Unix I – Einführung in das Unix Betriebssystem

Stephan Thesing

Das XWindow System

- X Client ↔ Server Architektur
- Was passiert beim Start von X?
- Optionen
- Windowmanager

X Client ↔ Server Architektur

X Window ist *das* graphische Benutzerinterface unter Unix.

Es ist hoch portable und läuft auf allen relevanten Unix Dialekten.

Es bietet:

- Programmierung portabler GUIs
- Plattformunabhängigkeit
- Realisiert ein Client↔ Server Modell
- Ist hochanpassbar

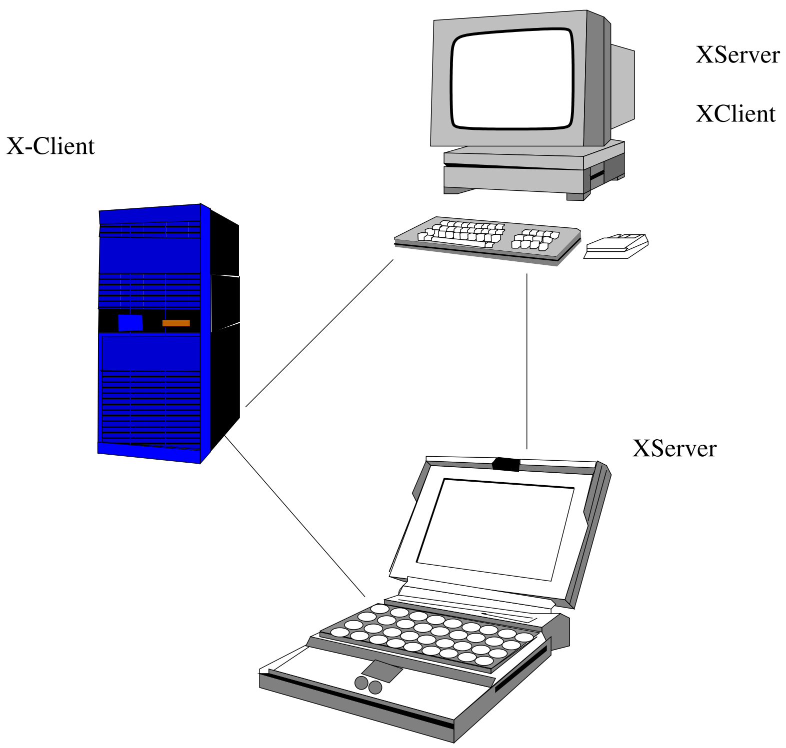
Normalerweise kann man GUI Applikationen nur auf einem Rechner starten, der Graphik darstellen kann.

Ist oft unerwünscht: Login übers Netz; Server die keinen Bildschirm haben.

Unter X Window können Programme auf einem Rechner laufen und ihre Ausgabe an einen Rechner senden, der Graphik-fähig ist.

Dabei läuft auf dem Graphik-Rechner ein sog. *X-Server*, der von sog. *Clients* Anweisungen zur Darstellung bekommt.

Die Clients können dabei auch auf anderen Rechnern laufen.



Die Kommunikation zwischen XClient und XServer wird durch das X-Protocol abgewickelt.

Dieses beschreibt abstrakte Mal- und Darstellungsbefehle.

Das Protokoll ist nicht für low-speed Links ausgelegt (wie etwa Modems).

⇒ Man sollte nicht von daheim versuchen, sich sein Netscape-Fenster von der Uni aus anzeigen zu lassen :-(

Wenn man einen XClient startet, kann man ihm sagen, wo die Ausgabe dargestellt werden soll!

Dazu muss man die Adresse des Ausgaberechners und den Bildschirm angeben:

```
kap01.studcs.uni-sb.de:0.0
134.96.220.1:0.0
:0.0
unix:0.0
```

Die Bildschirmangabe enthält:

- Die Nummer des Servers auf dem Rechner (meist 0)
- Den Bildschirm an diesem Server ([fast] immer 0)

Um global den Weg zum Server zu setzen, benutzt man die Umgebungsvariable `DISPLAY`:

```
setenv DISPLAY kap01.studcs.uni-sb.de:0.0
```

Man kann aber auch jedem Client auf der Kommandozeile das Display angeben, z.B.

```
xterm -display kap01.studcs.uni-sb.de:0
```

Aber: Der Server muss einem *erlauben*, Ausgaben zu tätigen!

Die Authorisierung unter X Window ist einfach, aber unsicher.

In dem File `~/.Xauthority` wird Information gespeichert, die dem Server geschickt wird, um die Verbindung zu authorisieren.

Ist bei uns relativ nutzlos.

Am einfachsten lässt sich die Authorisierung mit `xhost` einstellen.

`xhost +` erlaubt allen Clients Zugriff auf den Server, auf dem `xhost` ausgeführt wurde. Dies ist eine *sehr schlechte* Vorgehensweise.

`xhost -` verweigert allen (nicht-lokalen) Clients den Zugriff.

`xhost` *Rechner* erlaubt *Rechner* den Zugriff.

`xhost -`*Rechner* verweigert *Rechner* den Zugriff.

`xhost` zeigt die aktuellen Einstellungen.

Bei uns sollte nur der Mensch, der sich via `xdm` lokal an der Maschine eingelogt hat, `xhost` Einstellungen ändern können!

Wenn man an einer Maschine nur lokal arbeitet, dann logt man sich via `xdm` ein.

Damit wird der XServer auf der lokalen Maschine gestartet und die Initialisierungsfiles des Benutzers werden ausgeführt.

In diesen Files kann man angeben

- Welche Programme (Clients) beim Start des Servers gestartet werden.
- Die Tastaturbelegung ändern.
- Globale Vorgaben (Farbe, Grösse, ...) für Clients setzen

Was passiert beim Start von X?

Wenn der XServer startet, (lokaler Login via `xdm`), wird das File `~/.xsession` in einer Shell ausgeführt. Sobald die Ausführung dieses Files beendet ist, *beendet sich der Server wieder*.

Daher sollte dieses File sich dieses File erst dann beenden, wenn man sich auslogt!

In diesem File kann man nun verschiedene Clients starten (xterm, Uhren, Netscape, etc. pp)

Als wichtigstes wird in diesem File der *Windowmanager* gestartet (s.u.)

Ein Beispiel `.xsession` File:

```
#!/bin/sh
xset b off
xphoon
xclock -geometry 80x80-0+0 &
xmeter -geometry 250x80-90+-21 -update 5 -cpu ‘hostname‘&
xconsole -geometry 400x60+0+0&
xterm -geometry 80x45+0+90 &
xmail -iconGeometry +415+10 -iconic&
twm
```


Man kann die Tastaturbelegung unter X selbst ändern.

Dazu existiert das Programm `xmodmap`.

Damit kann man Einstellungen für die Maus und Tastatur ändern.

Etwa

```
xmodmap -e "keysym BackSpace = Delete"
```

Man kann sich aber auch den Rechner damit unbenutzbar machen:-)

Für weitere Details: `man xmodmap`

Unter X gibt es das Konzept der *Resourcen*.

Diese geben Werte vor, die die Clients beim Start lesen *können*.

Damit kann man etwa die Grösse des Fensters oder die Hintergrundfarbe voreinstellen.

Resourcen werden über das Program `xrdb` verwaltet.

Schreibt man die Resourcen etwa in ein File `.Xdefaults`, dann nimmt

```
xrdb -merge $HOME/.Xdefaults
```

diese in die Datenbank auf (in `.xsession` einfügen).

Allerdings zwingt niemand einen XClient, diese Datenbank zu benutzen:-(

Resourcen können für eine bestimmte Applikation, oder alle Applikationen oder Applikationen einer bestimmten Klasse geändert werden.

So setzt etwa eine Zeile der Form (in `.Xdefaults`)

```
*Background: grey
```

die Hintergrundfarbe aller Clients auf Grau.

```
XDvi*background:                white
```

setzt dagegen nur die Hintergrundfarbe für die `xdvi` Programme auf weiss.

Resourcen können auch hierarchisch sein, um etwa nur Attribute für bestimmte Komponenten in bestimmten Teilfenstern einer Applikation zu setzen:

```
BibView*mainwin.commandBox.width:       500
```

Verbindungen zwischen einzelnen Komponenten durch `.` oder `*`

Die Resourcen unterscheiden zwischen Klassen– und Instanzen-Namen.

Die Klasse ist für jeden Client fix und legt die Applikation fest (`BibView`)

Der Instanzenname kann auf der Kommandozeile angegeben werden (`-name foobar`) und spezifiziert eine Instanz der Klasse.

So kann man etwa zwei Programme `bibview` und `bibview-black` haben und in `Xdefaults`

```
BibView*background: white
bibview-black*background: black
```

Auch für Komponenten gilt diese Unterscheidung!

Nebenbei bemerkt kann man Resourcen auch auf der Kommandozeile angeben:

```
xterm -xrm "xterm*background: black"
```

Siehe auch `man X`

Optionen

Es gibt einen Satz von Optionen, den jeder Client interpretieren *sollte*.

- `-display` *display*
- `-geometry` wxh+/-x+/-y
- `-bg` *color*, oder `-background` *color*
- `-fg` *color*, oder `-foreground` *color*
- `-fn` *font*, oder `-font` *font*
- `-iconic`
- `-name` *name*
- `-title` *title*
- `-xrm` *resource*

Geometrie: Setzt die Grösse und Position eines Fensters.

Angabe der Form

$$[w\texttt{x}h][(+|\text{-})x(+|\text{-})y]$$

Beispiele:

- `100x200+0+0` Linker Rand an X-Position 0, oberer Rand an Y-Position 0, Grösse 100×200
- `100x200` Nur Grösse, Position wird vom Windowmanager bestimmt.
- `100x200-0+0` Linker Rand des Fensters an linker Schirmseite.
- `--10-10` Linker Rand 10 Pixel über linke Schirmseite. Oberer Rand 10 Pixel von Bildschirmoberseite.

Windowmanager

Anders als andere GUIs, zwingt X Window niemanden zu einem bestimmten Stil des Fensterausehens oder der Funktionalität des Desktops.

Daher bietet X nur ein sog. *Root Window* (Hintergrund) und versieht kein Fenster mit Rändern oder Ziehbalken, etc.

Daneben bietet es aber die Möglichkeit, einen sog. *Windowmanager* zu starten, der alle Fenster auf dem Desktop verwalten darf.

Der Windowmanager darf Rahmen um neue Fenster ziehen, darf deren Grösse ändern, etc.

Er ist auch für die Iconifizierung verantwortlich.

Man sollte den Aufruf des Windowmanagers als letztes in `.xsession` haben (im Vordergrund).

Verfügbare Manager:

- `twm` Tab Window Manager. Einfaches Aussehen.
- `fvwm` Etwas komfortabler. Man kann Buttons, etc auf den Desktop legen.
- (KDE)
- (Gnome)
- (mwm)

In der nächsten Vorlesung:

- Konfiguration von `twm` und `fvwm`
- Einige nützliche Utilities
- Toolkits