

# Remote Control von Desktops

mittels *x11vnc* & *tigervnc*

# Die Geschichte vom kleinen Nuc



# Das Wesen des Nuc ist *Plasma*

The screenshot displays a Linux desktop environment with a dark, hexagonal pattern background. On the left, a terminal window shows system statistics for 'ch1-1-ARCH on x8'. In the center, a weather widget displays the date 'sa. 06 JULI' and time '17:26:42', along with core temperatures and CPU usage. The bottom of the screen features a taskbar with system icons and the time '17:26'.

```
ch1-1-ARCH on x8
-----
Uptime: 0h 0m 45s
Frequency (in MHz): 1558
Frequency (in GHz): 1,56
RAM Usage: 515MiB/15,6GiB - 3%
Swap Usage: 0B/8,80GiB - 0%
CPU Usage: 2%
Processes: 148 Running: 0
-----
File systems:
 / 23,2GiB/219GiB
Networking:
Up: 0B - Down: 0B
-----
Name      PID  CPU%  MEM%
Xorg      490   1,50  0,44
x11vnc   1095   0,50  0,18
plasmashell 700   0,25  1,10
conky     838   0,25  0,05
```

sa. 06 JULI 17:26:42 Uptime: 0d 0:00

Core 0: 38°C CPU 0: 2%  
Core 1: 39°C CPU 1: 3%  
CPU 2: 3%  
CPU 3: 3%

Arch Linux  
Linux 5.1.7

Mem: 15.56 GiB  
Used: 0.32 GiB Free: 15.24 GiB

Swap: 8.80 GiB  
Used: 0.00 GiB Free: 8.80 GiB

17:26

# Seine Aufgabe ist die Aufnahme von TV-Programmen mit *Kaffeine*

The screenshot displays a Linux desktop environment. On the left, a terminal window shows system statistics:

```
x86_64 |
Uptime: 0h 13m 45s
Frequency (in MHz): 2260
Frequency (in GHz): 2,26
RAM Usage: 705MiB/15,6GiB - 4%
Swap Usage: 0B/8,80GiB - 0%
CPU Usage: 21%
Processes: 142 Running: 1

File systems:
/ 23,2GiB/219GiB
Networking:
Up: 0B - Down: 0B

Name PID CPU% MEM%
kaffeine 1136 10,24
x11vnc 1095 7,87
Xorg 490 3,94
kwin_x11 686 1,31
```

In the center, a date display shows "sa. 06 JULI".

The main window is the Kaffeine media player, titled "arte HD - Kaffeine". It features a channel selection menu on the left with "arte HD" selected. The video player shows a woman in a white tank top, with the text "arte REPORTAGE" overlaid. The playback controls at the bottom indicate the video is in German ("deu") and has a duration of 00:00:15.

The system tray at the bottom right shows the time as 17:39.

Dienstag, 6. August 2019

4 / 26



**Aber leider ist das  
Satellitenkabel am  
Fernseher zu kurz**



**Also wird der kleine Nuc  
kurzerhand neben den  
Fernseher postiert und  
der DVB-S2-Receiver mit  
dem Satellitenkabel und  
über USB mit dem Nuc  
verbunden**



**Weil aber ein weiterer  
PC-Monitor dort keinen  
Platz findet, muss also  
der Ferneher mittels  
HDMI-Anschluss ange-  
schlossen werden, damit  
man am Nuc arbeiten  
kann. Ferner fehlen  
noch Tastatur und Maus.**



**Sehr umständlich, das  
ganze Drumherum,  
nicht wahr?**



**Was bietet sich also an,  
um TV, Maus und  
Tastatur ‚einzusparen‘?**

**Mmmmmh....**

**Richtig! Es gibt ja die  
‚Fernsteuerung‘ von PCs,  
das ‚Remote Control‘  
Gute Sache das.**

**Mir fällt dazu von ganz  
früher *Fernterm* für DOS  
ein oder auch *Teamviewer*.  
Nee, zu mächtig, und eher  
eine *Wine*-Emulation.  
Bleiben für Linux spezielle  
X-Windows-Lösungen.**

**Als da wäre *X2Go*  
(*Client und Server*)**

**Leider haben sich mir da  
zu viele Nachteile  
gezeigt.**

**Nach ausgiebiger  
Recherche stieß ich  
dann auf eine  
zweigeteilte Lösung:**



***X11VNC als Server  
&  
Tigervnc als Client***

# 1a. Einrichten des Servers *x11vnc* auf dem Remote

- Das Programm bzw. das Paket *x11vnc* gibt es für alle gängigen Distributionen.
- *X11vnc* ist der Server und läuft auf dem entfernten Rechner (Remote), hier der *Nuc*.
- *X11vnc* wartet auf Anfragen von Clienten. Diese Anfragen kommen z.B von *vncviewer*, einem Programmteil von *tigervnc*

# 1b. Einrichten des Servers *x11vnc* auf dem Remote

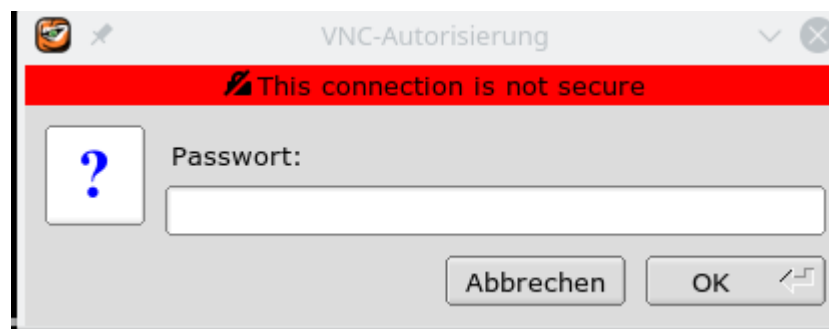
- Nachdem *x11vnc* heruntergeladen und installiert worden ist, muss der Server, am besten als *Benutzer* eingerichtet werden und später auch vom *Benutzer* gestartet werden.
- *X11vnc* wird initialisiert mit *vncserver*, welches ein Zugangspasswort abfragt und eine Grundkonfiguration des Servers erstellt.
- Diese Grundkonfiguration in `~/.vnc/xstartup` muss noch an den verwendeten Desktop angepasst werden!

## 2. Starten des Servers *x11vnc* auf dem Remote

- Man kann allerdings *x11vnc* mit den passenden Parametern *manuell* starten, wenn bereits eine Desktopumgebung wie *xfce*, *Gnome*, *Plasma* oder andere gestartet sind. Dann muss der Server(als Benutzer!) z.B. so gestartet werden:
- `/usr/bin/x11vnc -auth guess -geometry 1920x1080 -rfbauth /home/user/.vnc/passwd -shared -bg -repeat -forever -display :0`

# 3. Einrichten des Client-Programms *vncviewer*

- Das Programmpaket *Tigervnc* wird aus den Repositorien auf dem *Clienten* installiert.
- Der *Tigervnc* enthält das Programm *vncviewer*
- Dieser *vncviewer* wird als *Benutzer* gestartet, z.B. mit *vncviewer 192.168.178.xx:0*
- Unter *Arch* ist sein Erscheinungsbild simpel:





# 4. Verbindung mit dem Remote-Desktop des Servers

- Wird das korrekte Passwort, welches zuvor beim *Server* mittels *x11vnc -storepasswd* hinterlegt wurde, eingegeben, so sollte entweder das *Login*-Fenster oder der bereits laufende Desktop des *Remote*-Rechners angezeigt werden und falls nicht beschränkt, kann man dort mit Maus und Tastatur arbeiten.
- Sogar der Austausch von Zwischenablagen ist möglich (Drag`n Drop)!

# 5. Besonderheiten des *x11vnc* - Servers

- Wie gesagt, *x11vnc* kann manuell als Benutzer gestartet werden.
- Dieser Server kann aber auch schon beim Booten des Rechners gestartet werden, zumindest mit *Systemd*.
- Dafür muss zuvor auch ein *Login-Manager* (*sddm* oder *lightdm*) gestartet sein, was standardmäßig der Fall ist.

# 6. Der *Systemd*-Service für */etc/systemd/system/x11vnc@*

[Unit]

Description=VNC Server for X11

Requires=display-manager.service

After=display-manager.service

[Service]

Type=forking

ExecStart=/usr/bin/x11vnc -auth guess -geometry 1920x1080 -rfbauth  
/home/richard/.vnc/passwd -shared -bg -repeat -forever -display :0

Restart=on-failure

RestartSec=10

User=%l

[Install]

WantedBy=multi-user.target

# 7. Aktivieren und Starten des *Systemd* - Services

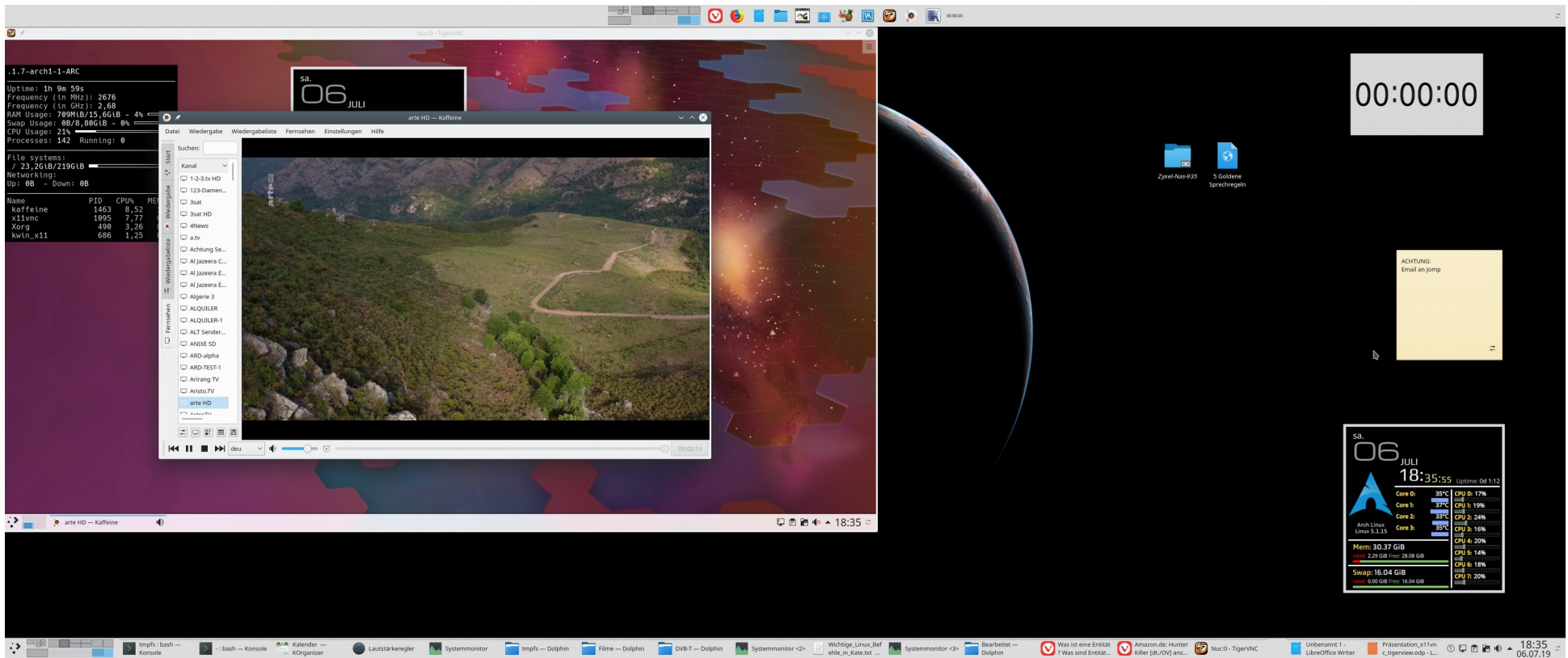
- Normalerweise werden *alle* Systemd-Services während des Bootens als *root* gestartet.
- Damit ein Service als *Benutzer* gestartet werden soll, gibt es im Servicenamen das @-Symbol, um den Service mit dem Namen des Benutzers zu verknüpfen. So wird solch ein Service z.B. aktiviert (enabled) mit:

```
systemctl enable x11vnc@benutzer.service
```


- Das ist in etwa so, als ob *x11vnc* vom Benutzer gestartet wurde.




**Nach Herstellung der  
Netzwerkverbindung mit  
*Nuc* erhalte ich auf meinem  
(dicken) Clienten das  
folgende Bild:**



Man beachte das linke, große Fenster!




**In diesem Fenster ist der  
*Nuc* mit 1920x1080 Pixeln  
als Fenster geöffnet. Jetzt  
kann ich mittels Maus und  
Tastatur auf dem *Nuc*  
arbeiten, also  
Konsolen öffnen, *Kaffeine*  
starten und zum  
Aufnehmen programmieren  
usw....**



**Ist meine Programmierung  
zur Aufnahme von TV-  
Programmen erledigt, kann  
ich ganz einfach das  
*Clienten-Fenster*  
oben rechts (x) schließen,  
während der *Nuc* seine Zeit  
abwartet, bis er aufnimmt.**





**Ich kann ihm sogar sagen,  
er soll sich selbst  
ausschalten.  
Und: ich kann den *Nuc*  
mittels *WakeOnLan*  
aufwecken, sobald mein  
Client (der Dicke) aktiv ist.**

# Literatur zu *x11vnc* und *tigervnc*

- *X11vnc*: <http://www.karlrunge.com/x11vnc/>  
→ homepage des Entwicklers
- *X11vnc* auf Arch:  
→ <https://wiki.archlinux.org/index.php/X11vnc>