

Anhang A. Einrichten von .xinitrc/.xsession

Contributed by Verin.

Die Bedeutung von xinit

Ein window manager ist einfach nur eine Anwendung für X11, wie `netscape` oder `gimp` oder `xterm`. Viele Neulinge in Sachen X11 glauben, dass X11 den window manager ausführt und dass der window manager Programme ausführt. Das ist so nicht richtig. Wenn alles richtig eingestellt ist, kann man alle Anwendungen unter X11 starten, den window manager beenden und einen anderen window manager starten.

Das *wirkliche* Programm was X11 ausführt, welches dann wiederum andere Programme ausführt, ist Ihr `.xinitrc` oder `.xsession` Skript. Wenn X11 gestartet wird, wird das `.xinitrc` oder `.xsession` Skript ausgeführt und wenn das Skript beendet ist, wird auch X11 beendet. Lassen Sie mich dies wiederholen, es ist wichtig: *wenn .xinitrc zu Ende gelaufen ist, endet auch X11*. Dies ist NICHT der Punkt, wo der window manager die Ausführung beendet.

Das Skript

Machen Sie sich zuerst folgendes klar: wenn Sie in einer Shell ein Kommando eingeben, können Sie so lange nichts anderes tun bis dieses Kommando beendet ist. Das `.xinitrc` oder `.xsession` Skript funktioniert genauso. Wenn es während seiner Ausführung auf ein Programm trifft, das lange läuft (so wie die meisten X11-Programme), wartet es genau an dieser Stelle so lange, bis das aufgerufene Programm beendet ist.

Im Idealfall sollte Ihr Skript genau *einen* solchen Punkt haben, an dem es 'wartet', und üblicherweise sollte dies am Ende des Skripts geschehen. Das bedeutet: wenn Sie andere X11-Programme ausführen wollen, bevor Sie diesen Haltepunkt erreichen, sollten diese im Hintergrund laufen. Dies erreichen Sie, indem Sie ans Ende jeder Zeile ein `&` setzen. Wenn Sie also zum Beispiel `xclock` zusätzlich zu allem anderen ausführen wollen, setzen Sie diese Zeile vor Ihren Haltepunkt:

```
xclock &
```

Das nächste was Sie sehen ist der `exec` Befehl, zu dem viele Quellen viele verschiedene Empfehlungen geben. Im Grunde ist dieser Befehl nicht notwendig, es reicht prinzipiell aus wenn Sie ihren Windowmanager in der letzten Zeile des Skripts starten - das Skript wird dann an diesem Punkt wie gewünscht 'warten'.

Warum dann überhaupt der `exec`-Befehl? Nun, angenommen Sie wollen mehrere Zeilen in Ihr Skript eintragen die einen Windowmanager starten, aber es soll nur eine davon funktionieren. Mit `exec` können Sie Ihre auserwählte Zeile ganz nach oben setzen. `exec` bedeutet nämlich folgendes:

„Ersetze 'mich selbst' mit diesem Programm, d.h. starte es und beende mich selbst, sobald das Programm beendet ist.“

Wenn Sie also zum Beispiel eine Zeile `exec wmaker` oberhalb einer Zeile `exec enlightenment` im Skript eintragen, endet das Skript wenn `wmaker` endet - es erreicht nie die nächste Zeile.

Sehen Sie, was ich mit "unnötig" meinte? Sie könnten genausogut eine Menge auskommentierter Windowmanager-Zeilen eintragen, und alles würde ganz genauso funktionieren.

Eine andere Methode

Als Alternative könnten Sie auch den Windowmanager *zuerst* starten, und dessen process ID in einer Umgebungsvariablen speichern:

```
wmaker & wmpid=$!
```

Dies versetzt `wmaker` in den Hintergrund (`&`) und speichert die process ID (`$!`) in einer Variablen (`wmpid`). Dann verwenden Sie `wait`, um den Haltepunkt zu erzeugen:

```
wait $wmpid
```

oder Sie verwenden ein Programm welches Sie immer verwenden wollen, zum Beispiel `gkrellm` als Haltepunkt, indem Sie es einfach nicht in den Hintergrund versetzen. Beachten Sie jedoch in diesem Falle: sobald Sie dieses Programm beenden, endet auch Ihre X11-Session.

Ich persönlich verwende die `wait`-Methode, da ich gern vor dem Start meiner Anwendungen meinen Windowmanager auswähle. Außerdem ändere ich gern zuvor die Einstellungen meines X11-Servers, zum Beispiel Auflösung, Bildschirmschoner oder Fonts. Außerdem stelle ich gern nachher die alten Einstellungen wieder her, um sicherzugehen dass alles wieder im ursprünglichen Zustand ist - zum Beispiel haben Displaymanager manchmal Probleme, den Font-Path zurückzusetzen.

Beispiel A-1. .xinitrc

```
# Bildschirmabschaltung aus- und energy star features einschalten
xset s off
xset dpms 600 60 60

# meine optionalen Fonts zum Font-Pfad hinzufügen
xset +fp "$X_FONTPATH"
xset fp rehash

# aktuelle Umgebungsvariablen speichern, falls Debugging nötig ist
env > ~/.xenv

# window manager
fluxbox & wmpid=$!

bbrun &
wmCalClock &
wmxmms &

# HANG POINT - abwarten, bis der window manager beendet ist
wait $wmpid

# X Standard-Font-Pfad wiederherstellen
xset fp default
```

[Zurück](#)[Advanced Theme Editing](#)[Zum Anfang](#)[Nach vorne](#)[Frequently asked questions \(FAQ\)](#)