desktop aktiv ist. Daher kann man z. B. das Plugin „Message Notification“ nutzen, das normalerweise bei Pidgin dabei ist. Falls nicht, kann man es durch das Paket **pidgin-plugin-pack** nachinstallieren.

Aktivieren kann man dieses Plugin in Pidgin über „Werkzeuge » Plugins“. Nun wählt man „Message Notification“ aus und setzt bei der Option „Set window manager 'URGENT' hint“ ein Häkchen. Die weitere Konfiguration kann man bei Bedarf ändern.

Trifft nun eine neue Meldung ein, erscheint in der linken Hälfte der Statusbar vor dem Namen des Desktops, auf dem Pidgin läuft, ein Stern „*“.



Statusbar mit eigenen Tags und Pidgin Message Notification. 🔍

Statusbar

Zuletzt noch eine Beschreibung der Statusbar, die auch sehr gut editierbar ist. In dieser kann man sich z. B. die Namen der Desktops, das gerade gespielte Lied (per shell-fm oder mocp), Akkuinformationen des Laptops, Infos aus uptime und die Zeit ausgeben lassen. Kurzum, alle Shellbefehle, die einem sinnvoll und wichtig erscheinen.

Eine minimalistische Statusbar sieht z. B. so aus:

```
# Status Bar Info
status() {
    echo -n $(date)
}
```

Und hier noch eine, die etwas mehr ausgibt:

```
# Status Bar Info
status() {
    echo -n $(echo info '%t by %a (%l) - %R' | \
nc localhost 54311) '|' $(acpi -b | \
sed 's/.*://' | sed 's/.*, //'') '% ' \
$(acpi -b | sed 's/ remaining.*://' | \
sed 's/.*, //'') '|' $(uptime | \
sed 's/.*: //; s/, //g') '|' $(date)
}
```

wmiir

wmiir (laut **man wmiir** „The wmii 9P filesystem client“) stellt das Grundgerüst von *wmii* dar. Mit diesem Programm kann man Informationen von *wmii* bekommen, aber auch gleichzeitig welche „senden“. Man kann also z. B. durch einen einfachen Befehl auch selber „Bars“ (vergleichbar mit der Statusbar) anlegen und auch wieder löschen.

Eigene Bars (hier auf der linken Seite) erstellt man mit

```
$ echo "wmii rockt!" | wmiir create~
/lbar/wmii
```

und löscht sie wieder mit

```
$ wmiir remove /lbar/wmii
```

Anstelle **lbar** kann man auch **rbar** angeben für eine Leiste auf der rechten Seite.

Mit wmiir kann man noch viel mehr anstellen, aber das würde den Rahmen dieses Artikels sprengen.

Weitere Informationen darüber erhält man in der Manpage von wmiir, im ArchLinux-Wiki [14] oder auf suckless.org [15].

wmii „neustarten“

Damit Änderungen an der **wmiirc** wirksam werden, muss man *wmii* nicht komplett neu starten. Es reicht, wenn man aus dem Menü, das beim Drücken von **Modkey** + **A** erscheint, „wmiirc“ auswählt und mit **Enter** bestätigt.

Vor jeder Änderung sollte man allerdings ein Backup von der Datei **wmiirc** machen, denn es erscheint kein Hinweis, wenn ein Fehler in der Konfiguration ist:

```
$ cp ~/.wmii-3.5/wmiirc ~/.wmii~
-3.5/wmiirc_`date +%Y%m%d_%H%M%S`.~
bak
```

LINKS

- [1] <http://de.wikipedia.org/wiki/Wmii>
- [2] <http://www.freiesmagazin.de/freiesMagazin-2007-07>

- [3] <http://de.wikipedia.org/wiki/Mercurial>
- [4] http://de.wikipedia.org/wiki/Hexadezimale_Farbdefinition
- [5] <http://wiki.ubuntuusers.de/Desktop-Hintergrund>
- [6] <http://wmii.suckless.org/faq> 
- [7] <http://de.wikipedia.org/wiki/WeeChat>
- [8] <http://en.wikipedia.org/wiki/Rxvt-unicode> 
- [9] <http://de.wikipedia.org/wiki/Stdin>
- [10] <http://de.wikipedia.org/wiki/dev/null>
- [11] <http://wiki.ubuntuusers.de/Wmii#Bildschirmfotos>
- [12] <http://www.freiesmagazin.de/freiesMagazin-2008-02>
- [13] <http://last.fm>
- [14] <http://wiki.archlinux.org/index.php/Wmii> 
- [15] http://code.suckless.org/contrib/guide/wmii-3/guide-en/guide_en/node7.html 

Autoreninformation

Yannic Hauptenthal nutzt nun schon seit über einem Jahr nur noch GNU/Linux und begeistert sich für die Unixphilosophie, die von suckless.org gut umgesetzt wird, sowie für Bastel-distributionen wie ArchLinux.

Diesen Artikel kommentieren 