

Die xterm-Terminal emulation

Einführung

In X-Window-Umgebungen sind trotz der Möglichkeiten von Menüs und File-Managern doch hin und wieder alphanumerische Eingaben angebracht. Oft benötigt man für die Ausführung einer Aktion über einen C-Shell-Alias weniger Zeit, als man durch Suchen in diversen PopUp-Menüs oder Icon-Gräbern verbraucht. So kommt man als Nutzer nicht umhin, sich mit einer der vielen Terminal emulatio nen, die die Workstation-Hersteller ihren Systemen mitgeben, zu beschäftigen (Terminal emulation ist das Window, in dem ein Shell-Prompt Nutzereingaben abfordert). Jedes System bietet nach dem X-Window-Start die hauseigene Terminal emulation (HP: hpterm, SUN: cmdtool, SGI: winterm, IBM: aixterm...). Sie besitzen in ihren Besonderheiten auch unbestreitbare Vorteile. Aber wehe dem Nutzer, der gezwungen ist, öfter als einmal die Workstation zu wechseln oder in seiner X-Umgebung unterschiedliche Client-Systeme in einer Oberfläche zu integrieren. Nicht nur, daß alles anders aussieht, sondern diese Speziallösungen sind auch weitestgehend unverträglich untereinander. Einen Ausweg aus diesem Dilemma bietet die xterm-Terminal emulation. Sie ist Bestandteil der X11-Distribution des X-Konsortiums und ist auf jedem System vorhanden. Sie ist nicht besser oder schlechter als ihre kommerziellen Konkurrenten, bietet aber den unschlagbaren Vorteil, daß sie überall gleich aussieht und auch von den unterschiedlichsten Herstellern interoperabel ist. Einige Eigenschaften, die in diesem Beitrag diskutiert werden sollen, sind:

- Ein-/Ausgabe-Verhalten wie ein VT102-Terminal oder ein grafikorientiertes Tek4014-Terminal;
- PopUp Menüs für viele xterm-Parameter;
- einen Schieberegler (Scrollbar), um herausgerollten Text wieder sichtbar zu machen, einen Ausschneide- und Einfüge-Mechanismus (Cut & Paste), um Text zwischen verschiedenen xterms (und anderen X-Applikationen) zu kopieren und eine Log-Datei, um xterm-Sitzungen in Dateien mitprotokollieren zu können;
- die Wechselwirkung mit Dienstprogrammen zur Manipulation von Fonts und Ressourcen;
- die Verwendung von xterm-Funktionen und VT-Escape-Sequenzen zur Definition eigener Tastatur- und Maus-Aktionen.

Um es gleich vorwegzunehmen, auch dieser Artikel kann die Möglichkeiten des betrachteten Objekts nicht annähernd erschöpfend behandeln. Er soll vielmehr dem Leser das Rüstzeug mitgeben, sich selbständig mit der xterm-Terminal emulation auseinanderzusetzen.

Interaktives Verhalten

Kommandozeilenoptionen

Die erste Begegnung mit dem xterm findet in der Regel auf der Kommandoebene einer herstellerspezifischen Terminal emulation statt. Der wichtigste Kommandozeilenparameter für xterm ist '-help', der alle anderen Optionen mit einer kurzen Erklärung ausgibt. Die meisten von ihnen verwaltet man am sichersten über Ressourcendefinitionen. Doch eine Handvoll sollte man parat haben, um auch auf einem fremden System das xterm ein wenig nach seinem Geschmack zu gestalten.

-display [*host*]:*server*[.*screen*]

Sollte keine Environmentvariable DISPLAY gesetzt sein (z.B. bei einer telnet-Sitzung oder nach einem rsh-Ruf), dann muß man das xterm auf seinen X-Server (der grafikfähige X-Bildschirm, vor dem man sitzt) adressieren. *host* gibt den Rechnernamen oder die IP-Nummer des Systems an, auf dessen X-Server die Sitzung läuft. Kommt das xterm vom lokalen Rechner, kann *host* auch weggelassen werden. *server* und *screen* werden von 0 (Null) beginnend durchgezählt. Ein Monitor, eine Tastatur und eine Maus erzeugen in der Regel ':0.0'. Hat man mehrere Monitore und zu jedem von ihnen eine eigene Tastatur und Maus, dann werden die *server* hochgezählt, existiert nur eine Tastatur und Maus für alle Bildschirme, dann erhöht sich der *screen*-Parameter für jeden Bildschirm.

-geometry *geometry*

Diese Option adressiert das xterm auf dem Bildschirm in der Form 'Breite x Höhe ± X-Offset ± Y-Offset', wobei Breite und Höhe Spalten und Zeilen des xterm kennzeichnen und die Offsets den Abstand in Pixeln vom Rand darstellen.

-name *name*

Damit erhält das xterm für die Ressourcenverwaltung einen neuen Namen (standardmäßig gilt 'xterm'). Dieser Name taucht im Window-Manager-Titelbalken und am Icon auf, wenn dafür nichts anderes vereinbart ist.

-title *title* und **-n** *iconname*

Für das xterm werden *title* im Window-Manager-Titelbalken und *iconname* am Icon definiert.

-fn *font*

Das xterm startet mit dem deklarierten Font.

-sb und -sl no

'sb' setzt auf die linke Seite des xterm einen Verschieberegler (Scrollbar), um 40 Zeilen des herausgerutschten Textes wieder zurückholen zu können. Sollten 40 Zeilen zu wenig sein, so kann mit 'sl' die Anzahl auf *no* erhöht werden.

-e *command* [*arguments*]

Im xterm wird gleich *command* zur Ausführung gebracht. Dieser Parameter **muß** als letzter in der Optionenketten auftauchen, da alles nach dem '-e' als zum Kommando gehörig betrachtet wird.

Beispiel:

```

mars% xhost +venus
mars% rsh -n venus xterm -display mars:0 -name
venus -sb -sl 200 -e /usr/local/bin/tcsh &

```

'xhost +venus' entspermt den eigenen X-Server (mars:0) für einen Zugriff von venus. Der anschließende rsh-Ruf startet auf dem Rechner 'venus' ein xterm für das Ziel 'mars:0' mit dem Namen 'venus' (erscheint im Titel und als Icon Name). Außerdem wird ein Scrollbar vereinbart, der sich 200 Zeilen merken soll und für die Sitzung eine 'tcsh' aktiviert.

Die VT102-Emulation

Nach dem Start eines xterm-Windows befindet man sich in einer VT102-Emulation. Zwei Eigenschaften gehen über die Eigenschaften eines VT-Terminals hinaus. Die eine ist der schon erwähnte Verschieberegler (Scrollbar). Hat man ihn sich erst einmal verschafft (Kommandozeile, andere Möglichkeiten kommen weiter unten), dann ist man in der Lage, sich alten Text, der nach oben herausgerutscht ist, wieder zu beschaffen. Bewegt man den Mauscursor in den Scrollbar, dann verwandelt er sich in einen vertikalen Doppelpfeil. Folgende Aktionen sind jetzt möglich:

- Maustaste 3 tippen: alter Text rutscht stückweise zurück.
- Maustaste 1 tippen: Text rutscht wieder in die Gegenwart.
- Maustaste 2 gedrückt ziehen: Text läßt sich stufenlos hin und her bewegen.

Die zweite angenehme Eigenschaft ist der Ausschneide- und Kopier-Mechanismus (Cut & Paste), der es ermöglicht, Text von einem xterm zu einem anderen xterm oder anderen X-Applikation zu transferieren. Mit dem Clipboard (siehe unten) ist man außerdem in der Lage, mehrere Textfragmente zu speichern. So geht man beim Textkopieren vor:

- Die Maustaste 1 gedrückt über den zu markierenden Text ziehen (er wird dabei invertiert). Wenn alles markiert ist, Taste wieder loslassen (Invertierung hebt sich wieder auf). Der Text ist in einen Cut-Buffer geladen worden.

- Den Textcursor an die Stelle bewegen, an die der Text übergeben werden soll.

- Die Maustaste 2 tippen. Der Text wird beginnend von Textkursorposition aus dem Cut-Buffer ausgeschrieben, wenn dort eine Texteingabe erwartet wird.

- Ein Doppelklick mit Maustaste 1 markiert ein Wort, ein dreifacher Klick eine ganze Zeile.

Außerdem ermöglicht xterm, mit beliebiger Spalten- und Zeilenzahl zu arbeiten. Diese können über Aufrufparameter (-geometry), über eine Window-Manager-Aktion (interaktiv) oder über Ressourcen-Parameter (siehe unten) festgelegt werden. Bei den neueren xterm-Implementationen werden gleich die entsprechenden Termcap/Terminfo-Einträge (z.B. LINES, COLUMNS) richtig gesetzt, um einer Applikation zu übermitteln, wie weit sie sich ausdehnen kann. Bei einem älteren xterm muß man mit dem Programm 'resize' nachhelfen. Ein entsprechender cshell-Alias sieht so aus:

```
alias rs 'set noglob; eval `resize`; unset noglob'.
```

Die xterm-Menüs

Ist xterm erst einmal aufgerufen, dann lassen sich eine Reihe von Parametern über die xterm-Menüs ändern. Man erhält sie, indem man den Mauscursor in das xterm fährt und bei gedrückter Control (Strg)-Taste eine Maustaste betätigt. Bei gleichzeitig gedrückter Control- und Maustaste kann dann der Mauscursor zur gewünschten Option bewegt und zu deren Aktivierung über ihr (der Option) die Maustaste losgelassen werden. Eine aktivierte Option ist an einem davor befindlichen Häkchen erkennbar. Es gibt vier verschiedene Menüs: Main Options, VT Options, VT Fonts und Tek Options.

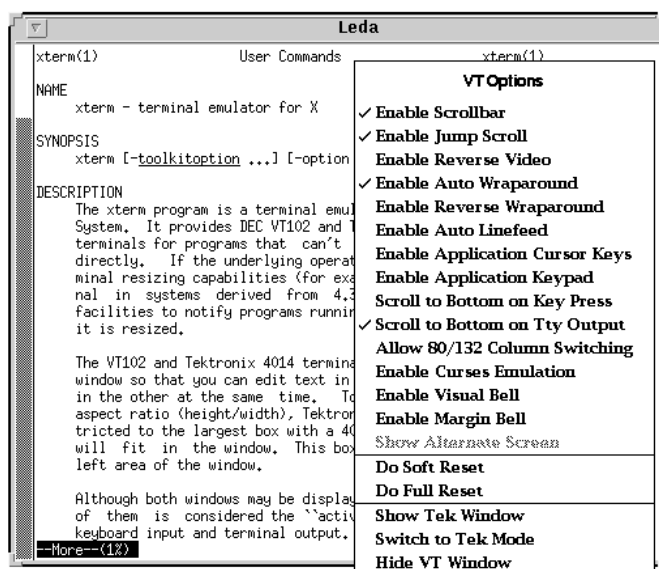


Abb: Das VT Options Menü eines xterm

Main Options ([Ctrl]-[Maustaste 1])

- Secure Keyboard

Wenn es anderen Rechnern erlaubt ist, auf den eigenen X-Server zuzugreifen (wie im obigen Beispiel dem Rechner 'venus'), dann besteht für Fremde die Möglichkeit, durch die Ausnutzung von X-Protokoll-Diensten den eigenen X-Server auszuhorchen (die Tastatureingaben mitzuschneiden, z.B. ein Paßwort). Um dies zu verhindern, kann man 'Secure Keyboard' zuschalten. Danach wird die Tastatureingabe ausschließlich und direkt an das xterm (d.h. dem Programm in ihm, z.B. login) gesendet. In dieser Zeit ist der xterm-Inhalt invertiert. Nur ein xterm kann zur gleichen Zeit im Secure-Modus sein. Sinnvoll ist diese Option, wenn man sich lokal als ROOT einloggt. Bei Logins über das Netzwerk geht zumindest das Paßwort im Klartext über die Leitung und mit der gewonnenen X-Server-Sicherheit ist es dann wieder vorbei.

- Log to File

Diese Option ermöglicht das Mitschneiden der xterm-Ein/Ausgabe in einer Datei. Nachdem Zuschalten wird diese im Home-Directory des Nutzers mit Namen XtermLog.xxxxx abgelegt, wobei xxxxx mit der Prozeß-ID des xterm korrespondiert. Wenn diese Option aktiv ist, sollte man auf keinen Fall ein

% cat XtermLog*

innerhalb des xterm ablaufen lassen, da xterm auch dies mitprotokolliert und XtermLog dann schnell bis zum Anschlag an eine Grenze (Quota oder Filesystem) wächst. Aufgrund eines Sicherheits-Loches bei einer früheren Implementierung steht diese Option nicht auf allen Systemen (z.B. SGI-IRIX 5.3) zur Verfügung.

- Redraw Window

Sollte der Inhalt des xterm einmal zerstört (unkenntlich) sein, so kann über diesen Schalter eine Erneuerung des Inhaltes erzwungen werden.

- Send [HUP | TERM | KILL] Signal, Quit

Alle diese Signale bewirken auf unterschiedliche Art eine Beendigung der xterm-Sitzung.

VT Options Menu ([Ctrl]-[Maustaste 2])

Dieses Menü ist nur im VT102-Window sichtbar.

- Enable Scrollbar

Schaltet den Scrollbar auf der linken Seite an (Standard: aus).

- Enable Auto Wraparound

Wird bei der Texteingabe das Ende der Zeile erreicht, dann wird bei aktiver Option automatisch auf der folgenden Zeile fortgesetzt (Standard: ein).

- Enable Reverse Wraparound

Bei besonders langen Kommandozeilen passiert es schon einmal, daß die Zeile nicht ausreicht. Der Text wird dann automatisch auf der nächsten Zeile weitergeführt. Stellt man jedoch fest, daß man kurz vor Ende der vorhergehenden Zeile einen Fehler gemacht hat, dann kommt man nicht zurück - zumindest solange nicht, bis diese Option eingeschaltet wurde (kann auch nachträglich geschehen, Standard: aus).

- Scroll to Bottom on [Key Press | Tty Output]

Diese Optionen legen fest, wie sich das xterm verhält, wenn mit dem Scrollbar alter Text zurückgeholt wurde. Bei 'Scroll to Bottom on Key Press' wird zur aktuellen Zeile zurückgesprungen, wenn eine Taste betätigt wurde (Standard: aus). Bei 'Scroll to Bottom on Tty Output' wird dieser Effekt wirksam, wenn Zeichen (z.B. Console-Ausschriften) auf dem xterm erscheinen (Standard: ein).

- Enable Visual Bell

Sollte aus irgendeinem Grund der X-Server nicht des Piepsens mächtig sein (wichtig beim 'vi'), dann kann hier ein sichtbarer Glockenton zugeschaltet werden. Bei jedem Pieps wird dann einmal kurzzeitig der Inhalt des xterm invertiert.

- Do [Soft | Full] Reset

Setzt das xterm teilweise oder vollständig (einschließlich löschen des Bildschirm- und manchmal auch des Scrollbuffer- Inhaltes) zurück.

- Show Tek Window

Bringt das Tektronix-Window zum Vorschein. Die Shell und damit auch die Tastatureingabe liegen aber noch beim VT102-Window.

- Switch to Tek Mode

Die Tastatureingabe wird auf das Tek-Window umgeschaltet - das VT-Window bleibt aber noch erhalten.

- Hide VT Window

Das VT-Window wird abgeschaltet (es verschwindet vom Bildschirm).

VT Fonts Menu ([Ctrl]-[Maustaste 3])

Dieses Menü ist nur im VT102-Window sichtbar.

- Default, Tiny, Small, Medium, Large

Man kann zwischen verschiedenen voreingestellten Fonts wählen.

- Escape Sequence

Dieses Feld ist blind, solange es nicht durch eine Escape-Sequence mit einem Font programmiert wurde (siehe weiter unten).

- Selection

Hier kann interaktiv der Font gewechselt werden - z.B. mit einer Mauscursor-Selektion oder mit 'xfontsel' (siehe unten).

Tek Options Menu ([Ctrl]-[Maustaste 2])

Dieses Menu ist nur im Tektronix4014-Window sichtbar.

- [Large | #2 Size | #3 Size | Small] Characters
Hier können wieder verschiedene Fonts gewählt werden.
- Page
Das ist die wichtigste Option des Tek-Menüs. Es löscht das Tek-Window (siehe nächster Abschnitt).
- Copy
Hier wird eine Log-Datei namens COPYyy-mm-dd.hh.mm.ss erzeugt, wobei yy-mm-dd.hh.mm.ss das aktuelle Datum und die genaue Uhrzeit des Log-Beginns darstellen.
- Show VT Window
Diese Option erzeugt das VT-Window wieder, falls es nicht mehr existiert.
- Switch to VT Window
Die Tastatureingabe wird an das VT-Window abgegeben.
- Hide Tek Window
Das Tek-Window wird vom Bildschirm genommen (der Inhalt geht nicht verloren - nach 'Show Tek Window' wird er wieder sichtbar).

Die Tek4014-Emulation

Neben der VT102-Emulation bietet das xterm noch eine Tektronix 4014-Emulation. Diese war zu den Zeiten, als Window-Oberflächen noch unbekannt waren, eine Möglichkeit, Grafik auf den Bildschirm zu bringen. Es gibt noch heute Anwendungen, die das

Bedürfnis verspüren, sich auf Tek-Ebene auszudrücken. So ist in System-V-UNIX-Systemen ein Programm verbreitet, daß zu vorgegebenen Zeiten Systemdaten einsammelt und in Dateien abspeichert. Mit 'sar' (System Activity Reporter) kann man sich diese Zahlengräber auflisten lassen, aber mit 'sag' (System Activity Graph) erhält man sie auch als grafische Darstellung - aber nur auf Tektronix 4014. In der Abbildung ist die CPU-Auslastung eines File Servers sowohl als Zahlenkolonne auf VT-Window als auch in grafischer Form auf dem zugehörigen Tek-Window dargestellt. Unbestritten besitzt die grafische Darstellung einen höheren anschaulichen Wert.

Zum Tek-Window gelangt man über das 'VT Options Menu' im VT-Window mit der Funktion 'Show Tek Window'. Dann erscheint ein zweites, größeres und leeres Window. Die Tastatureingabe liegt noch beim VT-Window. Wenn man im Tek-Window arbeiten möchte, dann muß man noch 'Switch to Tek Mode' aktivieren. Die Shell arbeitet dann im Tek-Window (es ist die gleiche wie im VT-Window, d.h. man hat alle dort definierten Environment-Variablen und die History). Die Arbeit im Tek-Window bietet einige Überraschungen, z.B. ist das Window zweispaltig organisiert. Hat man mit Kommandos und Bildschirmausgaben den Boden des Windows erreicht, dann wird oben in der Mitte fortgefahren. Das nächste, was auffällt, ist die Tatsache, daß nichts gelöscht wird. Hat man genügend viele Ausgaben produziert, dann ist nichts mehr zu erkennen, da alles übereinander geschrieben wurde (ist die zweite Spalte voll, dann wird mit der Ausgabe in der ersten weitergemacht - ohne sie zu löschen). Dann hilft nur noch, alles zu beseitigen - mit der Tek Menu

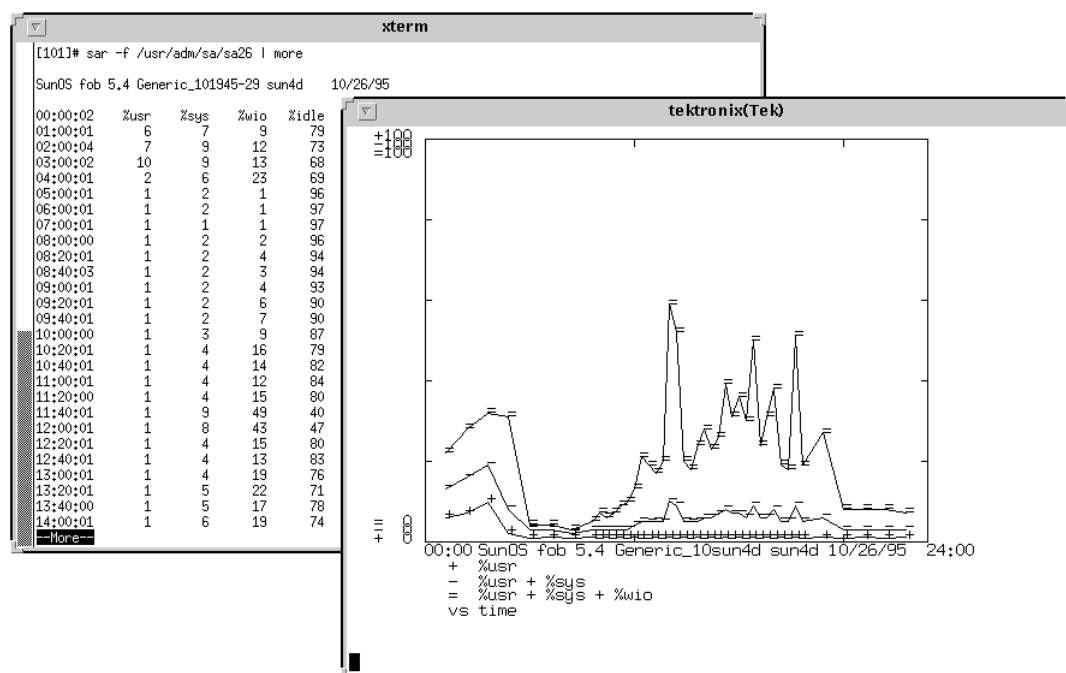


Abb.: VT- und Tek- Window mit System Activity Graph eines File Servers

Option: PAGE. Für die tägliche Arbeit ist dieses Window etwas zu anstrengend, aber nur mit ihm kann man die obige Abbildung realisieren (unset prompt; sag -T4014 -f). Zurück zum VT-Window kommt man via 'Switch to VT Mode' und 'Hide Tek Window'.

Wechselwirkung mit anderen Programmen

Font-Auswahl

Ist ein xterm-Window erst einmal gestartet, dann ist die Wahrscheinlichkeit sehr groß, daß der zur Erscheinung kommende Font nicht den Erwartungen des vor ihm sitzenden Nutzers entspricht. Eine Möglichkeit, dies zu ändern, besteht darin, sich des [VT Font Menu]-[Selection]-Mechanismus zu bedienen. Dabei wird der Inhalt des Cut-Buffer als Font interpretiert und das aktuelle xterm auf diesen Font umgestellt. Zwei Wege sind dabei beschreibbar.

- 1) Man listet sich mit dem Programm 'xlsfonts' die für den X-Server verfügbaren Fonts auf. Es sind sehr viele, deshalb sollte man eine Vorauswahl treffen - z.B. 'xlsfonts | grep courier | more', um sich nur Fonts mit fester Zeichenbreite anzeigen zu lassen. Dann markiert man sich einen Font (Maustaste 1 ziehen und am Ende loslassen) und schaut sich die Wirkung mit [VT Font Menu]-[Selection] an. Dieses Verfahren ist zeitaufwendig, da zu viele Schritte nötig sind, bevor man beurteilen kann, ob es der richtige Font ist.

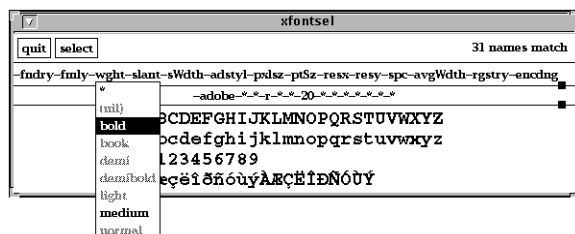


Abb.: xfontsel

- 2) Mit 'xfontsel' erhält man dagegen sofort einen Eindruck vom gewählten Font. Dieses Programm arbeitet mit Pop-Up-Menüs und einem Beispielzeichensatz, der den gerade erwählten Font darstellt. Zusätzlich gibt es einen [Select]-Button, mit dem der aktuelle Font in den Cut-Buffer kopiert wird. Von dort kann man ihn per [Maustaste 2]- Klick in einen Editor absetzen (wichtig für Ressourcen-Definitions-Datei um den Font resident festzuschreiben) oder ihn via [VT Font Menu]-[Selection] für das aktuelle xterm gültig machen.

Clipboard

Mit dem Programm 'xclipboard' steht eine Möglichkeit zur Verfügung, markierten Text aufzuheben. Nach dem Start steht ein leeres Blatt zur Verfügung, in das mit Cut & Paste von Hand Text vom xterm ins xclipboard transportiert werden kann. Mit einem [new]-Button können

neue leere Seiten erzeugt werden, mit [next] und [previous] können die Textfragmente durchgeblättert werden. Wie der Text automatisch (schon beim Markieren) ins xclipboard gelangt und auch von dort ohne erneutes Kopieren in ein xterm zurückgelangt, wird bei den xterm-Funktionen gezeigt.

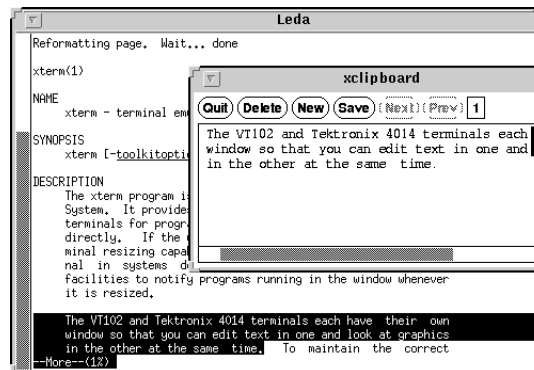


Abb.: xclipboard mit kopiertem Text

Ressourcen-Editor

Der Ressourcen-Editor 'editres' ist ein relativ junges Produkt aus der X11-Distribution. Die Idee ist, ein grafisches Programm zu starten, mit dessen Hilfe man sowohl den Ressourcen-Baum einer X-Applikation inspizieren als auch den Ressourcen-Attributen interaktiv neue Werte zuweisen kann¹. Dies setzt ein gewisses Kommunikationsvermögen zwischen 'editres' und der X-Applikation voraus. Die Kommunikation zwischen editres und xterm ist durchaus noch holprig zu nennen. Nicht jeder Ressourcen-Zweig wird angezeigt und oft gibt xterm keine Auskunft über Voreinstellungen (z.B. muß man erst alle xterm-Menüs einmal aktivieren, um sie im Ressourcen-Baum auch tatsächlich wiederzufinden). Man sollte trotzdem dieses Produkt im Auge behalten.

Nach dem Start von editres erhält man über [Commands]-[Get Widget Tree] den Mauscursor als Faden-Kreuz. Dieses klickt man über ein xterm ab und erhält den Ressourcen-Baum in editres. Das Kästchen links oben dient zum Bewegen des Ausschnittfensters (mit [Maustaste 1] gedrückt ziehen). Dann wählt man einen Instanzenwert aus und erzeugt sich mit [Commands]-[Show Resource Box] die Ressourcen-Box, in der man nun versuchen kann, gestaltend zu wirken. Das obige Ergebnis des veränderten 'Main Menu' ist durch editres erzeugt worden und wird im einzelnen nachfolgend beschrieben.

Ressourcen usw.

Ressourcen

Die Datei /usr/lib/X11/app-defaults/XTerm (SUN: /usr/openwin/lib/app-defaults/XTerm) stellt neben

¹ Zum Verständnis von Ressourcen siehe 'Ressourcenverwaltung unter X-Window' in RZ-Mitteilungen 4'92

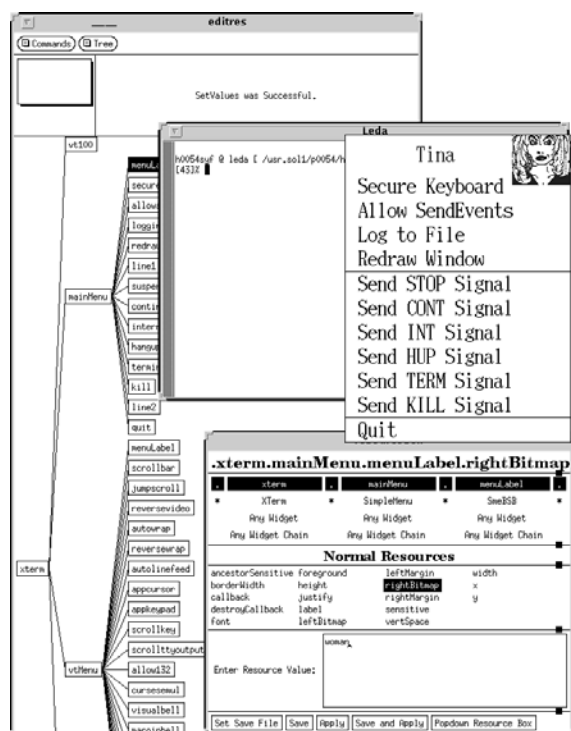


Abb: editres mit Ressourcen-Box (rechts unten) und xterm (Leda)
mit aufgeklapptem, selbst gestaltetem Main Menu (Tina)

editres den reichhaltigsten Fundus für die Beschreibung der Ressourcen-Werte dar. Sie kann in das HOME-Directory kopiert und dann nach Herzenslust editiert werden. Eine ganz gute Übersicht (aber nicht vollständig) gibt auch das Manual von xterm. Hier sei nur grob skizziert, was man erwarten darf.

Die Attribute der ersten Ebene legen die Erscheinung von `xterm` insgesamt fest, wie Titel und Icon-Name. Als weiterführende Einschränkungen treten

VT100, Tek4014 und SimpleMenu (xterm-Menüs) auf. Im VT-Window bestimmen die Attribute der zweiten Ebene solche Dinge wie Geometry (Spalten x Zeilen) oder ob ein ScrollBar (true/false) sichtbar ist. Die einzige hier verbleibende Konkretisierung ist Scrollbar, in dem die Attribute Thickness (in Pixeln) oder Foreground (Farbe) auftreten. Bei den SimpleMenu können das Label (Überschrift) oder die Geometry (Pixel x Pixel) gleich festgelegt werden. Hier lauert eine Falle - die Geometry. Wer lässig folgendes festlegt:

```
XTerm*geometry: 100x32
```

in der Absicht, das VT-Window in 100 Spalten x 32 Zeilen erscheinen zu lassen, der beschneidet gleichzeitig die xterm-Menüs auf 100x32 Pixel (damit sind alle Menü-Punkte verschwunden). Also schön ausschreiben:

XTerm.vt100.geometry: 100x32

Als Einschränkungen in den SimpleMenu treten die Menü-Optionen (SmeBSB) auf, wie z.B. menuLabel (Überschrift), securekbd (Secure Keyboard) usw. (siehe Manual xterm). In diesen können zusätzlich kleine Bitmaps (links oder rechts) aufgenommen werden, Fonts und Farben festgelegt werden usw.

Das Beispiel aus dem editres-Abschnitt kann wie folgt beschrieben werden:

! Überschrift vom Main Menu

```
XTerm.mainMenu.label: Tina
```

! Farbe der Überschrift des Main Menu

```
XTerm.mainMenu.menuLabel.foreground: blue
```

! Bitmap nach rechts

XTerm.mainMenu.menuLabel.rightBitmap: woman

! für Bitmap im gesamten Menü Rand auf 70 Pixel

XTerm.mainMenu.SmeBSB.rightMargin: 70

! für alle Menüs Font vergrößern

```
XTerm.SimpleMenu.SmeBSB.Font: 12x24
```

XTerm (xterm)	Title (title)	Scrollbar (scrollbar)	Thickness (thickness)
	IconName (iconName)	ScrollBar (scrollBar)	Background (background)
		SaveLines (saveLines)	Foreground (foreground)
	VT100 (vt100)	Font (font)	
		Geometry (geometry)	
		Foreground (foreground, cursorColor, pointerColor)	LeftBitmap (leftBitmap)
	Tek4014 (tek4014)		RightBitmap (rightBitmap)
		SmeBSB (menuLabel ...)	Foreground (foreground)
	SimpleMenu (mainMenu, vtMenu, tekMenu, fontMenu)	Label (label)	Font (font)
		Geometry (geometry)	HorizontalMargins (rightMargin, leftMargin)

Abb.: Auszug aus dem Ressourcen-Baum von xterm

xterm-Funktionen

In xterm-Windows ist eine Neudefinierung von Tasten- und Maus-Aktionen über die 'Translations'-Ressource möglich. Eine Übersicht der möglichen, einzubinden- den Funktionen liefert wieder das xterm Manual. Ein einfaches Beispiel soll die Syntax verdeutlichen:

```
XTerm.VT100.Translations: #override\
<Key>F1: string("\pr -Pvps ") \n\
<Key>F2: string("\lpq -Pvps") string(0x0d) \n\
<Key>F3: redraw()
```

Der Eintrag besteht eigentlich nur aus einer Zeile, durch '\ ' werden Fortsetzungzeilen ermöglicht und mit '\n' wird eine Definition abgeschlossen. F1 gibt auf die Position des Textkursors die Zeichenkette 'lp -Pvps '

als echo) und vom xterm interpretiert. Ein kleiner Aus- zug der Möglichkeiten sei hier angefügt (^G steht für Hex 07 - BEL) ²:

```
^[1;string^G - ändert Icon-Name auf string
^[2;string^G - ändert Window-Titel auf string
^[46;string^G - ändert den Namen der XtermLog-Datei
auf string
^[50;string^G - ändert den Font auf string
```

So erzwingt das folgende Kommando

```
% echo "^[50;8x13bold^G"
```

die sofortige Umstellung des xterm-Fonts auf '8x13bold' und dieser steht dann auch im [VT Fonts Menu]-[Escape Sequence] zur Verfügung. Ein weiteres Beispiel zeigt die Abbildung:

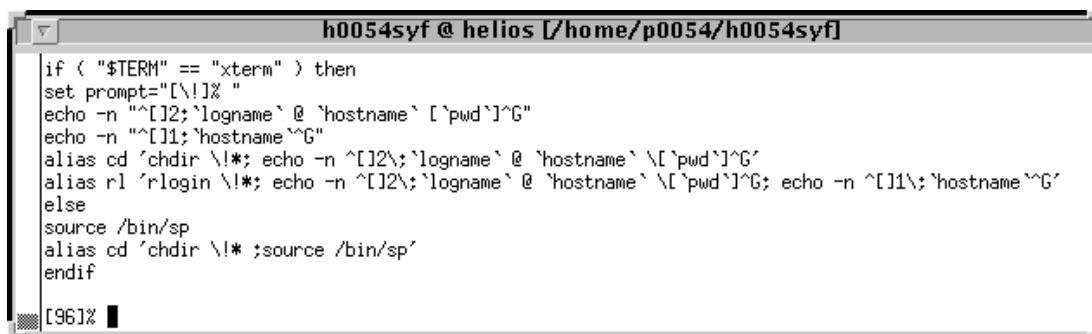


Abb.: Steuerung des Window-Titels über Escape-Sequenzen

aus und ermöglicht noch weitere Eingaben. F2 dagegen schickt das Kommando 'lpq -Pvps' gleich mit abschließendem Carriage Return an die Shell. F3 bewirkt ein Refresh des Window. Die Brücke zum xclipboard läßt sich mit folgender Definition schlagen:

```
XTerm.VT100.Translations: #override\
Button1 <Btn3Down>: select-end(CLIPBOARD) \n\
<Btn2Up>: insert-selection(PRIMARY,CLIPBOARD)
```

Hier wird Maustaste 1 zum modifizieren benutzt. Wenn mit Maustaste 1 Text markiert wird, dann kann man, bevor die Taste losgelassen wird, durch Drücken der Maustaste 3 die Selektion ins xclipboard transportieren. Einfügen kann man wieder mit Maustaste 2. Wenn kein Text im Primary-Buffer liegt, dann wird der Clipboard-Speicher benutzt. Die Ctrl- und Meta(Alt)-Tasten dürfen bei der Einfüge-Operation nicht gedrückt sein (ausgeschlossene Modifizierer), da [Ctrl]-[Maustaste 2] schon das 'VT Options Menu' aktiviert.

Escape-Sequenzen

Terminalemulationen lassen sich im allgemeinen über sogenannte Escape-Sequenzen steuern - so auch das xterm. Wie der Name sagt, fangen Escape-Sequenzen mit einem Escape an (Hex 1B, wird oft als ESC oder ^[dargestellt), denen noch weitere Zeichen folgen. Sie werden an das xterm als Zeichenstrom gesendet (z.B.

Der Window-Titel und der Icon-Name wird hier über Escape-Sequenzen gesteuert. Im Titel sollen Nutzer- und Rechnername und augenblickliches Directory erscheinen. Alles dafür Nötige wird in die ~/.cshrc gepackt. Diese testet zuerst, ob sie sich in einem xterm befindet, dann initialisiert sie Window-Titel und Icon-Name (die beiden echo-Zeilen). Für jedes 'cd' wird eine neue Escape-Sequence mit dem aktuellen Directory an das xterm geschickt (alias für cd). Um nach der Rückkehr von einem Rlogin zum nächsten Rechner den aktualisierten Window-Titel (Icon-Name) zu bekommen, wird 'rlogin' mit dem alias rl gekapselt (wenn auf dem entfernten System der gleiche Eintrag in der ~/.cshrc vorhanden ist, dann initialisiert sich der Titel/Icon-Name ohnehin richtig für diesen Rechner).

Schlußbemerkung

Dieses Thema hat einen anspruchsvollen Kern. Wer keine Zeit (oder auch keine Freude) zum Spielen hat, sollte sich nicht damit unglücklich machen, daß er mit aller Macht versucht, sein xterm in ein bestimmtes Outfit zu pressen. Vor allem in den Ressourcen-Dschungel sollte man sich nur wohlüberlegt hineinwagen. Die lockere Serie zum Thema X-Window wird in einem der nächsten Hefte mit den administrativen Aufgaben zu X-Verwaltung fortgesetzt.

Frank Sittel

² Eine vollständige Übersicht der Escape-Sequenzen findet man in "X Window System User's Guide" von O'Reilly & Associates