

# Screen – der Terminal-Multiplexer

## Schöner leben mit der Konsole

Christian Perle

21. April 2008

# Gliederung

- 1 Kurze Einführung
- 2 Anwendungsfälle
- 3 Spezialitäten

# Was ist screen überhaupt?

- Ein „Window Manager“ fürs Terminal

# Was ist screen überhaupt?

- Ein „Window Manager“ fürs Terminal
- Man kann Sitzungen parken, sich ausloggen. . .  
(Sitzung läuft im Hintergrund weiter)

# Was ist screen überhaupt?

- Ein „Window Manager“ fürs Terminal
- Man kann Sitzungen parken, sich ausloggen. . .  
(Sitzung läuft im Hintergrund weiter)
- . . . und von woanders fortsetzen

# Aber die Shell kann das auch?!

- Die Shell kann Hintergrundprozesse ausführen, man kommt an diese nach dem Ausloggen nicht mehr interaktiv heran

# Aber die Shell kann das auch?!

- Die Shell kann Hintergrundprozesse ausführen, man kommt an diese nach dem Ausloggen nicht mehr interaktiv heran
- Manche Prozesse brauchen ein Terminal, um weiterlaufen zu können

# Begriffe

Screen unterscheidet zwischen **Sitzungen** und **Fenstern**:

- Eine screen-Sitzung ist ein laufender screen-Prozess mit einem oder mehreren Fenstern



# Begriffe

Screen unterscheidet zwischen **Sitzungen** und **Fenstern**:

- Eine screen-Sitzung ist ein laufender screen-Prozess mit einem oder mehreren Fenstern
- Fenster unterteilen eine Sitzung in mehrere virtuelle Terminals

# Starten und bedienen von screen

- Starten mit: `screen`

# Starten und bedienen von screen

- Starten mit: `screen`
- Wichtigste Taste (Hotkey): `ctrl-a`

# Starten und bedienen von screen

- Starten mit: `screen`
- Wichtigste Taste (Hotkey): `ctrl-a`
- Neues Fenster erzeugen: `ctrl-a c`

# Starten und bedienen von screen

- Starten mit: `screen`
- Wichtigste Taste (Hotkey): `ctrl-a`
- Neues Fenster erzeugen: `ctrl-a c`
- Aufs nächste Fenster schalten: `ctrl-a Space`

# Starten und bedienen von screen

- Starten mit: `screen`
- Wichtigste Taste (Hotkey): `ctrl-a`
- Neues Fenster erzeugen: `ctrl-a c`
- Aufs nächste Fenster schalten: `ctrl-a Space`
- Direkt auf Fenster N schalten: `ctrl-a N`

# Starten und bedienen von screen

- Starten mit: `screen`
- Wichtigste Taste (Hotkey): `ctrl-a`
- Neues Fenster erzeugen: `ctrl-a c`
- Aufs nächste Fenster schalten: `ctrl-a Space`
- Direkt auf Fenster N schalten: `ctrl-a N`
- Sitzung parken (detach): `ctrl-a d`

# Starten und bedienen von screen

- Starten mit: `screen`
- Wichtigste Taste (Hotkey): `ctrl-a`
- Neues Fenster erzeugen: `ctrl-a c`
- Aufs nächste Fenster schalten: `ctrl-a Space`
- Direkt auf Fenster N schalten: `ctrl-a N`
- Sitzung parken (detach): `ctrl-a d`
- Laufende Sitzungen auflisten: `screen -ls`



# Starten und bedienen von screen

- Starten mit: `screen`
- Wichtigste Taste (Hotkey): `ctrl-a`
- Neues Fenster erzeugen: `ctrl-a c`
- Aufs nächste Fenster schalten: `ctrl-a Space`
- Direkt auf Fenster N schalten: `ctrl-a N`
- Sitzung parken (detach): `ctrl-a d`
- Laufende Sitzungen auflisten: `screen -ls`
- Sitzung fortsetzen (reattach): `screen -r`

# Starten und bedienen von screen

- Starten mit: `screen`
- Wichtigste Taste (Hotkey): `ctrl-a`
- Neues Fenster erzeugen: `ctrl-a c`
- Aufs nächste Fenster schalten: `ctrl-a Space`
- Direkt auf Fenster N schalten: `ctrl-a N`
- Sitzung parken (detach): `ctrl-a d`
- Laufende Sitzungen auflisten: `screen -ls`
- Sitzung fortsetzen (reattach): `screen -r`
- Konfigurationsdatei: `.screenrc`

# Unterbrechung einer ssh-Verbindung

- Man ist per `ssh` auf einem Rechner eingeloggt und editiert eine wichtige Konfigurationsdatei

# Unterbrechung einer ssh-Verbindung

- Man ist per `ssh` auf einem Rechner eingeloggt und editiert eine wichtige Konfigurationsdatei
- Die `ssh`-Verbindung bricht ab, automatischer Detach

# Unterbrechung einer ssh-Verbindung

- Man ist per `ssh` auf einem Rechner eingeloggt und editiert eine wichtige Konfigurationsdatei
- Die `ssh`-Verbindung bricht ab, automatischer Detach
- Neu einloggen, dann reattach mit `screen -r`

# Unterbrechung einer ssh-Verbindung

- Man ist per `ssh` auf einem Rechner eingeloggt und editiert eine wichtige Konfigurationsdatei
- Die `ssh`-Verbindung bricht ab, automatischer Detach
- Neu einloggen, dann reattach mit `screen -r`
- Weiterarbeiten, als wäre nichts gewesen :-)

# Persistenter „Konsolendesktop“

- **Anwendungen in separaten Fenstern:**  
`mutt, irssi, w3m, mc...`

# Persistenter „Konsolendesktop“

- Anwendungen in separaten Fenstern:  
`mutt, irssi, w3m, mc...`
- Fenster sind nach den Anwendungen benannt, um sie einfacher wiederzufinden



# Persistenter „Konsolendesktop“

- Anwendungen in separaten Fenstern:  
`mutt`, `irssi`, `w3m`, `mc`...
- Fenster sind nach den Anwendungen benannt, um sie einfacher wiederzufinden
- Zusätzliche Informationen in der Statuszeile

# Hilfesitzung mit ssh und screen

- Lokaler Benutzer startet `screen`

# Hilfesitzung mit `ssh` und `screen`

- Lokaler Benutzer startet `screen`
- Helfer loggt sich mit `ssh` als gleicher Benutzer ein

# Hilfesitzung mit `ssh` und `screen`

- Lokaler Benutzer startet `screen`
- Helfer loggt sich mit `ssh` als gleicher Benutzer ein
- Helfer hängt sich an die existierende Sitzung dran:  
`screen -x`

# Hilfesitzung mit ssh und screen

- Lokaler Benutzer startet `screen`
- Helfer loggt sich mit `ssh` als gleicher Benutzer ein
- Helfer hängt sich an die existierende Sitzung dran:  
`screen -x`
- Beide können nun tippen/zusehen/korrigieren

# Technisches

- Zwei Prozesse:

# Technisches

- Zwei Prozesse:
  - SCREEN (Backend, Terminalemulation)

# Technisches

- **Zwei Prozesse:**
  - `SCREEN` (Backend, Terminalemulation)
  - `screen` (Frontend, sichtbare Darstellung im Terminal)



# Technisches

- Zwei Prozesse:
  - `SCREEN` (Backend, Terminalemulation)
  - `screen` (Frontend, sichtbare Darstellung im Terminal)
- Frontend und Backend kommunizieren über einen lokalen Socket

# Technisches

- Zwei Prozesse:
  - `SCREEN` (Backend, Terminalemulation)
  - `screen` (Frontend, sichtbare Darstellung im Terminal)
- Frontend und Backend kommunizieren über einen lokalen Socket
- In einer `screen`-Sitzung ist die Umgebungsvariable `TERM` auf `screen` gesetzt

# Statuszeile

- Sitzung ist deutlicher erkennbar  
*„bin ich in screen oder nicht?“*

# Statuszeile

- Sitzung ist deutlicher erkennbar  
*„bin ich in screen oder nicht?“*
- Bessere Übersicht über die Fenster

# Statuszeile

- Sitzung ist deutlicher erkennbar  
*„bin ich in screen oder nicht?“*
- Bessere Übersicht über die Fenster
- Syntax: `hardstatus` *optionen format*

# Statuszeile

- Sitzung ist deutlicher erkennbar  
*„bin ich in screen oder nicht?“*
- Bessere Übersicht über die Fenster
- Syntax: `hardstatus` *optionen format*

# Statuszeile

- Sitzung ist deutlicher erkennbar  
*„bin ich in screen oder nicht?“*
- Bessere Übersicht über die Fenster
- Syntax: `hardstatus` *optionen format*

## Beispiel:

```
hardstatus alwayslastline \  
"%?%{wb}%-Lw%?%{Yb}%n*%f %t%?(%u)%?%?%{wb}%+Lw%?%{wb}%=%c"
```

# Scrollback Buffer, Copy/Paste ohne Maus

Jedes screen-Fenster besitzt seinen eigenen Scrollback Buffer

- Copy/Scrollmodus aktivieren `ctrl-a ESC`



# Scrollback Buffer, Copy/Paste ohne Maus

Jedes screen-Fenster besitzt seinen eigenen Scrollback Buffer

- Copy/Scrollmodus aktivieren `ctrl-a ESC`
- Cursor bewegen `Cursortasten, PgUp, PgDown`

# Scrollback Buffer, Copy/Paste ohne Maus

Jedes screen-Fenster besitzt seinen eigenen Scrollback Buffer

- Copy/Scrollmodus aktivieren `ctrl-a ESC`
- Cursor bewegen `Cursortasten, PgUp, PgDown`
- Auswahlmarken setzen `Space`

# Scrollback Buffer, Copy/Paste ohne Maus

Jedes screen-Fenster besitzt seinen eigenen Scrollback Buffer

- Copy/Scrollmodus aktivieren `ctrl-a ESC`
- Cursor bewegen `Cursortasten, PgUp, PgDown`
- Auswahlmarken setzen `Space`
- Auswahl ins Terminal einfügen `ctrl-]`

# Scrollback Buffer, Copy/Paste ohne Maus

Jedes screen-Fenster besitzt seinen eigenen Scrollback Buffer

- Copy/Scrollmodus aktivieren `ctrl-a ESC`
- Cursor bewegen `Cursortasten, PgUp, PgDown`
- Auswahlmarken setzen `Space`
- Auswahl ins Terminal einfügen `ctrl-]`
- Funktioniert auch zwischen verschiedenen Fenstern

# Überwachen von Aktivität/Inaktivität

Ausgaben oder längere Inaktivität in einem nicht sichtbaren Fenster lassen sich von screen melden

- Aktivität überwachen  
z.B. das Fenster eines Chat-Clients

```
ctrl-a M
```

# Überwachen von Aktivität/Inaktivität

Ausgaben oder längere Inaktivität in einem nicht sichtbaren Fenster lassen sich von screen melden

- Aktivität überwachen `ctrl-a M`  
z.B. das Fenster eines Chat-Clients
- Inaktivität überwachen `ctrl-a _`  
z.B. einen Compilerlauf

# Split View

- Zwei Fenster einer Sitzung gleichzeitig darstellen

# Split View

- Zwei Fenster einer Sitzung gleichzeitig darstellen
- Splitten

```
ctrl-a S
```



# Split View

- Zwei Fenster einer Sitzung gleichzeitig darstellen
- Splitten `ctrl-a S`
- Umschalten `ctrl-a TAB`

# Multiuser screen

- zwei **verschiedene** Benutzer teilen sich eine Sitzung

# Multiuser screen

- zwei **verschiedene** Benutzer teilen sich eine Sitzung
- unterschiedliche Berechtigungen möglich (ACLs), vergeben vom **Owner** der Sitzung

# Multiuser screen

- zwei **verschiedene** Benutzer teilen sich eine Sitzung
- unterschiedliche Berechtigungen möglich (ACLs), vergeben vom **Owner** der Sitzung
- Nachteil: screen muss setuid root laufen

# „Fernsteuern“ einer Sitzung

- Sitzung benutzen, ohne zu attachen

# „Fernsteuern“ einer Sitzung

- Sitzung benutzen, ohne zu attachen
- Syntax: `screen -X kommando`

# „Fernsteuern“ einer Sitzung

- Sitzung benutzen, ohne zu attachen
- Syntax: `screen -X kommando`

# „Fernsteuern“ einer Sitzung

- Sitzung benutzen, ohne zu attachen
- Syntax: `screen -X kommando`

Beispiel:

Aufs nächste Fenster schalten

```
screen -X next
```



# Keybindings

- **Browser mit Freshmeat auf `ctrl-a W` legen:**  
`bind W screen w3m http://freshmeat.net/`

# Keybindings

- **Browser mit Freshmeat auf `ctrl-a W` legen:**  
`bind W screen w3m http://freshmeat.net/`
- **Sperrfunktion von screen deaktivieren:**  
`bind x next`

# screen detached starten

- Für Dienste mit ncurses-Interface

# screen detached starten

- Für Dienste mit ncurses-Interface
- `mp3blaster` mit großer Playlist im Hintergrund starten, bei Bedarf wird die Sitzung attached und es kann interaktiv eingegriffen werden:

```
screen -d -m mp3blaster -a /etc/playlist
```

# screen als minicom-Ersatz

- Kommunikation über serielle Leitung

# screen als minicom-Ersatz

- Kommunikation über serielle Leitung
- Syntaxbeispiel:

```
screen /dev/ttyS0 57600 8n1
```

# Zitat aus der Manpage

Es geht noch viel mehr...

„A weird imagination is most useful to gain full advantage of all the features.“