ctys-uc-MeeGo(7)
Setup MeeGo

March 12, 2011

Contents

List of Figures

.

1 General

The current document shows the basic installation of MeeGo, which is basically a derivative of RedHat(TM) Linux variant.

The following environment is used here:

- Debian-5.0.6 with VirtualBox-3.2.10

- CentOS-5.4 with kvm-83 / Qemu-0.9.1

- MeeGo-1.0.0
  The current description is based on the Netbook edition for ia32/i386 architecture. Download the image:

  ```
  Netbooks/meego-netbook-ia32-1.0.0.20100524.1.img
  ```

  Additionally download the packages as required. For the installation and execution of ctys at least the 'openssh-server' is required.

- UnifiedSessionsManager - ctys-01.11.011

.

2 Setup of Host-OS and Hypervisor

The installation for the following variants has to be performed by the appropriate standard setup of the Host OS, which quite straight forward:

- Debian with VirtualBox

- CentOS with QEMU/KVM

3 Setup of the UnifiedSessionsManager

3.1 Install tgz BASE-Package + DOC-Package on Debian

1. Apply the standard installation procedure:

   ```
   ctys-distribute -F 2 -P UserHomeCopy root@tst137
   ```

2. Open a Remote Shell by call of CLI plugin:

   ```
   ctys -t cli -a create=l:tst137 root@tst137
   ```

3. Check the plugins states by calling ctys-plugins:

   ```
   ctys-plugins -T all -E
   ```

3.2 Install rpm BASE-Package + DOC-Package on CentOS

The following steps are required for a RPM based setup on CentOS. The installation is relocatable, but located at '/opt', and installed locally by ctys-distribute(1) .

1. Install BASE package.

   ```
   rpm -i ctys-base-01.11.011.noarch.rpm
   ```

2. Now install a a local version, here by copy. The PATH prefix is important here, particularly in case of updates. The path is resolved to it's actual path by eliminating any symbolic link, and used for consistent link of libraries.

```
/opt/ctys-01.11.011/bin/ctys-distribute -F 2 -P UserHomeCopy
```

3. Next the menu is setup.

```
ctys-xdg --menu-create
```

4. Now the help is available as eihter a Gnome or KDE menu. Alternatively could be called from the commandline.

3.3 Setup of the Gnome Menue

The setup of the Gnome Menu is quite simple, the contained tool **ctys-xdg(1)** sets up a standard menu by the call:

```
ctys-xdg --menu-create
```
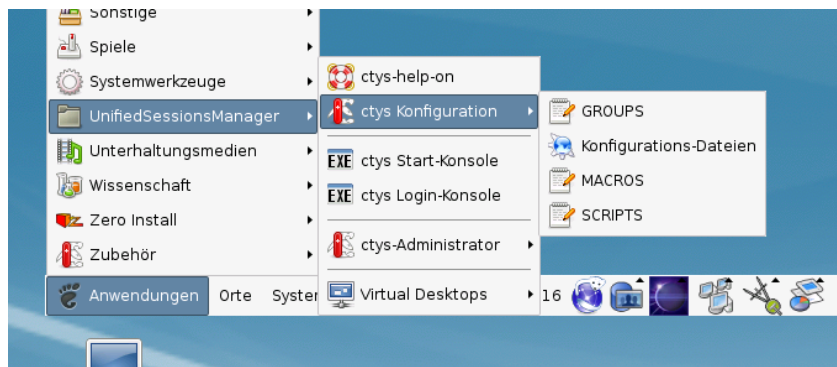

Figure 1: Create Menue

The call

```
ctys-xdg --menu-cancel
```

removes the installed files. For current version no checks for changed files is done.

The menues could be edited and extended by the call

```
ctys-xdg --menu-edit
```

which opens the related directories for modification of '*.menu', '*.desktop', and '*.directory' files.

4 Creation of the the Raw-VM

4.1 Creation of the Raw-VM with QEMU/KVM

The demo example VM is here named tst139, this is the hostname of GuestOS too.

1. Login into the machine where VirtualBox is installed.

```
ssh -X lab02
```

2. Change to the vmpool and create a directory and change into.

```
mkdir tst139
```

3. Call the install and configuration utility for VMs. Here some values are set by environment variables, a complete list including the actually assigned values could be displayed by the option **–levo**.

```
ARCH=i386 \
DIST=MeeGo \
DISTREL=1.0.0 \
OS=Linux \
OSREL=2.6 \
ctys-createConfVM -t qemu --label=tst138
```

This call creates a virtual image(hda.img), the call-wrapper(tyt139.sh), and the configuration file(tst139.ctys). The files are created from templates by assigning configuration values either from pre-configured default values, or interactive variation.

4. Once the set of files is created the virtual machine is prepared for startup. For some other systems complete installation routines are available, e.g. debian and CentOS. The current state could be checked now by the following call.

```
./tst138.sh --console=vnc  --vncaccessdisplay=47 --print --check
```

4.2   Creation of the Raw-VM with VirtualBox

The creation of the raw VM is first step to be executed at the host oeprating system. This could be either performed locally or remote and requires the usage of the provided tools by VirtualBox(TM).

1. Login into the machine where VirtualBox is installed.

```
ssh -X lab02
```

2. Execute the VirtualBox(TM) console.

```
VirtualBox
```

3. Create the VM, the machine is called here 'tst137'. The OS is 'Linux', the version is 'Linux 2.6'.

Figure 2: Create Virtual Machine

4. Set RAM to 512MByte.

Figure 3: Set virtual RAM

5. Create a virtual HDD, here 8GByte is choosen. When finished the raw VM is present and could be used as required, for basic functions of ctys no additional configuration is required.
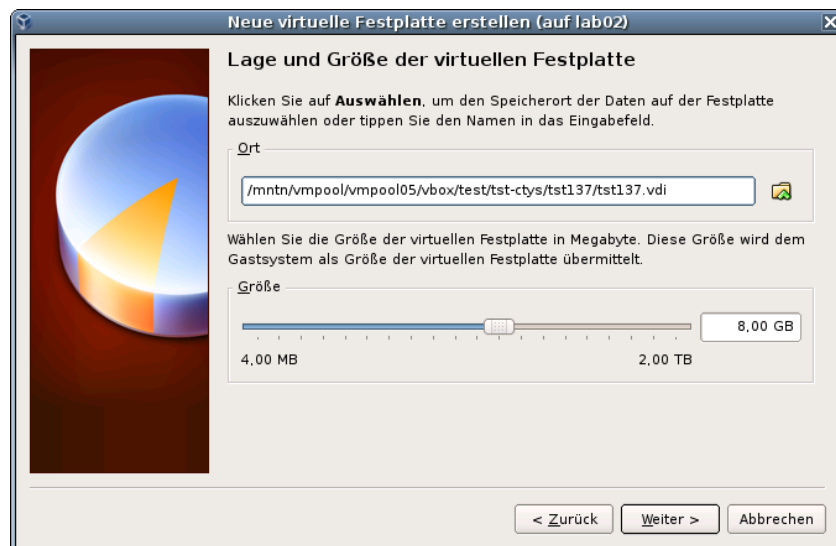

Figure 4: Create Virtual HDD

6. When additional information should be stored coallocated to the VM and scanned automatically into a database, than the tool **ctys-createConfVM(1)** should be applied. This generates additional detailed information related to the specific VM and the inherent guest OS.
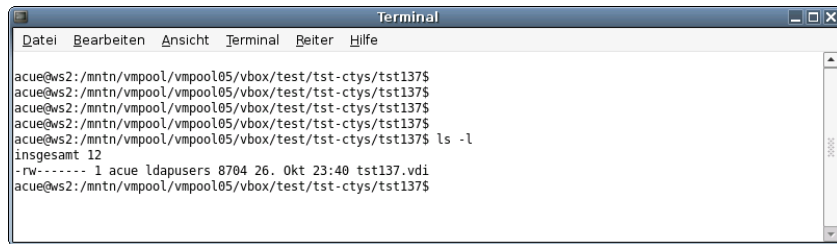
Figure 5: Check HDD image file

The call could be executed either interactive or automatic.

Call within the same directory for first inspection:

```
ctys-createConfVM -t vbox --label=tst137 --levo
```

This lists some defaults for the specific hypervisor. These could be preconfigured by specific template files within the configuration directory **ctys-createCOnfVM.d**. The result should look like the following:

```
Not all values require to be set, some will be requested later by
dialogue.
Thus it is not neccessary to have values assigned to the complete
displayed set.


Actually used sources for default values:
  no-marker  = Pre-Set value, either from defaults configuration, or
               by commandline.
  no-value   = Either requested by dialog later, or the defaults of
               the finally called
               application are used.
  (g)        = Dynamically generated.
  (c)        = Read from actual configuration file, e.g. vmx-file.
  (h)        = Used from current host as default.

Applicable modifications:
  blue       = By call option, defines dependency for others.
  green      = By environment, 'could be set almost independent'
               from other values.
  cyan       = By miscellaneous facilities, but is dependent from
               others.
               E.g. LABEL defines by convention the network
               'hostname', thus the TCP/IP params.
               This could ..., but should not be altered!

Most of the missing values will be fetched during actual execution
of this tool by dynamic evaluation.


              VAR name:Initial Value

         C_SESSIONTYPE:VBOX
                 LABEL:tst137
                   MAC:
                    IP:
                BRIDGE:
```

```
                       DHCP:
                    NETMASK:
                        TCP:
                    GATEWAY:

                     EDITOR:acue

                       UUID:b1ff0d36-a552-41ce-be3c-4b3717c2e768 (c)

                       DIST:debian (h)
                    DISTREL:5.0.6 (h)
                         OS:Linux (h)
                      OSREL:2.6.26-2-amd64 (h)

                       ARCH:x86_64 (h)
                ACCELERATOR:HVM (c)
                        SMP:1 (c)
                    MEMSIZE:512 (c)
                 KBD_LAYOUT:de

                 STARTERCALL:/usr/bin/VirtualBox

            DEFAULTBOOTMODE:HDD

          DEFAULTINSTTARGET:/mntn/vmpool/vmpool05/vbox/test/tst-ctys...
                            .../tst137/tst137.vdi
     HDDBOOTIMAGE_INST_SIZE:8192M

                    VMSTATE:ACTIVE
```

```
Remember that his is a draft pre-display of current defaults.
No consistency-checks for provided values are performed at this stage.
Some missing values are evaluated at a later stage dynamically.
```

The following call generates the appropriate configuration

```
DIST=MeeGo \
DISTREL=1.0.0 \
OS=Linux \
OSREL=2.6 \
MAC=00:50:56:13:11:65 \
IP=172.20.2.241 \
ARCH=i386 \
ctys-createConfVM --label=tst137 -t vbox \
```

The result displayed with –**levo** is:

```
Not all values require to be set, some will be requested later by
dialogue.
Thus it is not neccessary to have values assigned to the complete
displayed set.
```

```
Actually used sources for default values:
  no-marker  = Pre-Set value, either from defaults configuration,
               or by commandline.
```

```
    no-value   = Either requested by dialog later, or the defaults
                  of the finally called
                  application are used.
    (g)        = Dynamically generated.
    (c)        = Read from actual configuration file, e.g. vmx-file.
    (h)        = Used from current host as default.


Applicable modifications:
    blue       = By call option, defines dependency for others.
    green      = By environment, 'could be set almost independent'
                  from other values.
    cyan       = By miscellaneous facilities, but is dependent
                  from others.
                  E.g. LABEL defines by convention the network
                  'hostname', thus the TCP/IP params.
                  This could ..., but should not be altered!


Most of the missing values will be fetched during actual execution
of this tool by dynamic evaluation.



                    VAR name:Initial Value

             C_SESSIONTYPE:VBOX
                     LABEL:tst137
                       MAC:00:50:56:13:11:65
                        IP:172.20.2.241
                    BRIDGE:
                      DHCP:
                   NETMASK:
                       TCP:
                   GATEWAY:

                    EDITOR:acue

                      UUID:b1ff0d36-a552-41ce-be3c-4b3717c2e768 (c)

                      DIST:MeeGo
                   DISTREL:1.0.0
                        OS:Linux
                     OSREL:2.6

                      ARCH:i386
               ACCELERATOR:HVM (c)
                       SMP:1 (c)
                   MEMSIZE:512 (c)
                KBD_LAYOUT:de

               STARTERCALL:/usr/bin/VirtualBox

           DEFAULTBOOTMODE:HDD

         DEFAULTINSTTARGET:/mntn/vmpool/vmpool05/vbox/test/tst-ctys/...
                           ...tst137/tst137.vdi
  HDDBOOTIMAGE_INST_SIZE:8192M

                   VMSTATE:ACTIVE
```

Remember that his is a draft pre-display of current defaults.
No consistency-checks for provided values are performed at this stage.
Some missing values are evaluated at a later stage dynamically.

When the call is finished the file 'tst137.ctys' with additional configuration information information is stored.

7. Add the install image as a bootable CD/DVD and set this as the boot device fir the VM:
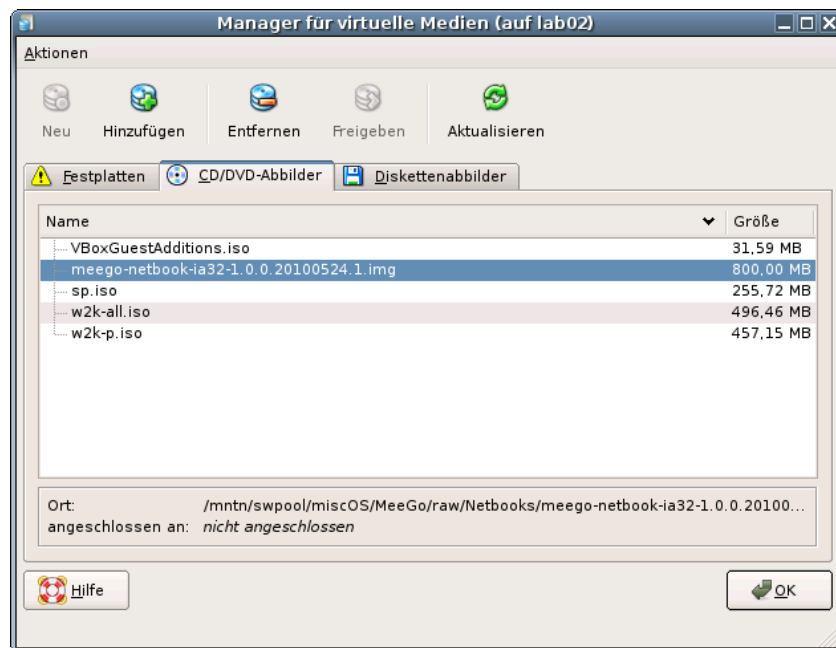
Netbooks/meego-netbook-ia32-1.0.0.20100524.1.img
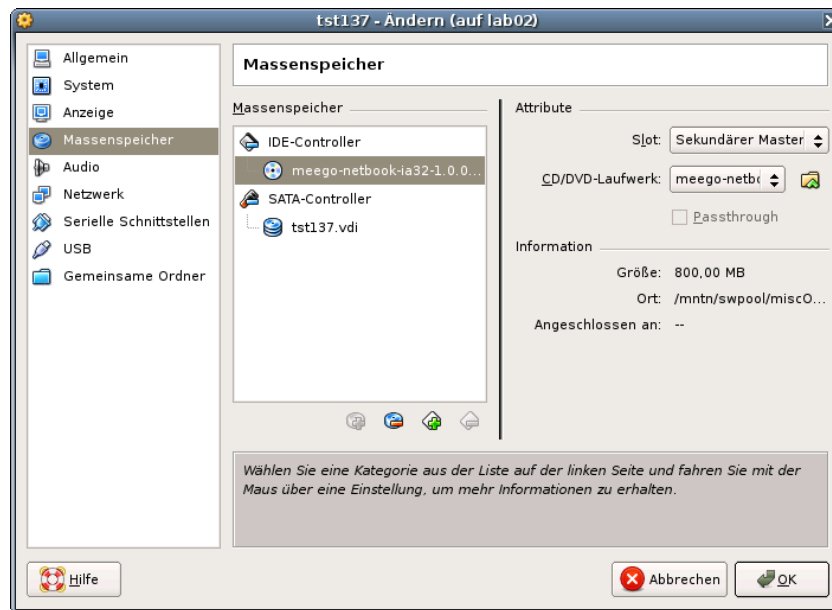

Figure 6: Register CD/DVD Install Sources

Figure 7: Connect CD/DVD Install Sources
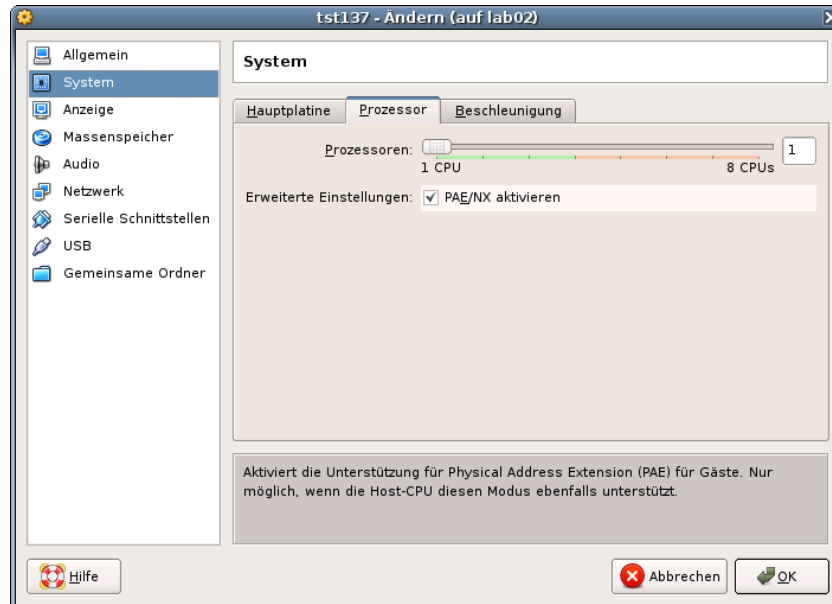
8. Set PAE for virtual CPU.

Figure 8: VirtualBox VCPU - PAE

5   Installation of the GuestOS - MeeGo

1. The start of the VMs of QEMU/KVM and VirtualBox vary slightly, even tough the following native procedures within the GuestOS are identical.

(a) Start QEMU/KVM
The start facilities of the plugin QEMU offer several options. Here the manual local start of the wrapper script is choosen. The first start of MeeGo is proceeded with the SDL console, this has some advantages for the later required 'quick-pressing' of the ESC key for the display of the boot menue. The option **--instmode** sets the bootdevice, here a preconfigured CD/DVD-image for boot.

```
./tst138.sh --console=sdl  --print --instmode
```

An alternate call for the start of the remote installation is:

```
ctys -t qemu \
 -a create=l:tst138,id:${TST138}/tst138.ctys,instmode,console:sdl\
 app2
```

This starts the same by transforming to the target host 'app2' and calling the previous wrapper script.

(b) Start VirtualBox
The start of the VM could be proceeded either by calling VirtualBox, or by the VBOX plugin. But both require in current version the pre-configuration of the appropriate install procedure. Either by mounted install media like a CD/DVD-image, or by usage of PXE for networl based installation. The folloing call starts the VirtualBox console.

```
VirtualBox
```

The following call call for the starts the remote VM with a VirtualBox console:

```
ctys -t vbox \
 -a create=l:tst137,id:${TST137}/tst138.ctys,console:vbox\
 app2
```

2. Now boot the VM and choose 'Installation Only'.
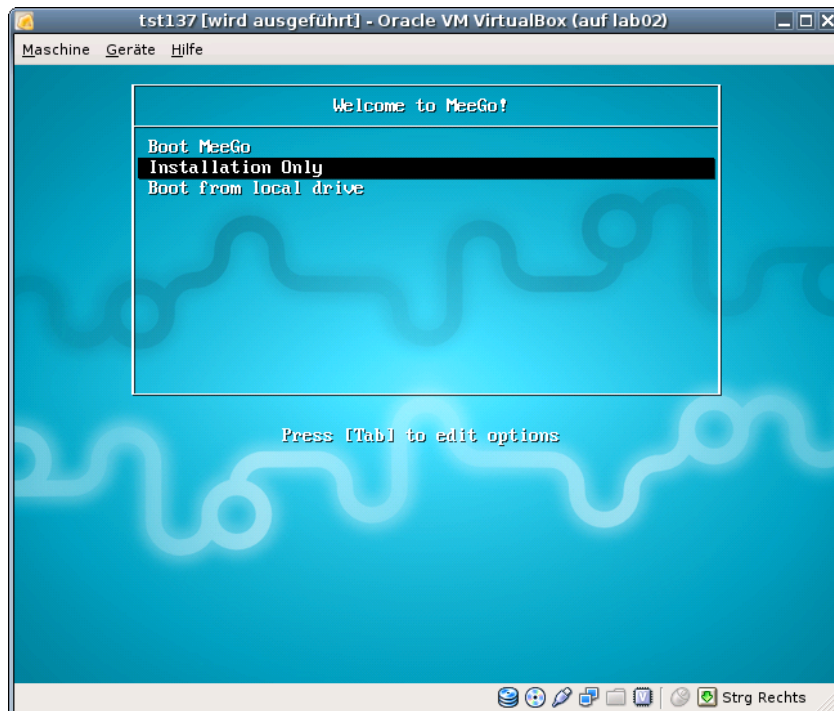

Figure 9: Install Menue

After some seconds the MeeGo screen occurs. The install procedure is quite similar to the RHEL based distributions.
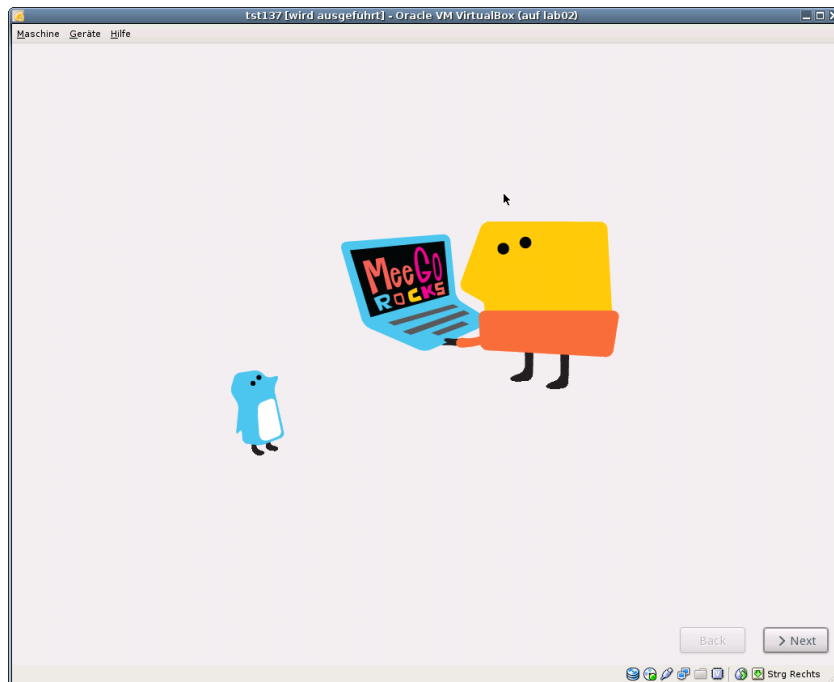

Figure 10: MeeGo Screen

3. When the HARDDISK error is displayed just press init again. In this description the default is choosed.
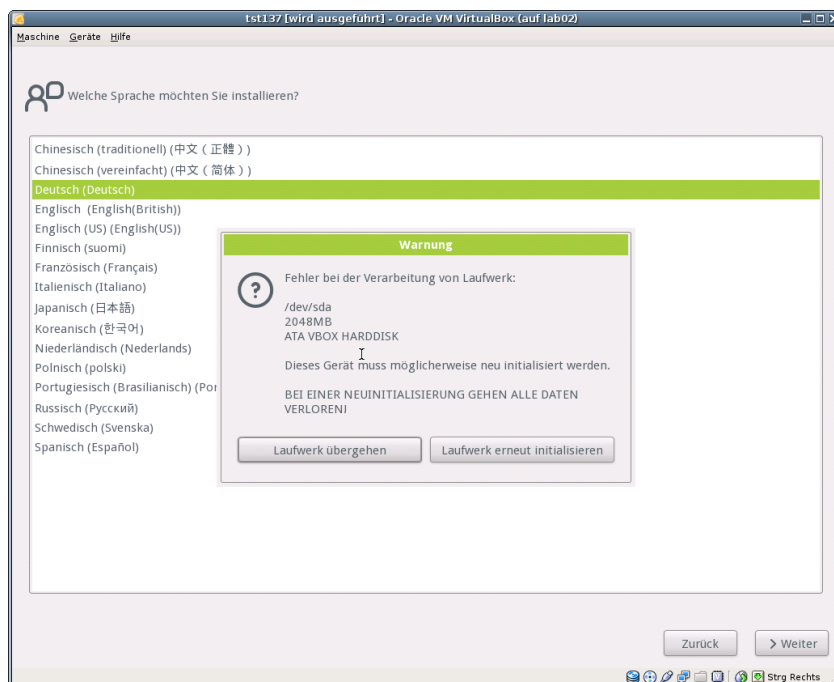

Figure 11: HDD-Init

4. Once the installation is complete, unmount the CD/DVD image and reboot.

   (a) QEMU/KVM
   In order to reboot just shutdown and boot again without the 'instmode' option. The shutdown could be proceeded by the 'quit' command within the monitor. The **monitor mode** is entered e.g. by **Ctrl-Alt-2**. One possible call is:

   ```
   ctys -t qemu \
       -a create=l:tst138,id:${PWD}/tst138.ctys,console:sdl \
       app2
   ```

   (b) VirtualBox
   Simply reboot without mounted install media. In case of a fresh start the call could look like:

   ```
   ctys -t vbox \
       -a create=l:tst137,id:${PWD}/tst137.ctys,console:vbox \
       app2
   ```

   The default console is here RDP.

5. Press ESC once immediately when the display mode first changes, the boot menue should now occur. If this fails just repeat it. Once the boot menue is visible press TAB and edit the boot parameters. Remove the keyword 'quiet' and append 'init 3'.

   Now MeeGo should boot and the console login should occur. The default password for the root account is 'meego'.
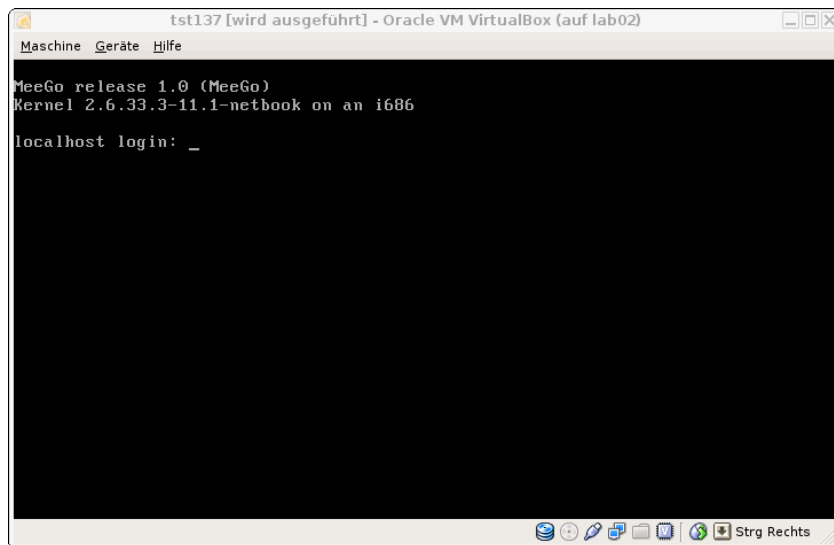

Figure 12: MeeGo ASC-II Console

- Set in the inittab the default boot level to 3. Edit '/etc/resolv.conf' and set your nameserver.

- Edit '/boot/extlinux/extlinux.conf' and change:
  - Remove 'quiet'
  - Comment 'menu hidden'

– Comment 'menu auto...'

In level 3 install the patched library 'libglx.so' by replacing
'/usr/lib/xorg/modules/extensions/libglx.so'. And change the mode to
'u+x,g+x,o+x'. Change mode for '/usr/bin/Xorg' to '+s'.

The required patch and/or library is available from
'http://202.112.3.1/libglx.so'.

REMARK: This is not the author's link, download is on your own responsibility. Anyhow, the
personal test worked in a test-environment and seems to be OK.

6. Reboot and start **twm** by calling **startx** from the ASC-II console.
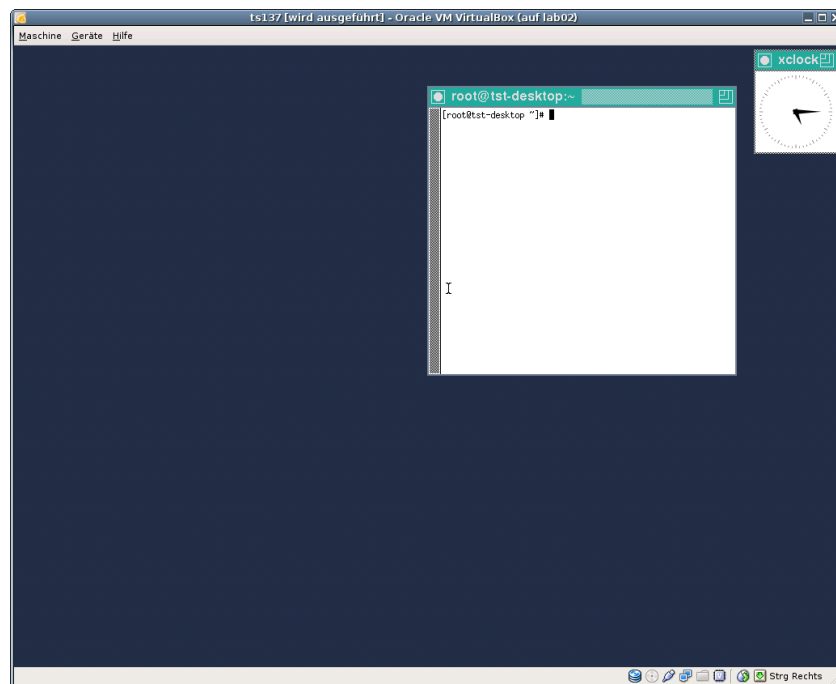

Figure 13: MeeGo X11 twm

7. Call 'firstboot' from within an xterm, and set basic configurations, particularly your keyboard.

8. **For VirtualBox only:**
   Install the **VBoxGuestAdditions** and patch the **/etc/init.d/vboxadd-service** by extending

   ```
   if [ -f /etc/redhat-release ]; then
   ```

   to

   ```
   if [ -f /etc/redhat-release -o -f /etc/meego-release ]; then
   ```

9. Reboot. Either set init level to 5, or call from command line 'init 5'.

Figure 14: Welcome MeeGo on VirtualBox

Figure 15: Welcome MeeGo on QEMU/KVM

Anyhow, for me the instalation currently does not work stable with the original 'moblin-dm'. VirtualBox installation works 'sometimes'(???), Qemu doesn't work at all. But the twm based X11 desktop works perfectly, so basically some drivers must be in place. So I am going to solve this later, and additionally installing than

the SDK packages too.

The target for now is to show the integration, therefore the current state is fine.

6 Creation of the Inventory - cacheDB

In case of a common mounted NFS filesystem for the pool VMs for simplicity just change into the directory of the VM on any machine. Call for the first check ctys-vdbgen(1) with the **–stdio** option for display only.

```
ctys-vdbgen --append --base=$PWD --stdio -- lab02
```

When the result is displyed correctly just call

```
ctys-vdbgen --append --base=$PWD -- lab02
```

The following output should be displayed:

```
Prepare execution-call:

Require DB-PATH,         USE: DEFAULT_DBPATHLST="/homen/acue/.ctys/db/default"
Require DB-PATH,         USE: -o => "/homen/acue/.ctys/db/default"
APPEND mode                 : ON(1)
STDIO mode off              : OFF(0)
Set TYPE scope          ADD: DEFAULT="-t ALL"
Preload TYPE set        ADD: DEFAULT="-T ALL"
For splitted operations ADD: DEFAULT="-b sync,seq "
Nameservice cache       OFF: DEFAULT="-c off "
Data cache              OFF: DEFAULT="-C off "

Resulting ENUMERATE     ADD: DEFAULT="-a enumerate=...
    ...matchvstat:active%disabled%empty,machine,\
    b:/mntn/vmpool/vmpool05/vbox/test/tst-ctys/tst137 \
    -C off  -c off  -T ALL  "

-> generate DB(may take a while)...
-----------------------------------
START:08:38:35
------


------
END:08:39:03
DURATION:00:00:28
-----------------------------------
RET=0
-----------------------------------

Cached data:

  Mode:                   APPEND
  Pre-Appended:           834 records
  Appended:               1 records
  Fetched Records Raw:      records
  Fetched Records Unique:   records
  Final:                  835 records


-----------------------------------
    ...finished.
```

This shows that only one entry is appended to the existing database with 834 VM-Entries. Now check the database entry by calling:

`ctys-vhost tst137`

The following result should be displayed:

```
label |stype|accel|distro|distrorel|os   |osrel|PM   |if |TCP
------+-----+-----+------+---------+-----+-----+-----+---+------------
tst137|VBOX |     |MeeGo |1.0.0    |Linux|2.6  |lab02|0  |172.20.2.241
```

7   Graphical Start of the Virtual Machine

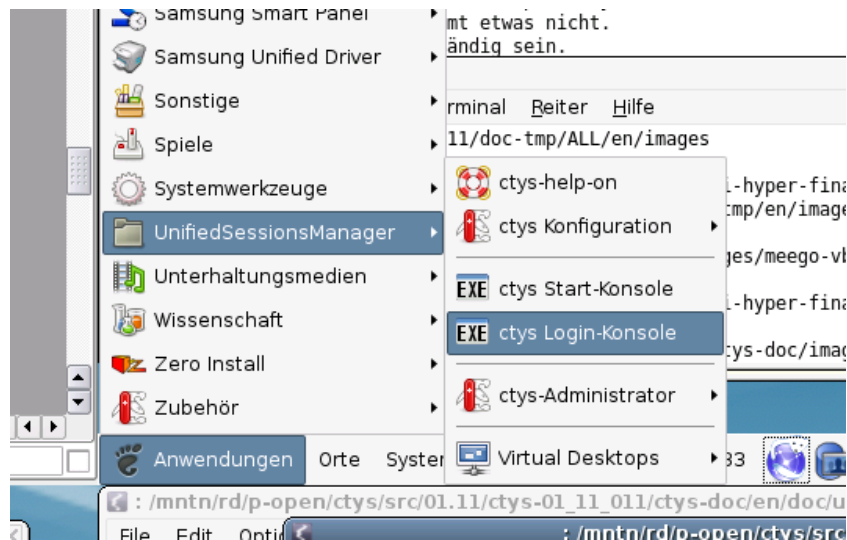Now call the menue item for start of the VM 'tst137'.

Figure 16: MeeGo Start Menue

The created cacheDB record for thr VM 'tst137' is now automatically visible in the list of startable virtual machines.
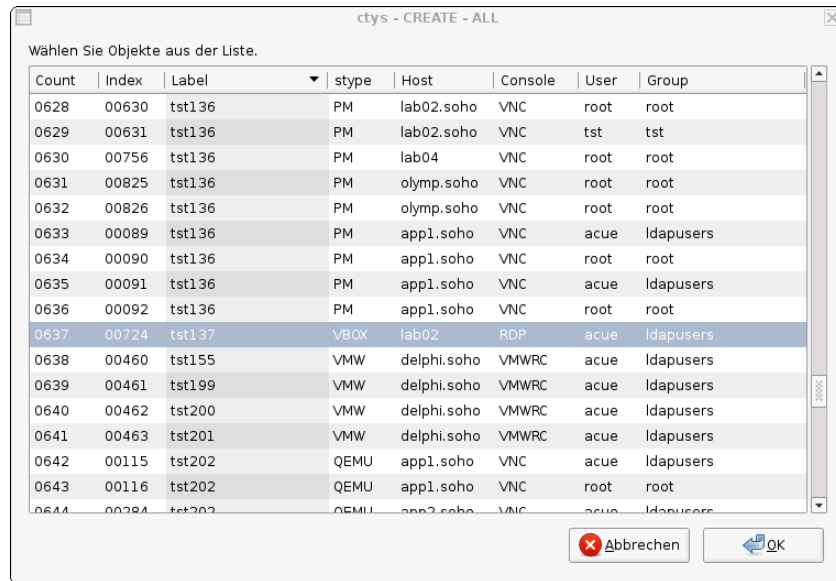
Figure 17: MeeGo VM Selection
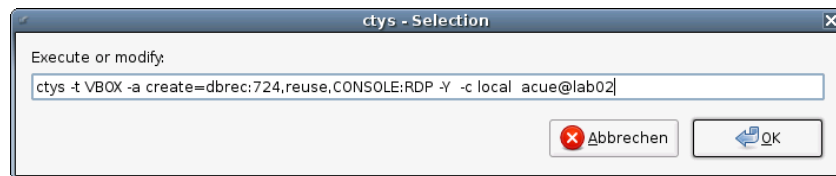
Confirm the selected entry.

Figure 18: MeeGo Call Confirmation

8 Manage the VM

8.1 Prepare MeeGo

1. Set yum repository in '/etc/yum.repo.d/'

2. Install the following additional Packages:

   (a) openssh-server
   (b) make
   (c) gcc
   (d) kernel-devel
   (e) kernel-netbook-devel

8.2 Install UnifiedSessionsManager in GuestOS - MeeGo

Apply standard procedure:

```
ctys-distribute -F 2 -P UserHomeCopy root@tst137
```

8.3   Open a Remote Shell

Call CLI plugin:

```
ctys -t cli -a create=l:tst137 root@tst137
```

8.4   Check Plugins States

Call ctys-plugins:

```
ctys-plugins -T all -E
```

8.5   Open a Remote X11-Terminal

ffs.

8.6   Open a Remote VNC-Desktop

ffs.

9 SEE ALSO

ctys(1) , *ctys-CLI(1)* , *ctys-configuration-VBOX(7)* , *ctys-createConfVM(1)* , *ctys-distribute(1)* , *ctys-groups(1)* , *ctys-macros(1)* , *ctys-plugins(1)* , *ctys-uc-VBOX(7)* , *ctys-vdbgen(1)* , *ctys-vhost(1)* , *ctys-VNC(1)* , *ctys-X11(1)*
For System Tools:
MeeGo(TM): [http://meego.com]

10 AUTHOR

Maintenance:	<acue_sf1@users.sourceforge.net>
Homepage:	<http://www.UnifiedSessionsManager.org>
Sourceforge.net:	<http://sourceforge.net/projects/ctys>
Berlios.de:	<http://ctys.berlios.de>
Commercial:	<http://www.i4p.com>

11 COPYRIGHT

Copyright (C) 2008, 2009, 2010, 2011 Ingenieurbuero Arno-Can Uestuensoez
For BASE package following licenses apply,

- for software see GPL3 for license conditions,

- for documents see GFDL-1.3 with invariant sections for license conditions,

This document is part of the **DOC package**,

- for documents and contents from DOC package see

  'Creative-Common-Licence-3.0 - Attrib: Non-Commercial, Non-Deriv'

  with optional extensions for license conditions.

For additional information refer to enclosed Releasenotes and License files.