

larswm.
Der Fenstermanager
für das Wesentliche.

Johannes Hüsing

ELUG
11.12.2001



Grafische Benutzeroberflächen

Geschichte

- Alan Kay, Xerox PARC, 1970er Jahre: Verknüpfung von grafischem Display, Fenster-, Maus- und Menüsteuerung, objektorientierter Programmierung
- Reimplimentationen: Apple Lisa/Macintosh, OS/2, MS Windows, NeXTStep, Aqua
- Unix/Linux: Traditionelle Differenzierung von grafischem Display, Fenstersteuerung und objektorientierten Frameworks
 1. Grafisches Display (X Window System, Framebuffer, Berlin)
 2. Window Manager (fvwm, Window Maker, Blackbox, Sawfish, larswm)
 3. Objektorientiertes Framework (Qt+KDE, GTK+Gnome)



Vor- und Nachteile des klassischen Designs

- Vorteile:
 - Übersichtlichkeit (vs. virtuelle Konsolen/GNU screen)
 - Grafische Oberfläche möglich, wo sinnvoll (Bsp.: Gimp)
 - Nachteile:
 - Ressourcenverbrauch des X-Servers und eventueller Desktop-Software
 - Manuelles Fenstermanagement, Hantieren mit der Maus
 - Verschwendung von Anzeigefläche durch Menüs, Werkzeugleisten, Icons
- ⇒ Vorteile behalten, Überflüssiges streichen!



Anderer Ansatz

Auf was kann man eher verzichten?

- Icons: werden überdeckt
- Desktopmenüs: Rootfenster überdeckt
- Fensterleisten: Zum Anfassen und für den Fenstertitel, nicht unbedingt wichtig
- Maus: Sinnvoll zum Reinformen, aber man muss loslassen können

Wo ist Intuition wichtig? Wir nehmen

- vi oder Emacs für Texte
- mutt für Mail

⇒ heftig benutzte Anwendungen wenig intuitiv und tastaturorientiert, aber schnell.
Warum dann nicht beim Fenstermanager?



Die Fensterverwaltung

Idee

- Aufgabe des Fenstermanagers
- Und nicht des Mausschublers



Typen von Fenstern

- gekachelt

Ein Fenster links, von oben nach unten reichend. Alle anderen Fenster verkleinert und rechts aufgestapelt.

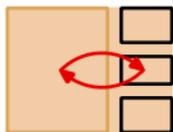
- frei

Neues Fenster so, wie in `.Xresources` definiert



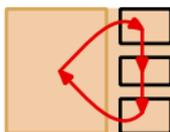
Gekachelte Fenster

- In der Regel ist das aktive Fenster das linke.
- Links klicken auf ein anderes Fenster macht dieses aktiv und rückt es nach links.



- Ctl-Alt-↑ tauscht linkes und rechtes oberes.
- Ctl-Alt-↓ macht den Paternoster im Uhrzeigersinn. Das Fenster unten rechts

kommt nach links und wird aktiv.



freie Fenster

- frei positionierbar
- verschiebbar, streck- und stauchbar: nur mit Tastatur
- freie und gekachelte Fenster bieten zwei verschiedene Ebenen eines Desktops: den Subdesktop



sonstige Fenstereigenschaften

- Werkzeug (Tool)

unverdeckbar, unverrückbar, umflossen von anderen Fenstern. Sollte freies Fenster sein.

- klebrig (sticky)

wird auf jeden Desktop mitgenommen. Kann freies oder gekacheltes Fenster sein.

- treibend (floating)

verdeckt alle anderen Fenster



Die Aufgabenleiste

Lage und AusmaÙe

- Höhe: eine Zeile
- Breite: Display



Inhalt

- Fenstertitel
- Menü
- Desktoptitel
- Desktopeigenschaften



Desktopeigenschaften

- T Aktiver Subdesktop. Gekachelt ist T, frei ist U.
- C Klick wird an Anwendung weitergegeben (und dient nicht nur der Aktivierung des Fensters)
- R
- S Aktiviertes Fenster wird nach vorne geholt.
- B Fensterbewegungen um jeweils 5 % der Bildschirmbreite anstatt um 1 Pixel.
- H Dieser Desktop enthält verdeckte Fenster.



Menü

Acht Menüpunkte – fest verdrahtet

1. *Pass clicks on to client?* Mausclick nicht nur um Fokus zu bekommen, sondern auch für die Anwendung.
2. *Resize window when tiling?*
3. *Retile when focus changes?* Aktiviertes Fenster wird nach vorne geholt.
4. *Show class info on status bar?* In Statuszeile Fensterklasse und Fenstereigenschaften anzeigen.
5. *Current window should be floating?* Aktives Fenster treibend.
6. *Current window should be a tool window?* Aktives Fenster unverdeckbar.
7. *Current window should be sticky?* Aktives Fenster mitwandernd.
8. *Current window should be tiled?* Aktives Fenster stapeln.



Konfiguration

Programme

- sind in Konfigurationsdatei durch aufsteigende Zahl gekennzeichnet, zum Beispiel `larswm.application.2: mozilla`
- können Eigenschaften zugesprochen bekommen, zum Beispiel Tastatur- oder Mauskürzel, zum Beispiel `larswm.application_key.2: m`,
`larswm.application_mod.2: Mod4` ermöglicht Start von Mozilla mit Windows/Linux-Taste und `m`.



Fensterklassen

- sind die Fensterklassen, wie sie von X vergeben werden und mit `xprop` oder `xwininfo` ermittelbar sind, zum Beispiel `larswm.dtclass.2: Gimp`
- können einem Desktop zugeordnet werden, zum Beispiel `larswm.dtnum.2: 1` legt alle Gimp-Fenster auf den *zweiten* Desktop (fängt bei Null an zu zählen)
- können Fenstertypen zugeordnet werden, zum Beispiel `larswm.dotileclass.0: Mozilla-bin` oder `larswm.floatclass.0: Gimp~toolbox`



sonstige Einstellung

- Abstand zwischen gekachelten Fenstern in Pixeln: `larswm.tile_pad`
- Fensterrahmenbreite in Pixeln: `larswm.border`
- Fensterrahmen- (aktiv) und Statuszeilenfarbe: `larswm.foreground`
- Fensterrahmenfarbe (inaktiv): `larswm.background`
- Font in Statusleiste: `larswm.font`
- Anzahl Desktops (bis zu 16): `larswm.desktops`



nützliche Einstellungen

- zeilenorientierte Anwendungen sorgen nicht dafür, dass ein wieder vergrößertes Fenster den alten Inhalt wieder anzeigt \Rightarrow die Standardterminaleinstellung sollte `xterm -e screen bash` lauten, `screen` merkt sich Terminalinhalt
- Neustart des Managers mit `larsremote restart`
- Text auf Statuszeile mit `larsremote message` geben
- für ungestapelte Fenster: in `.Xresources` Lage und Größe eingeben mit `Anwendung*Geometry: 240*120-0-0` (240 Pixel breit, 120 Pixel hoch, rechte untere Ecke)
- Breite der linken Spur lässt sich desktopabhängig einstellen. Nützlich zum Beispiel: Browserfenster breiter, Terminalfenster nicht so breit.



Fazit

Askese ist nicht immer Käse

- Was man gerade braucht, ist schön groß
- sparsam im Verbrauch (Bildschirmfläche, Hauptspeicher, Rechenzeit)
- dadurch schnelles Arbeiten möglich (manche legen sich den Paternoster auf das Mausrad!)
- Abhängigkeiten: libc, Xlibs
- Weiterentwicklung, bis auf Bugfixes, ist eingestellt
- Bestandteil von Debian GNU/Linux (unstable) und FreeBSD (ports)



Verbesserungswürdig

- nicht alle Tastenbelegungen funktionieren überall so wie vorgesehen
- einige Belegungen erfordern Ct1 und Alt – Vorsicht!
- Statuszeile lässt sich nicht verbergen
- Bestimmte Eigenschaften hängen mit der Fensterklasse und nicht mit dem Namen zusammen



Weitere Verweise

- Homepage: <http://www.fnurt.net/larswm>
- Mailingliste: An majordomo@fnurt.net eine Nachricht senden mit der einzigen Zeile `subscribe larswm` im Rumpf
- Fanseite: <http://www.math.fu-berlin.de/~guckes/larswm/>

