

LUG-CAMP 2002 // Flensburg 17~20. Mai 2002

Manchmal kommen sie wieder -- *screen*

von Heike Jurzik (LinuxMagazin 01/2001) // ergänzt von Christian

Haben Sie sich schon oft darüber geärgert, dass Sie zwischen vielen Terminals hin- und herschalten mussten, weil Sie mehrere Anwendungen gleichzeitig im Vordergrund laufen lassen wollten? Oder waren Sie sauer, weil ein Prozess, den Sie auf einem Rechner am Arbeitsplatz starten mussten, nicht rechtzeitig zu Feierabend fertig war, Sie aber die Ausgabe des Programms kontrollieren wollten? *screen* ist ein äußerst mächtiges Tool, das Ihnen viele Aufgaben erleichtern kann.

Mit *screen* können Sie bis zu zehn virtuelle Fenster in einem einzigen Xterm (oder auf der Konsole) simulieren. In all diesen Fenstern können Sie nun Programme laufen lassen -- jedes der virtuellen Fenster ist dabei von den übrigen unabhängig. Tippen Sie einfach mal *screen* -- nach einem kurzen Begrüßungstext gibt es den Hinweis, wie es weitergeht:

```
[Press Space or Return to end.]
```

Mit der Leertaste treten Sie also ein ins Reich der unendlichen Terminal-Weiten. Es stehen Ihnen eine Reihe von Kommandos zur Verfügung, die alle mit [Strg-a] beginnen: Halten Sie dazu die [Strg]- oder auf englischen Tastaturen die [Ctrl]-Taste gedrückt und tippen Sie [a]. Jetzt wartet das Programm auf die weiteren Befehlseingaben: [Strg-a] [?] gibt beispielsweise eine komplette Übersicht der Tastenbelegung (s. Tabelle 1).

Tabelle 1: Aufrufen von *screen* auf der Konsole

Aufruf	Kommando	Bedeutung
screen	start	Startet eine neue screen session. Es können auf einem Rechner mehrere screen session's laufem, unterschieden werden diese durch ihre PID unterschieden
screen -ls	listing	zeigt die laufenden screens der Maschine mit PID und Status
screen -r	resume	wiederherstellen des detached screens, nur möglich wenn bei einem screen. Laufen mehrere screens kommt eine Fehlermeldung.
screen -r PID	resume	wiederherstellen des detached screens mit der PID
screen -r -d	resume	wiederherstellen des attached screens, dieser wird autmoatisch auf der Gegenstelle detached
screen -r -d PID	resume	wiederherstellen des attached screens mit der PID, dieser wird autmoatisch auf der Gegenstelle detached
screen -r -D	resume	wiederherstellen des attached screens, dieser wird autmoatisch auf der Gegenstelle detached und der User ausgelockt
screen -r -D PID	resume	wiederherstellen des attached screens mit der PID, dieser wird autmoatisch auf der Gegenstelle detached und der User ausgelockt
screen -r -x	resume	wiederherstellen des screens, dieser wird gespiegelt , der User auf der Gegenstelle bleibt eingelockt und sieht alles was man selbst macht, ebenso sieht man, was der User gegenüber macht. Eignet sich hervorragend zum konfigurieren.

Tabelle 2: Tastenkombinationen im <i>screen</i>		
Tastaturkürzel	Kommando	Bedeutung
[Strg-a] [?]	help	Listet alle Tastenbelegungen auf.
[Strg-a] [c]	create	Öffnet ein weiteres virtuelles Fenster.
[Strg-a] [Leertaste]	next	Wechselt zum nächsten Fenster, wiederholt man das Kommando, kann man durch alle Fenster "durchlaufen".
[Strg-a] [0...9]	select n	Wechselt zu Fenster mit Nr. n.
[Strg-a] [w]	windows	Zeigt in einer Zeile am unteren Rand für kurze Zeit, wieviel Fenster gestartet sind, das aktuelle wird mit * hervorgehoben.
[Strg-a] [A]	title	Erlaubt die Eingabe eines Titels für dieses Fenster. Diese werden von [Strg-a] [w] genutzt und angezeigt
[Strg-a] [H]	log	Loggt die Standardausgabe in eine Datei, abhängig von der Nummer des Fensters (1-10) heißt das Logfile <i>screenlog.n</i> , erneuter Aufruf von [Strg-a H] beendet das Mitschneiden
[Strg-a] [d]	detach	"Befreit" den <i>screen</i> , alle darin gestarteten Prozesse laufen weiter, aber das Programm verabschiedet sich vom Terminal: Nun kann man sich ausloggen. Mit <i>screen -r</i> kann der <i>screen</i> wieder aufgerufen werden (ausführliche Erklärung folgt im Text).
[Strg-a] [D] [D]	pow_detach	"Power Detach" -- löst nicht nur den <i>screen</i> los, sondern loggt auch direkt aus dem Terminal aus.
[Strg-a] [K]	kill	Zerstört den ganzen <i>screen</i> -- netterweise gibt es hier eine Sicherheitsabfrage: <i>Really kill this window [y/n]</i>
[Strg-a] [Strg-a]	other	Wechselt immer zwischen zwei Fenstern hin und her.

Out of the blue -- screen!

Neben den ganzen Steuerungskommandos innerhalb der Fenster kann man das Programm natürlich auch mit verschiedenen Parametern beim Start versehen. Für den Fall, dass Sie *screen* mehrmals gestartet haben und nicht mehr wissen, wieviele und ob diese gerade aktiv sind, gibt es die Option *-ls* (steht für: *-list*):

```
huhn@asteroid:~$ screen -list
There are screens on:
  1200.pts-10.asteroid  (Attached)
  1203.pts-14.asteroid  (Detached)
```

Hier sehen Sie einmal die Prozess-ID (*pid*), dann das virtuelle Terminal (*tty*), in dem der *screen* gestartet wurde, den Host (*asteroid*) und als letzte Information, ob er gerade aktiv ("attached") ist oder schlafen gelegt wurde ("detached"). Inaktive *screens* können mit *screen -r [pid.tty.host]* wiederbelebt werden. Die Angabe der Prozessnummer und des Terminals sind nur erforderlich, wenn mehrere *screens* inaktiv sind. Sie können sich die Sache wesentlich erleichtern, wenn Sie der Session direkt am Anfang einen Namen geben: *screen -S petronella* tauft Ihren *screen* auf den Namen "petronella". In der Übersicht heißt dieser dann: *1364.petronella* -- der Name ersetzt also Terminal und Host. Übrigens: Hat sich ein *screen*-Prozess einmal aufgehängt, können Sie diesen in der Übersicht am Statusflag "dead" erkennen. Sie werden ihn mit dem Parameter *screen -wipe* elegant los.

Wenn Sie einen schlafengelegten *screen* wiederbeleben, möchten Sie von Zeit zu Zeit zurückscrollen können, um die letzten Ausgaben laufender Programme zu betrachten. Standard für den Puffer sind 100 Zeilen. Dieses kann man mit Hilfe der Option *-h zeilenanzahl* ändern. Bei einem *screen -h 1000* können Sie nun also 1000 Zeilen zurückgehen. Um sich in diesem Puffer zu bewegen, gibt es eine Reihe von Tastaturkommandos. Dazu begeben Sie sich zunächst in den Copy/Scrollback-Modus (s. Tabelle 1, [Strg-a] [Esc]). Wenn Sie den Editor *vi* bereits kennen und nutzen, sind Ihnen die Kommandos zur Cursor-Bewegung sicher vertraut. Andernfalls finden Sie in Tabelle 2 eine Kurzreferenz über die Befehle.

Text kopieren auf der Konsole - mit screen geht's!

Tabelle 3: Die wichtigsten Kommandos zur Navigation im Copy/Scrollback-Modus	
Kommando	Bewegung
[Strg-a] [Esc]	Wechselt in den Kopiermodus: Sollte keine Maus zum Markieren von Text vorhanden sein, kann man jetzt mit den Buchstaben <i>h, j, k, l</i> zur gewünschten Stelle auf dem Bildschirm gehen, eine Markierung mit der Leertaste setzen, dann zur nächsten Stelle gehen, wieder die Leertaste drücken, um in die "Zwischenablage" zu speichern. Mit [Strg-a] "]" (also [Strg-A] gefolgt von einer schließenden eckigen Klammer) fügt man den markierten Text übrigens ein, mit [Esc] bricht man die Aktion ab. [STRG]-[A] [[] - startet den kopiermodus (wie auch [STRG]-[A] [ESC]) mit den Pfeiltasten den Startpunkt aussuchen [Leertaste] - Setzt den Startpunkt mit den Pfeiltasten den Endpunkt aussuchen [Leertaste] - Setzt den Endpunkt [STRG]-[A] [[] - fügt in einem beliebigen Screen den Inhalt ein
<i>h, j, k, l</i>	bewegen den Cursor Zeile für Zeile oder Spalte für Spalte, links, rechts, rauf, runter.
<i>0, \$</i>	zum äußersten linken oder rechten Ende der Zeile.
<i>H, L, M</i>	bewegt den Cursor in der Spalte links außen nach ganz oben, nach ganz unten oder in die Mitte.
<i>+, -</i>	Zeile hoch oder runter.
<i>G</i>	springt ans Ende des Puffers.
<i>g</i>	springt an den Anfang des Puffers.
<i>w, b, e</i>	wortweise springen: zurück, vor und ans Wortende.

Tabelle 4: Gimmicks	
Kommando	Bewegung
	splitten
[Strg-a] [S]	Splittet den Screen horizontal in zwei gleich grosse Fenster Hier gelten die normalen screen Befehle weiter <i>Wichtig: unbedingt GROSS 'S'</i>
[Strg-a] [TAB]	wechselt zwischen den screens
[Strg-a] [c]	öffnet ein neues screen fenster
[Strg-a] [0-9]	zum aufrufen der vorhanden screens in das "gesplittete" Fenster
[Strg-a] [Q]	Hebt die Splittung wieder auf, das aktive Fenster wird wieder komplett angezeigt, die screens laufen natürlich weiter
	locken
[Strg-a] [x]	Sperrt den <i>screen</i> -- nach einer gültigen Passworteingabe können Sie weiterarbeiten

Mit *screen*, Charme und Melone

Sie können in Ihrem Home-Verzeichnis eine Datei *.screenrc* anlegen, in die Sie bestimmte Wünsche zum Programmverhalten eintragen. Tragen Sie zum Beispiel

```
startup_message off
```

ein, dann entfällt die Begrüßung beim Programmstart. Praktisch ist auch die Möglichkeit, sich eigene Kommandos zu definieren. Schreibt man in die *.screenrc* z. B.

```
bindkey ^f screen ssh marvin.cologne.de
```

(und nicht wie in der Man-Page beschrieben *bind xy!*), wird, wenn im *screen*-Fenster [Strg-f] gedrückt wird, automatisch ein neuer Screen mit einer *ssh*-Verbindung zum Rechner *marvin.cologne.de* aufgemacht. Auf diese Weise können Sie eine Menge nützlicher Aliase definieren. Möchten Sie als Standard-Puffer mehr als die 100 Zeilen haben, können Sie durch einen Eintrag

```
defscrollback 1000
```

eine eigene Puffer-Größe festlegen. Ein nettes Feature ist auch die sogenannte *vbell_msg*. Dazu muss als erstes definiert werden: *vbell on*, danach die gewünschte Meldung, die erscheinen soll, wenn ein Fenster einen "beep" ([Strg-g]) empfängt, z. B.

```
vbell_msg "Hallo! Hier piept's!"
```

Es gibt noch eine ganze Reihe von Tipps und Tricks zu diesem Thema in der sehr ausführlichen Man-Page. Es lohnt sich darüber hinaus, einen Blick in die Default-Konfigurationsdatei */etc/screenrc* zu werfen. Wenn Sie noch mehr zum Thema lesen möchten, gibt es bei den meisten Distributionen ein sehr gut geschriebenes *README* und eine *FAQ*. Das Verzeichnis, in dem sich diese Dateien befinden, hängt von der Distribution ab. Für Red Hat ist es */usr/share/doc/screen-3.9.5/*, für Debian */usr/doc/screen/* und für Mandrake */usr/doc/screen-3.9.5/*.

(Tipp: Sie können diese Dateien, wenn sie die Endung *.gz* haben, also *gzip*-komprimiert sind, mit dem Programm *zless* lesen -- dies ist z. B. bei Debian Linux der Fall.) Ansonsten gilt:

```
Send bugreports, fixes, enhancements, t-shirts, money, beer & pizza to  
screen@uni-erlangen.de
```

(hge)

Screen URLs

- [1] Die Homepage des GNU-Projektes *screen* ist <http://www.gnu.org/software/screen/>
- [2] Eine schöne Sammlung von Informationen ist unter <http://www.math.fu-berlin.de/~guckes/screen> zu finden.