Diese Anleitung bezieht sich auf Virtualbox, kann man aber vom Prinzip her auch auf VMware anwenden. Die Original URL diese Dokumentes findet sich hier im Anschluss. Diese Anleitung beschreibt das Vorgehen sehr genau, wenngleich man etwas um die Ecke denken muss. In Oberon kann man Befehle samt Parameter direkt in den Text schreiben.

Bei Review step 2 für die Änderung des Partitionstypes muss man im Test die Parameter händisch ändern: Partitions.ChangeType IDE0#01 11 76 ~

Ansonsten hält man sich genau an die Anleitung, dann funktioniert auch die Installation. In der VM stellt man 8MB Arbeitsspeicher, 1GB virtuelle Platte und das Diskettenlaufwerk ein. Wenn man Oberon nativ auf einem alten Rechner installieren will, muss man die maximale Größe der eingebauten Festplatte beachten. Diese darf nicht größer sein als 8,4GB. Das hängt grob mit dem Int13 zusammen. Sonst lassen sich keine Partitionen auswählen.

Have a lot of fun.

Linux User Group Nürnberg,

01.03.2024

Alle Angaben ohne Gewähr.

R. S. Doiel

https://rsdoiel.github.io/blog/2021/03/17/NativeOberon-VirtualBox.html

ETH Oberon System 3 on VirtualBox 6.1

By R. S. Doiel, 2021-03-17

In this post I am walking through installing Native Oberon 2.3.7 (aka ETH Oberon System 3) on a virtual machine running under VirtualBox 6.1. It is a follow up to my 2019 post FreeDOS to Oberon System 3. To facilitate the install I will first prepare my virtual machine as a FreeDOS 1.2 box. This simplifies getting the virtual machines' hard disk partitioned and formatted correctly. When Native Oberon was released back in 1990's most Intel flavored machines shipped with some sort Microsoft OS on them. I believe that is why the tools and instructions for Native Oberon assume you're installing over or along side a DOS partition.

Building our machine

Requirements

1. Install VirtualBox 6.1 installed on your host computer.
2. Download and install a minimal FreeDOS 1.2 as a virtual machine
3. Downloaded a copy of Native Oberon 2.3.7 alpha from SourceForge
4. Familiarized yourself Oberon's Text User Interface
5. Boot your FreeDOS virtual machine using the Oberon0.Dsk downloaded as part of NativeOberon_2.3.7.tar.gz
6. Mount "Oberon0.Dsk" and start installing Native Oberon

Before you boot "Oberon0.Dsk" on your virtual machine make sure you've looked at some online Oberon documentation. This is important. Oberon is very different from macOS, Windows, Linux, DOS, CP/M or Unix. It is easy to read the instructions and miss important details like how you use the three button mouse, particularly the selections and execute actions of text instructions.

Virtual Machine Setup

VirtualBox 6.1 can be obtained from virtualbox.org. This involves downloading the installer for your particular host operating system (e.g. Linux, macOS or Windows) and follow the instructions on the VirtualBox website to complete the installation.

Once VirtualBox is installed, launch VirtualBox.

Click the "New" button and name your machine (e.g. "Native Oberon 2.3.7 Alpha") and choose type of "Other" and version "DOS". Click continue. I accepted the default memory size of 32 MB. This is plenty for Oberon. I clicked on create disk and accepted the default VDI (VirtualBox Disk Image). Press continue. I think accepted "Dynamically allocated", press continue. I chose a "disk size" of 100.00 MB. Oberon System is tiny. Press create button.

Make sure your new machine is highlight on the left side of the VirtualBox management panel. Click on Settings button, it looks like a gear towards the top. Click "Display" on the model dialog and bump the Video Memory up to 128 MB. I also clicked Enable 3D Acceleration (though I don't think Oberon uses this). Before clicking OK click on the Network icon in the modal dialog box. Change "NAT" to "Bridged Adapter". Now click "OK" to close the modal dialog box.

Your VirtualBox is now ready, before pressing “Start” we need to install FreeDOS 1.2.

Make a FreeDOS 1.2 machine

Download the FD12CD.iso file from the FreeDOS project.

“Insert” the “FD12CD.ISO” file into our VirtualBox 6.1 CD-ROM drive. Go to the VirtualBox management panel. In the area that summarizes storage click the word “Empty” in the line with “[Optical Drive]”. Find the “FD12CD.ISO” you downloaded and select it.

Now press the green “Start” arrow in the VirtualBox management panel. This should start your virtual machine and it will boot using the CD-ROM drive.

This will display a welcome screen with installation options. Press your “tab” key once. This should cause a boot string to be displayed. Type a space and then type the word “raw” (without quotes). Press enter. Next select the language you want to install with (e.g. English). Choose “Yes - Continue with installation” on the next prompt. You should then be given a dialog box that indicates “Drive C: does not appear to be partitioned.”, select “Yes - Partition drive C:”. Then that completes press “Yes - Please reboot now”.

This will cause the machine to reboot and you will be faced with the “Welcome to FreeDOS 1.2” screen once again. Press the “tab” Add a space and type “raw” to the boot string as before. Select the language again then select “Yes - Continue with installation”. The screen should now say something like “Drive C: does not appear to be formatted”, select “Yes - Please erase and format drive C:”.

When done it’ll gather some info on the system and ask you which keyboard you’re using. Pick yours (e.g. “US English (Default)”). It will then give you a choice of what to install. Since we’re going to overwrite this when we install Oberon just select the base package, then select “Yes - Please install FreeDOS 1.2”

Before selecting “Yes - Please reboot now” when the install is finished you want to “eject” your FD12CD.ISO from the virtual CD-ROM drive. Switch back to your VirtualBox management panel. Click the text that says “FD12CD.iso” and select “remove disk from virtual drive” in the popup menu. Switch back to your Virtual machine and select “Yes - Please reboot now”

If all goes well the machine will boot into FreeDOS 1.2. When you see the “C:>” prompt type “shutdown” (without the quotes) and press enter. We’re now ready to start installing Native Oberon 2.3.7.

Native Oberon 2.3.7

Native Oberon used to be hosted at ETH where Oberon and the Oberon System was first developed as a research and instructional project. Unfortunately this seems to no longer be supported by ETH. Prof. Wirth has long been retired now and they no longer choose to use such a useful language or Operating System.

SourceForge has a mirror of the original sources and some of the remaining community has put at a “new” release of 2.3.7 Alpha bringing Native Oberon a little closer to the present. It’s this version we’ll use. You can read more at the SourceForge as well as at the Oberon Wikibook. ETH also still maintains an email list for Oberon and it is active. It can be found at

https://lists.inf.ethz.ch/mailman/listinfo/oberon. I recommend browsing the archives of the Email list if you run into problems. I've found very helpful information there and the people on the list seem happy to answer a novices question.

We are going to be downloading files from the Native Oberon Project's Files page at SourceForge.

> https://sourceforge.net/projects/nativeoberon/files/nativeoberon/

In the Files page download the instructions NativeOberonInstall.pdf or or the text version. This document by Pieter Muller (May 1999) explains the installation process. It is good for its overview though I found the actual process simpler than what was described for May 1999.

On the Files page you'll also see a green button to "Download Latest Version", NativeOberon_2.3.7.tar.gz. Click the button and download it.

The NativeOberon_2.3.7.tar.gz contains the files we'll need to run NativeOberon on our VirtualBox. Ungzip and untar the file into a location that is convenient for you. I put mine in `src/NativeOberon-2.3.7` and I had downloaded the file into my home directory's "Downloads" folder.

```
    mkdir -p src/NativeOberon-2.3.7    cd src/NativeObeorn-2.3.7    tar zxvf
~/Downloads/NativeOberon_2.3.7.tar.gz
```

You now have the software ready to proceed in installing the system in VirtualBox.

NativeOberon in a VirtualBox

Go back to your VirtualBox management panel. We need to place he boot disk image in the virtual floppy drive. In the files we unpacked (i.e. ungzip and untar) there is a file named "Oberon0.Dsk". We want to mount that in the virtual floppy drive. Click on the word "Empty" next to "Floppy Device 0:" in the management panel. You are then given a modal dialog box and we want to select "Choose a disk file". You can then find the files you save and select "Oberon0.Dsk".

Booting with Oberon0.Dsk

We can now click "Start" button at the top of the VirtualBox management panel. This will boot the virtual machine using "Oberon0.Dsk". Oberon itself loads completely into memory.

You now have a running Oberon System but we need to install it on the virtual hard drive. Fortunately our running system comes with built in instructions. It is here that people how haven't used Oberon before are going to run into trouble.

Oberon System uses all three buttons of a three button mouse. On most mice I've encountered to day there are two buttons and a scroll wheel. The scroll wheel is click able and functions like the middle button on an Oberon mouse.

The left mount button sets the pointer, the middle button (our scroll wheel if your mouse is like mine) is used to execute commands and the right mouse button is used to select text. In our installation instructions displayed on our virtual machine we generally be middle clicking the blue colored text.

In Native Oberon all text is potentially actionable. Unlike in Unix where you type a command press enter then have to retype (or use the command history) to execute the next command we're going to click on the text and sometimes select text to execute commands. Before we proceed I highly recommend readying and trying a tutorial out before attempting to install Oberon on your virtual hard drive. There is an Oberon System 3 - Main Tutorial available at the Internet Archive's Wayback machine.

Installing to our Virtual Hard disk

Along the right side (in the system track) is the text "Edit.Open Introduction.Text". Click with your middle button (scroll wheel on my mouse) and this will open the text in the "Edit" track on the left side. Read this text if you haven't before. Scrolling through the text is a little different than the scroll bars on macOS, Windows, X Windows. They are on the left side and the middle mouse button sets the scroll position. The left button pages down, the right pages up. You can close the "Introduction.Text" windows by middle clicking "System.Close" in the upper menu bar.

Review step 1.

We need to configure the hard drive by middle clicking on "Config.Desk Standard ATA/EIDE"[1]. In the console viewer above you should see something like

```
Disk: Standard ATA/EIDEStatic BootLinker for OM Object Files / prk linking
Native.Bin 255388
```

Review step 2.

Middle click on "Edit.Open InstallFiles.Tool". A "Tool" file is like a text file but usually contains instructions and a sorta menu or recipe of commands. In fact our instructions in the lower viewer of the system track is a "Tool" file called "Install.Tool". Using "Edit.Open" to open the tool or text file opens a viewer on the left track, the edit track. If you had clicked on "System.Open Install.Tool" it would open a viewer on the right, or systems track. In either track by default the viewers will tile (not overlap). If you want to close a view you can click on "System.Close" in the viewer's menu bar. Now open our InstallFiles.Tool in the edit track.

InstallFiles.Tool

We now are going to prepare our virtual hard drive. Like our "Install.Tool" text we have a series of instructions which commands we can click on (the ones in blue).

Middle click on "Partitions.Show ~". This will open a pain showing the partition information. You should see something like

```
Disk: Diskette0, GetSize: no media, removableDisk: IDE0, 99MB, VBOX
HARDDISKIDE0#00        99MB  --- (Whole disk)IDE0#01        99MB  ---
(Free)Disk: IDE2, GetSize: no media, removable, VBOX CD-ROM
```

This tells us we have three drives in our VirtualBox visible to Oberon. The first is the floppy drive. It shows "no media". That might seem odd but when you read the "Oberon0.Dsk" it read that into memory and the whole OS is not running in memory, not from disk! While the disk is still "in the

drive" as far as VirtualBox is concerned it isn't "mounted" from the point of view of the operating system.

The second disk section describes our virtual hard drive. The third describes the virtual CD-ROM drive.

We're interested in using the disk "IDE0" with the device number of "01", we express that as "IDE0#01".

In the "Partitions.Text" viewer where we see the partitions information we can type the command described in "InstallFiles.Tool"

```
Partitions.ChangeType IDE0#01 6 76 ~
```

We then middle mouse click on line we just type. This should produce output in the "System.Log" view in the upper right of the screen that looks like

```
IDE0#01 changed to type 76
```

I had to do a modified version of step 3 of "InstallFiles.Tool" choosing option "b".

In the output of "Partitions.show ~" (i.e. the Partitions.Text viewer) want to middle click on the "I" of "IDE0#01". Then right mouse button select "IDE0#01".

From 3a in the "InstallFiles.Tool" viewer middle mouse button click on "Partitions.Format ↑". This should result in the "System.Log" viewer showing

```
IDE0#01 format successful
```

After formatting the drive I was able to complete step 3b by middle clicking the commands as provided in the "InstallFiles.Tool"

> NOTE: you may need to scroll that window to see all of step 3

Middle click on "Partitions.UpdateBootFile ↑" The "Systems.Log" viewer should show

```
IDE#00 update successful
```

Middle click on "FilesSystem.Mount DST AosFS ↑" in 3b. The "System.Log" viewer should show

```
DST: mounted
```

We are ready for Step 4. This command does the brunt of the work of coping things over. The command "Configuration.DoCommands" take a list of Oberon commands and executes them one after the other. Middle Mouse click on "Configuration.DoCommands". The "System.Log" viewer will show many messages that are a result of each command taken. Make sure there are no errors. The last series of commands renamed files so you should see something like

```
System.RenameFilesDST:Rel.Obeorn.Text => DST:Oberon.Text
renamingDST:Rel.Network.Tool => DST:Network.Tool renamingDST:Rel.DOS.Tool =>
DST:DOS.Tool renaming
```

For step 5 of “InstallFiles.Tool” we can close our “InstallFiles.Tool” viewer by middle clicking on “System.Close” in the menu bar. You can also close the “Partitions.Text” viewer using its menu bar and middle clicking “System.Close”.

Right now we’ve formatted our hard drive and copied a bunch of files too it. We still need to configuration our system before it is self hosting.

In the “Install.Tool” viewer we want to open our “Configure.Tool”. Middle click on the “Edit.Open Configure.Tool”.

Configuring our Oberon System

The configuration tool breaks configuration into a series of parts. First part is configure the display in Part two we make the hard disk bootable.

To configure out display we want to middle click on the blue text in “Config.DetectVesa (BIOS cal might hang some systems!)”. You will be presented with a list of screen resolutions. I middle clicked the blue text in “Config.Vesa 00000147H 1600 * 1200 * 32”. In the “System.Log” viewer this showed

```
Vesa mode 00000147H
```

> NOTE: You will likely need to scroll down the page using the scroll bar

In part two we need to decide how we want to boot Oberon. In our case I recommend Option 2, boot Oberon directly (non-FAT hosted). Middle click the blue text “Config.BootParition menu ~”. The “System.Log” viewer should output

```
IDE0#01 config written
```

Middle click the blue text “Partitions.Show ~”. Like in “InstallFiles.Tool” this will open a new “Partitions.Text” viewer with content like

```
Disk: Diskette0, GetSize: no media, removableDisk: IDE0, 99MB, VBOX
HARDDISKIDE0#00       99MB  --- (Whole disk)IDE0#01       99MB   76 * Native
Oberon, AosDisk: IDE2, GetSize: no media, removable, VBOX CD-ROM
```

Using your right mouse button select “IDE0#01” then in the “Configure.Tool” viewer middle click the blue text “Partitions.Activate ↑”. The “System.Log” viewer should show

```
IDE0#01 already active
```

We don’t have a partition to deactivate so we can skip the last step of option 2. This is a good place to “eject” our floppy disk “Oberon0.Dsk” before we “System.Reboot”.

To eject the disk click on “Oberon0.Dsk” in the VirtualBox manager panel. The should then change the text to “Empty”.

Finally we’re ready to move to the last step in “Configure.Tool”. Scroll down and find “System.Reboot”. Middle click on the blue text “System.Reboot”. At this point the virtualbox should reboot from the virtual hard drive. This reboot will take a little longer than the floppy boot

and the screen size of the virtualbox will be large based on the settings you picked early. You have a minimal working Oberon System 3. Now to install some more programs and flesh the system out.

Install some programs

First we need to get the zip files provided in NativeOberon_2.3.7.tar.gz on to the hard drive. Historically these were done via 1.44 MB floppy disks. We're going to make it easier. Native Oberon 2.3.7 can read an ISO formatted CD-ROM.

Making our virtual CD-ROM

Under Ubuntu 20.04.2 LTS machine creating a ISO image is one command. Below is I am going to create an ISO image of the directory "NativeOberon-2.3.7" and save the image as "NativeOberon-2.3.7.iso".

```
    mkisofs -J --iso-level 3 \        -o NativeOberon-2.3.7.iso
NativeOberon-2.3.7
```

The `-J` says to use the Joliet extensions, the `--iso-level` sets the level of ISO support, in this case to 3. See the manpage for `mkisofs` for details.

On macOS this involves two commands. First use the "Disk Utility" to create an image of the folder where you unpacked NativeOberon_2.3.7.tar.gz. This will result in a ".dmg" or disk image file common on macOS. Next we need to convert this to an ISO formatted image file. For that we use a command line macOS tool called `hdiutil` to convert the disk image to an ISO format. In the process you will create the ISO file but it will have the extension of ".cdr". You can rename (i.e. mv) that file so it has a ".iso" extension. This is suitable to mount in VirtualBox's virtual CD-ROM drive.

```
    hdiutil convert NativeOberon-2.3.7.dmg -format UDTO -o
NativeOberon-2.3.7.iso    mv NativeOberon-2.3.7.iso.cdr NativeOberon-2.3.7.iso
```

Go to the VirtualBox 6.1 management panel and mount the ISO image file you created. Now we're ready to return to our Native Oberon virtual machine.

Installing from an ISO CD-ROM

I suggest create the following as it's own tool text. But if you want you can also type in the commands and execute one by one.

```
These are the instructions from installing the NativeOberon 2.3.7 zip archive
files. Steps:1. See what the CD-ROM mount point by reviewing the partitions
Paritions.Show ~On my virtual machine the second disk is IDE2 and thatis where
we'll find the CD-ROM.2. Mount the ISO image as CD    FileSystem.Mount CD ISOFS
IDE2 ~3. Check to see what files are on the CD-ROM    System.Directory CD:* ~4.
Copy the files from the CD-ROM to the harddisk    System.CopyFiles
CD:Apps1.zip => Apps1.zip        CD:Apps2.zip => Apps2.zip        CD:Docu.zip
=> Docu.zip        CD:Gadgets.zip => Gadgets.zip        CD:Pr3Fonts.zip =>
Pr3Fonts.zip        CD:Source1.zip => Source1.zip        CD:Source2.zip =>
Source2.zip        CD:Source3.zip => Source3.zip        ~5. Unzip all our
archives using the ZipTool.    ZipTool.ExtractAll \o \p SYS:        Gadgets.zip
Docu.zip Apps1.zip Apps2.zip        Pr3Fonts.zip Pr6Fonts.zip
Source1.zip Source2.zip Source3.zip        ~
```

You should now have a full installed Native Oberon 2.3.7 system running under VirtualBox 6.1. Enjoy your explorations.

---

1. This blue text makes it clear the command is actionable, like a link in the web browser. But the actual text is the command not the color.↩️

RSS feeds and website built with pttk, Bash, Make and Pandoc.