Dokumentation

QEMU

Inhalt:

1. Installation von KQEMU

Installation der Kernelheader um KQEMU zu kompilieren:

```
apt-get install linux-headers-2.6.18-6-686
```

KQEMU kompilieren:

```
tar xzf kqemu-1.3.0pre11.tar.gz
cd kqemu-1.3.0pre11
./configure
make
make install
```

Kernelmodul laden:

```
modprobe kqemu
```

`/sbin/modprobe kqemu` (in /etc/rc.local am Ende einfügen)

Erstellen einer Networkbridge:

```
apt-get install bridge-utils
apt-get install uml-utilitis

brctl addbr br0
ifconfig eth0 0.0.0.0 promisc up
brctl addif br0 eth0

ifconfig br0 192.168.99.102
```

Anpassen der Netzwerkschnittstelle:

(*Damit die Einstellungen nach einem Neustart beibehalten werden*)

/etc/network/interface

```
# The bridge network interface(s)
auto br0
iface br0 inet static
 address 192.168.1.3
 network 192.168.1.0
 netmask 255.255.255.0
 broadcast 192.168.1.255
 gateway 192.168.1.1
 bridge_ports eth0
 bridge_fd 9
 bridge_hello 2
 bridge_maxage 12
 bridge_stp off

#auto eth0
#iface eth0 inet dhcp
```

Anpassen der QEMU Interface-Datei:

(Damit das Interface der Bridge hinzugefügt wird)

/etc/qemu-ifup

```
#!/bin/sh
sudo /sbin/ifconfig $1 0.0.0.0 promisc up
sudo /usr/sbin/brctl addif br0 $1

chmod 666 /dev/net/tun
```

Skript um die Tunneling-Devices anzulegen:

(Wird ab Kernel >= 2.6.18 benötigt. Zum Starten wird dieses Skript verwendet.)

/usr/bin/qemu-tap

```
#!/bin/sh
# script to manage tap interface allocation
# for linux kernels >= 2.6.18

# set up a tap interface for qemu
# USERID - uid qemu is being run under.
USERID=`whoami`
iface=`sudo /usr/sbin/tunctl -b -u $USERID | xargs echo -n `
echo -n interface
echo $iface
# generate a random mac address for the qemu nic
# shell script borrowed from user pheldens @ qemu forum
ranmac=$(echo -n DE:AD:BE:EF ; for i in `seq 1 2` ; \
do echo -n `echo ":$RANDOM$RANDOM" | cut -n -c -3` ;done)

# specify which NIC to use - see qemu.org for others
model=ne2k_pci
# model=ne2k_isa

# start qemu with our parameters
/usr/bin/qemu $@ -net nic,vlan=0,macaddr=$ranmac,
model=$model \
-net tap,vlan=0,ifname=$iface

# qemu has stopped - no longer using tap interface
sudo /usr/sbin/tunctl -d $iface &> /dev/null
```

Freigabe der Rechte für alle User, die Tunneling-Devices mit dem Skript anlegen:

/etc/sudoers

```
ALL       ALL = NOPASSWD: /sbin/ifconfig
ALL       ALL = NOPASSWD: /usr/sbin/brctl
ALL       ALL = NOPASSWD: /usr/sbin/tunctl
```

2. Installation von QEMU

```
cd /

tar xzf qemu-0.9.1-i386.tar.gz
```

Installation des QEMU-Launcher:

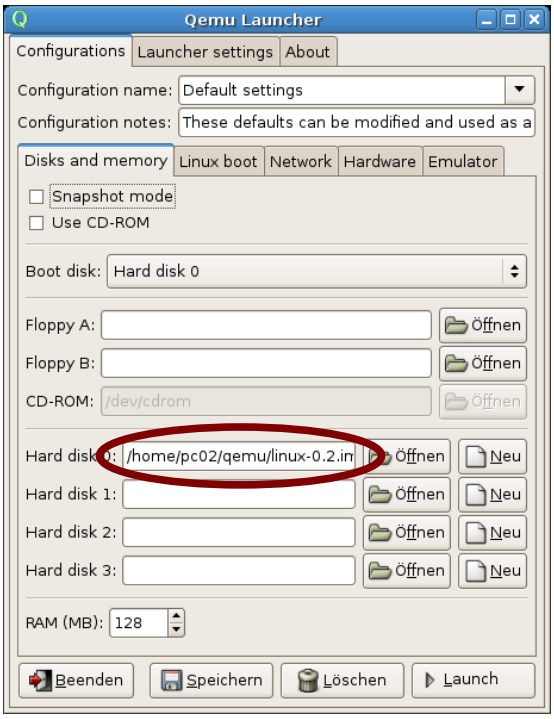
```
apt-get install qemu-launcher
```

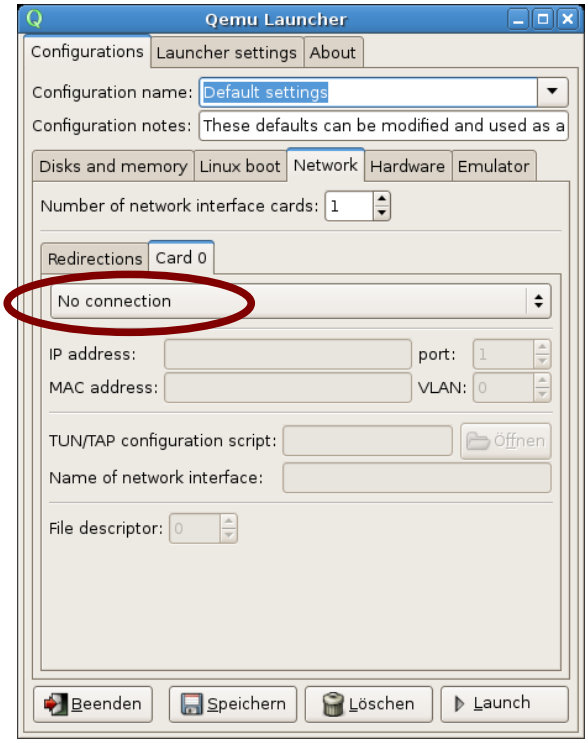
3. Starten eines Images

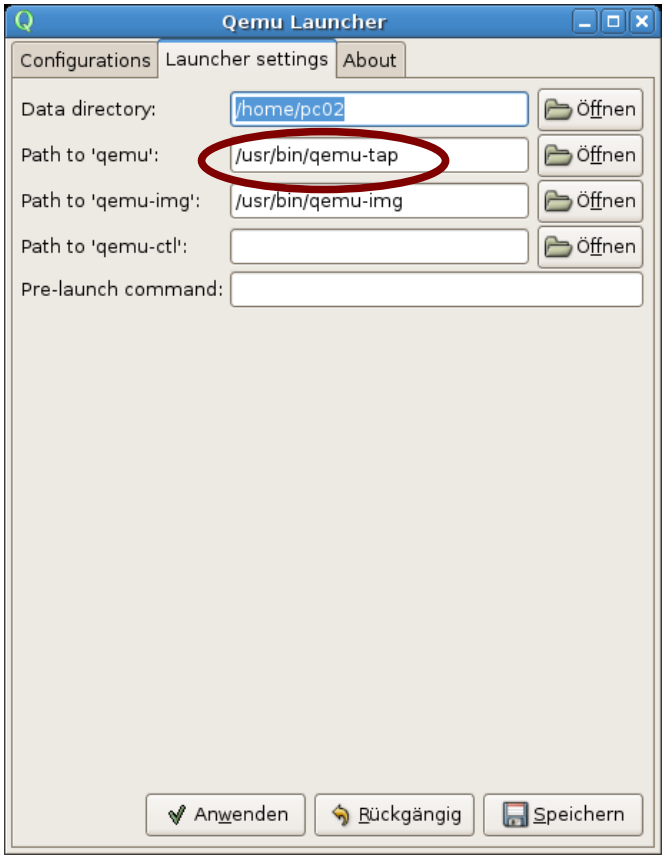
über Konsole:

```
qemu -kernel-kqemu linux-0.2.img
```

über QEMU Launcher:




Qemu Launcher
Configurations
Launcher settings
About
Configuration name:
Default settings
Configuration notes:
These defaults can be modified and used as a
Disks and memory
Linux boot
Network
Hardware
Emulator
Number of network interface cards:
1
Redirections
Card 0
No connection
IP address:
port:
MAC address:
VLAN:
TUN/TAP configuration script:
Öffnen
Name of network interface:
File descriptor:
Beenden
Speichern
Löschen
Launch


Qemu Launcher
Configurations
Launcher settings
About
Data directory:
/home/pc02
Öffnen
Path to 'qemu':
/usr/bin/qemu-tap
Öffnen
Path to 'qemu-img':
/usr/bin/qemu-img
Öffnen
Path to 'qemu-ctl':
Öffnen
Pre-launch command:
Anwenden
Rückgängig
Speichern

Getestet wurde QEMU mit folgendem Hostsystem:

Intel Pentium 4 2.8 GHz und 512 MB RAM.
Als Gastsystem wurde Debian etch verwendet (fertiges Image).
Die Bereitstellung der Ressourcen ist vom Hostsystem abhängig und muss bei Gastsystemen mit höherer Ressourcenanforderung berücksichtigt werden.

4. Quellen

http://fabrice.bellard.free.fr/qemu/

http://www.oszoo.org/wiki/index.php/Main_Page

http://compsoc.dur.ac.uk/~djw/qemu.html

http://www.tldp.org/HOWTO/BRIDGE-STP-HOWTO/set-up-the-bridge.html

http://calamari.reverse-dns.net:980/cgi-bin/moin.cgi/FrequentlyAskedQuestions#head-2511814cb92c14dbe1480089c04f83c281117a86