

Linux auf dem Lenovo Yoga 300 11 IBR

Dieter Drewanz

Dokument begonnen: September 2016

Dokument zuletzt geändert: Februar 2017

Kurzfassung:

Das Netbook wurde vor der Installation des Betriebssystems Linux für Dual-Boot zunächst untersucht. Insbesondere traf dies auf die Belegung der Festplatte ab Werk durch den Hersteller und Windows zu. Danach wurde die Belegung der Festplatte für die Aufnahme des Linux Betriebssystems Linux vorbereitet und dieses installiert.

Die weiteren Teile beinhalten die Konfiguration, Installation von weiteren nützlichen Anwendungen und deren Verwendung. Die Verwendung der Shell steht dabei oft im Vordergrund.

Im Prinzip wurde hier fast alles dokumentiert, ins Netz gestellt für eigene Zwecke zum Nachschlagen bei Bedarf und vielleicht hilft es auch dem Einen oder Anderen weiter.

Inhaltsverzeichnis

1 Einleitung.....	2
2 Vorbereitungen zur Installation.....	3
2.1 Der erste Start des Netbooks mit vorinstalliertem Betriebssystem.....	3
2.2 Startversuche mit einem Live-Linux vom USB-Stick.....	4
2.2.1 Vorbereitung des Netbooks für einen Live-Linux-USB-Stick Startvorgang.....	4
2.3 Starten eines Live-Linux vom USB-Stick.....	5
2.3.1 Erstellen eines bootfähigen Live-Linux-USB-Sticks.....	5
2.3.2 Untersuchung des Inhaltes der eingebauten Festplatte im Netbook.....	9
2.3.3 Sicherung der Partitionen.....	10
3 Unterstützte Hardware des Lenovo Yoga 300 unter Linux.....	13
3.1 Bei Verwendung von xubuntu 16.04.....	13
3.2 Bei Verwendung von xubuntu 17.04.....	14
3.3 Bei Verwendung von opensuse tumbleweed.....	14
3.4 Weitere interessante Linux-Distributionen.....	14

4 Installation eine Linux-Betriebssystems als Dual-Boot Betrieb.....	15
4.1 Vorbereitungen für eine Dual-Boot Installation.....	15
4.2 Änderung der Partionierung.....	15
4.2.1 Geplante Partitionierung.....	15
4.2.2 Sicherung der Dateien auf dem Rechnern.....	16
4.2.3 Verkleinern der Windows-Partition.....	16
4.2.4 Nachgeholte Sicherung der Daten auf der Festplatte.....	17
4.2.5 Anlegen von zusätzlichen Partitionen für Linux.....	20
4.3 Installation eines Linux-Betriebssystems auf dem Netbook.....	21
4.3.1 Installation von Ubuntu 16.04.....	21

1 Einleitung

Der hier geschriebene Text wurde parallel als kleine Dokumentation angelegt. Eine Nebenabsicht war auch um bei Bedarf schnell ein ähnliches System wieder aufzusetzen und nachzusehen, was gemacht wurde für spätere Änderungen oder Fortsetzungen.

Somit ist dieser sprachlich einfach gehalten und auch nicht weiter quergelesen. Ohne Absichten und Aussichten auf irgendwelche Zuwendungen wurde verständlicherweise hier auf den schriftlichen Ausdruck auch nicht besonders geachtet. Manchmal wurden auch nur die Notizen mit den Links zu Seiten im Internet mit interessanten Informationen, Beschreibungen und Anleitungen gesammelt.

Eingestellt im Netz kann es vielleicht dem Einen oder Anderen nützlich sein zur Information oder Fehlerbehebung.

Nachdem das ASUS-Netbook eeepc vom Sommer 2009 mittlerweile die Festplatte ab und zu laut klackerte und durch die häufige Benutzung auch schon arg verschlissen war, bereits ein zweites Akku in dem Gerät seinen Dienst leistete, die Netzteilbuchse einen leichten Wackelkontakt aufwies (noch 5 Monate durchhielt), wurde es vielleicht nun doch Zeit ein neues Netbook zu suchen und zu kaufen. Nachdem das Notebook mindestens drei USB-Anschlüsse haben sollte, reduzierte dies die Auswahl deutlich. Zuletzt fiel die Wahl auf ein gutes Angebot eines Händlers zum Leno Yoga 300.

Negativ an allen angebotenen Geräten war, dass diese keine leicht austauschbaren Akkus besitzen. Nach einigen Jahren wird der Austausch in der Regel richtig teuer. So wird in der Zukunft der Elektronikschrott sicherlich nicht weniger werden.

Auf dieses Gerät sollte natürlich wieder ein richtiges Betriebssystem installiert werden. Daher wurde auch jeweils im Internet recherchiert, ob bereits diese Geräte von Linux-Nutzern als tauglich eingestuft wurden. Die Rechner ohne Betriebssystem waren zwar durchwegs zu einem niedrigeren Preis angeboten, aber unter Berücksichtigung der geringeren Leistungen gemäß Datenblatt, blieb eigentlich nur noch ein marginaler Unterschied von vielleicht fünf bis zehn Euro übrig. Somit ergab sich fast zwangsweise ein Gerät für den Betrieb als Dual-Boot-Rechner (Windows mit Linux).

Technische Daten des Netbooks Lenovo Yoga 300 11BR mit Kommentaren:

1. Prozessor Intel N3060 (1,6G Dual Core bis 2,3G Boost). Die Virtualisierungsunterstützung

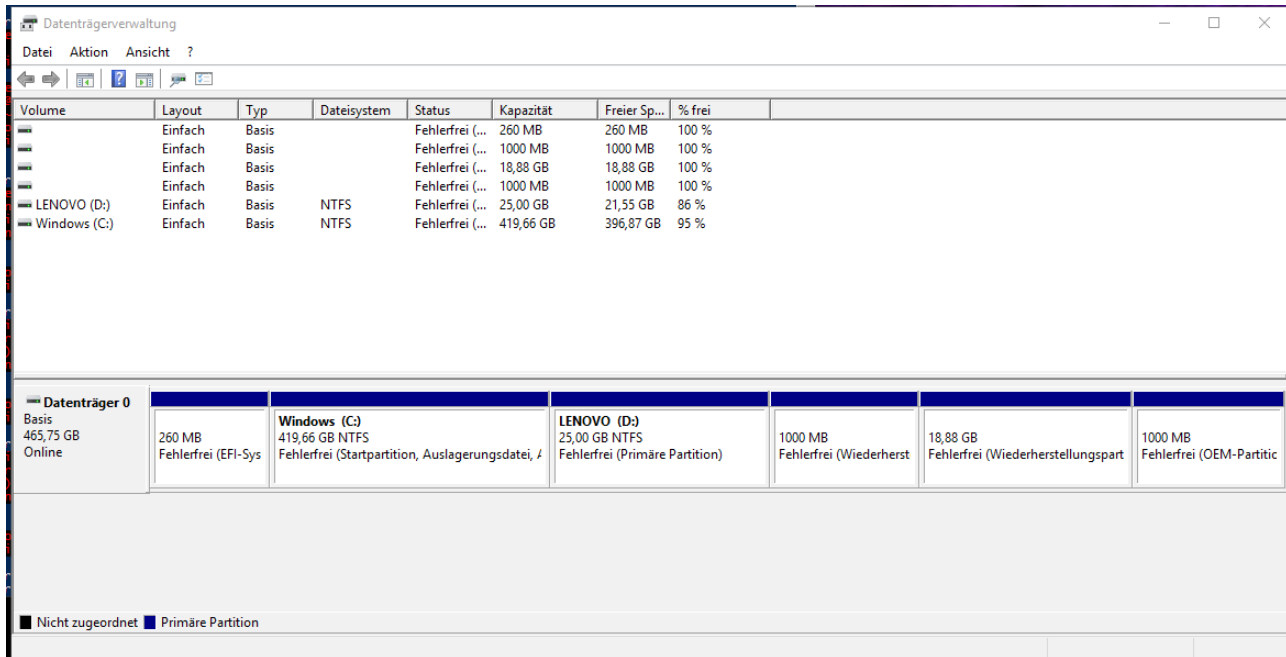
ließ sich im UEFI-BIOS aktivieren (Intel virtualization „enable“). Für Office, Internetnutzung und Spiele ohne große Anforderungen ist dies ausreichend. Für lange Akku-Laufzeit (8h) so weit ok; Wer häufiger Videobearbeitung machen möchte, wird Wert auf einen Quad-Core-Prozessor legen. Ein Netbook mit AMD-Prozessor war nicht erhältlich, wäre genau so gut gewesen.

2. Speicher 4GB RAM (nicht erweiterbar). Für die genannten Anwendungen ist dies ausreichend. Wenn virtuelle Maschinen genutzt werden sollen auf dem Netbook wird mehr Speicher benötigt. Als maximale Anforderung Xubuntu zu verwenden für Bürotätigkeiten würde es auch noch mit 2GB RAM funktionieren.
3. Festplatte 500GB (8GB Flash). Die Geräte mit 32GB haben zu wenig Speicherplatz. Für Windows 10 als einziges Betriebssystem ist dies zu wenig. Im Internet sind genügend Blogs zu finden, dass hier bereits schon häufig die standardmäßigen Updates zu Problemen führen. Meine Empfehlung wäre daher von Geräten unter 100GB die Finger lassen. Für Dual-Boot um beide Betriebssysteme gut verwenden zu können, sollten die Festplattenkapazität nicht unter 200GB liegen.
4. Bildschirm 11", 1320x768, Touchscreen; Platz für mindestens 800 Pixel-Zeilen statt 768 wäre von Seiten des Gehäuses kein Problem gewesen;
5. 3xUSB (2x 2.0 und 1x 3.0); Auf mindestens drei USB-Anschlüsse sollte geachtet werden. Wenn bereits zwei Anschlüsse für den USB-Live Stick und für die Maus belegt werden, dann gäbe es bei vielen anderen Geräten bereits keinen freien Platz mehr für einen weiteren USB-Stick um Daten auszutauschen. Wer schleppt schon gerne immer einen USB-Hub mit sich herum?

2 Vorbereitungen zur Installation

2.1 *Der erste Start des Netbooks mit vorinstalliertem Betriebssystem*

Das Gerät besaß eine 500GB Festplatte mit einem vorinstallierten Windows Betriebssystem. Als Boot-System wurde UEFI, bzw. UEFI-BIOS ermittelt. Das Betriebssystem Windows 10 wurde erstmals ohne Internetverbindungen gestartet und in der Windows Administrator Konsole mittels Eingabe von `diskmgmt.msc` die Partitionierung der Festplatte nachgesehen für die spätere Einrichtung von weiteren Betriebssystemen.



Volume	Layout	Typ	Dateisystem	Status	Kapazität	Freier Sp...	% frei
	Einfach	Basis		Fehlerfrei (...)	260 MB	260 MB	100 %
	Einfach	Basis		Fehlerfrei (...)	1000 MB	1000 MB	100 %
	Einfach	Basis		Fehlerfrei (...)	18,88 GB	18,88 GB	100 %
	Einfach	Basis		Fehlerfrei (...)	1000 MB	1000 MB	100 %
LENOVO (D:)	Einfach	Basis	NTFS	Fehlerfrei (...)	25,00 GB	21,55 GB	86 %
Windows (C:)	Einfach	Basis	NTFS	Fehlerfrei (...)	419,66 GB	396,87 GB	95 %

Datenträger 0						
Basis						
465,75 GB						
Online						
260 MB	Windows (C:)	LENOVO (D:)	1000 MB	18,88 GB	1000 MB	
Fehlerfrei (EFI-Sys)	Fehlerfrei (Startpartition, Auslagerungsdatei, /	Fehlerfrei (Primäre Partition)	Fehlerfrei (Wiederherst	Fehlerfrei (Wiederherstellungspart	Fehlerfrei (OEM-Partitic	

Es handelt sich dabei um keine traditionelle DOS-Partition mit Einträgen im MBR (Master-Boot-Record) sondern um GPT/GUID-Partitionen.

Literaturliste:

<https://de.wikipedia.org/wiki/Partitionstabelle>

<https://wiki.ubuntuusers.de/Partitionierung/Grundlagen/>

https://de.wikipedia.org/wiki/GUID_Partition_Table

2.2 Startversuche mit einem Live-Linux vom USB-Stick

2.2.1 Vorbereitung des Netbooks für einen Live-Linux-USB-Stick Startvorgang

Auf einschlägigen Seiten im Internet stand, dass nur Linux-Betriebssysteme mit 64bit auf Rechnern mit GPT und Windows 10 funktionieren würden. Als weitere Maßnahmen müssen die Schnellstarts im UEFI-BIOS und in den Betriebssystemeinstellungen von Windows abgestellt werden. Es wird empfohlen eine Linux-Distribution zu wählen, die „Secure Boot“ unterstützt, wie zum Beispiel Ubuntu (mit Grub 2) ab der Version 14.

Eine Suche im Internet ergab, dass das Menu des UEFI-BIOS beim Lenovo Yoga 300 11IBR über F12 beim Start (Umschalttaste für Keyboard+F12, bzw. Fn+Helligkeit/F12) oder über Betätigung des kleinen Tasters hinter einem kleinen Loch neben der Tastatur mit einer aufgebogenen Büroklammer erreicht würde. Diese ist nicht so spitz, wie zum Beispiel eine Nähnadel, und verschleißt somit den Taster an der Oberfläche nicht so stark, falls diese häufig benutzt werden sollte. Mit noch unveränderten Einstellungen des UEFI-BIOS klappte die erste Methode die UEFI-BIOS-Einstellungen zu erreichen fast gar nicht und es wurde daher die zweite Methode für die erstmaligen Änderungen an den UEFI-BIOS Einstellungen verwendet. Je nach Vorkonfiguration funktioniert der Zugang zum UEFI-BIOS über die Tastatur meist erst, wenn die Schnellstarts abgeschaltet wurden.

Trotz der Anforderung nur 64bit-Linux-Live-Distros zu verwenden, wurden einige Versuche mit verschiedenen älteren 32bit-Versionen durchgeführt, die vom USB-Stick gestartet wurden. Hierzu

mußte „secure boot“ deaktiviert („disabled“) und „BIOS Boot“ aktiviert („enabled“) werden. Beim Windows müssen auch alle Optionen zum Schnellstart abgeschaltet werden im UEFI-BIOS und auch in den Betriebssystemeinstellungen.

Alle getesteten älteren Linux-Versionen 32bit (Knoppix 6.4, Lubuntu 10.04, Xubuntu 10.04 und 16.04) versuchten vom Stick zu starten blieben aber an unterschiedlichen Stellen hängen. Das Knoppix bootete bis auf die Konsole, aber ein Einloggen war unmöglich, da nach dem Betätigen von Strg+Alt+F12 nach spätestens drei Sekunden der Bildschirm auf den leeren X11-Bildschirm umschaltete.

Die positiven Ergebnisse dieser Versuche waren, dass erstens alle notwendigen Einstellungen am Netbook zum Booten mit einem USB-Stick ausgeführt wurden und zweitens, welche USB-Sticks vom Netbook als bootfähig erkannt wurden. Auf einen dieser ausgetesteten USB-Sticks wurde anschließend ein 64bit Live-Linux aufgespielt.

Folgende Maßnahmen fallen an um Booten vom Linux-Live-USB-Stick und später vom installierten Linux zu ermöglichen:

1. Im UEFI-BIOS sind die Schnellstartoptionen zu deaktivieren. Hinweis: Inneren Taster über das kleine Loch neben Einschalter mit einem geeigneten Hilfsmittel drücken, falls über F12 beim Booten (bei anderen Modellen können es andere Tasten sein)
2. In den Windows-Optionen sind alle Schnellstarts zu deaktivieren.
3. Gegebenenfalls kann „Secure Boot“ abschalten und „BIOS-Boot“ einschalten notwendig sein. Empfohlen wird jedoch eine Distro zu wählen, die solche Änderungen nicht benötigt.
4. Über F12 erscheint nun ein Boot-Menü mit den Auswahlmöglichkeiten von welchem Medium gebootet werden soll, oder ob das Einstellungsmenü des BIOS (UEFI-BIOS) aufgerufen werden soll.

2.3 Starten eines Live-Linux vom USB-Stick

2.3.1 Erstellen eines bootfähigen Live-Linux-USB-Sticks

Unter den bootfähigen USB-Sticks wurde ein Mini-USB-Stick ausgewählt, der weniger als einen Zentimeter aus dem Gehäuse ragt, damit dieser nicht für jeden Transport abgezogen werden muss um nicht abzubrechen oder auch die Buchsen zu beschädigen.

Hingewiesen werden soll noch auf für eine nur für Mini-Stick besonderes Problem beim Schreibvorgang. Wenn auf dem Stick mit hoher Geschwindigkeit mehrere GByte an Daten in einem Zug geschrieben werden, kann es passieren, dass der Stick sich zu sehr erwärmt und der Schreibvorgang abbricht. Daher sollte beim Schreiben mit dem Finger die Erwärmung geprüft werden. Abhilfe kann erreicht werden durch starten weiterer Prozesse (Anwendungen), die den Vorgang ausbremsen, verwenden eines älteren ausbremsenden USB-Hubs, oder eines USB-Verlängerungskabels, so dass der Stick besser gekühlt werden kann (Gehäuse, Kühlfläche).

Auf einen USB-Stick wurde mit Hilfe von Unetbootin ein xubuntu 16.04 64-bit installiert. Dieser Stick wurde angeschlossen (USB-Anschluss auf der Seite des Einschalters). Beim Starten des Rechners mit F12 erschien ein Boot-Menü des UEFI-BIOS. Nacheinander wurde das Live-System über die Auswahl als BIOS/Legacy-Mode oder UEFI gestartet. Der BIOS-Mode funktionierte beim ersten Start, der UEFI-Mode nicht. Anschließend wurde unter dem BIOS-Mode das WLAN-Passwort eingegeben und eine Zeit im Internet nachgelesen. Anschließend wurde der UEFI-Mode noch ein-

mal gestartet und lief anstandslos. Beide Methoden waren erfolgreich und brauchten ungefähr ähnlich lange bis zum Startbildschirm. Die Auswahl von UEFI startete etwas schneller durch.

Im Boot-Menü des USB-Sticks war der Persistent-Modus nur beim Start unter der Wahl von BIOS (legacy mode) verfügbar und bei der Wahl von UEFI-Boot mußte jeweils die Editierfunktion aufgerufen werden um "persistent" zu ergänzen. Eine Recherche der Konfigurationsdateien ergab als Ursache, dass diese unterschiedliche Angaben enthielten.

Die Datei „grub.cfg“ enthielt keine Optionsangabe „persistent“ im Vergleich zur Datei „syslinux.cfg“. Die sichtbaren Textausgaben des Auswahlmenüs bei beiden Varianten waren zu knapp gehalten, so dass die wesentlichen Unterschiede eine Sichtung der *.cfg Datei erforderten, oder der Editiermodus aufgerufen werden mußte.

Unter (U)EFI-Boot wird zuerst folgende Konfigurationsdatei aufgerufen:

```
$ cat ./boot/grub/x86_64-efi/grub.cfg
insmod part_acorn
insmod part_amiga
insmod part_apple
insmod part_bsd
insmod part_dfly
insmod part_dvh
insmod part_gpt
insmod part_msdos
insmod part_plan
insmod part_sun
insmod part_sunpc
configfile /boot/grub/grub.cfg
```

Diese verweist auf folgende Konfigurationsdatei:

```
$ cat ./boot/grub/grub.cfg

if loadfont /boot/grub/font.pf2 ; then
    set gfxmode=auto
    insmod efi_gop
    insmod efi_uga
    insmod gfxterm
    terminal_output gfxterm
fi

set menu_color_normal=white/black
set menu_color_highlight=black/light-gray

menuentry "Try Xubuntu without installing" {
    set gfxpayload=keep
    linux /casper/vmlinuz.efi file=/cdrom/preseed/xubuntu.seed boot=casper quiet
    splash ---
    initrd /casper/initrd.lz
}
menuentry "Install Xubuntu" {
    set gfxpayload=keep
    linux /casper/vmlinuz.efi file=/cdrom/preseed/xubuntu.seed boot=casper only-ubi-
    quity quiet splash ---
    initrd /casper/initrd.lz
}
menuentry "OEM install (for manufacturers)" {
    set gfxpayload=keep
    linux /casper/vmlinuz.efi file=/cdrom/preseed/xubuntu.seed boot=casper only-ubi-
    quity quiet splash oem-config/enable=true ---
    initrd /casper/initrd.lz
}
menuentry "Check disc for defects" {
    set gfxpayload=keep
```

```
linux /casper/vmlinuz.efi boot=casper integrity-check quiet splash ---
initrd /casper/initrd.lz
}
```

Um nicht jedes Mal für den Persistent-Mode die Editierfunktion aufrufen zu müssen, wurde ein Menü-Eintrag ergänzt. Zur Verfolgung des Bootvorganges bevorzuge ich "quiet" und "splash" (ggf. "nosplash" ergänzen notwendig) nicht zu verwenden. In einem Ubuntu-Forum wurde empfohlen bei Startproblemen es auch mit der Option "ignore_uuid" zu probieren.

Quelle für weitere Optionen: <http://manpages.ubuntu.com/manpages/trusty/man7/casper.7.html>

```
menuentry "Try Xubuntu without installing and with persistent /boot/grub/grub.cfg" {
    set gfxpayload=keep
    linux /casper/vmlinuz.efi file=/cdrom/preseed/xubuntu.seed boot=casper persis-
tent ---
    initrd /casper/initrd.lz
}
```

Die Freude mit dem Live-Linux auf dem USB-Stick hielt nicht lange an, da dieser bald nicht mehr funktionierte. Verursacht wurde dies durch die Standardeinstellung der automatischen Updates, die jeweils früh am morgen von selbst starteten. In der Regel waren diese Updates harmlos, aber wenn diese den Kernel (oder grub, syslinux, isolinux) betreffen in Verbindung mit einem Durchlauf zur Erzeugung von den Dateien "vmlinuz" und "initrd" funktioniert der Live-Stick anschließend meistens nicht mehr.

Die Fehlermeldung für diesen Fall war: "(initramfs) unable to find a medium containing a live-file-system"

Der Live-Linux-USB-Stick wurde eine Zeit lang mit einem anderen Rechner betrieben. Dabei lief ein apt-get update und apt-get upgrade nacheinander durch. Die Installation dauerte lange, da Teile des neueren Kernels compiliert und komprimiert wurden. Unter dem BIOS/Legacy-Mode startete das Linux. Allerdings beim Start als UEFI hing es immer im Boot-Prozess nach dem Menu mit einer Fehlermeldung, dass das System nicht gefunden wurde. Beim durchsehen der Dateien, stellte sich heraus, dass im UEFI-Teil die betreffende Datei nicht aktualisiert war.

```
Im Verzeichnis /caspar/
27137771 Oct 16 18:10 initrd.lz
7053472 Oct 16 18:10 vmlinuz
7055384 Oct 16 18:10 vmlinuz.efi
7049432 Jul 20 01:01 vmlinuz.efi.01.old
```

Unter dem UEFI-Mode wurde wieder erneut ein "apt-get update" und "apt-get upgrade" durchgeführt. Diesmal wurden anscheinend beide vmlinuz erzeugt, da nun beide Dateien unterschiedliche Länge besaßen und in der Zeitangabe nur gering unterschieden.

Die initrd paßte nicht mehr zur vmlinuz.efi. Es wurde daher die Datei "vmlinuz.efi" umbenannt und die "vmlinuz" kopiert und diese in "vmlinuz.efi" umbenannt. Allerdings stockte der Boot-Vorgang wieder, aber einige Schritte später. Erst nachdem im BIOS/Legacy-Mode gestartet der Rechner einige Zeit im Internet war (später noch einmal irrtümlich Win10 startete und dabei anscheinend Updates ausführte, ohne jemals unter Windows mit Internetverbindung gelaufen zu sein), startete auch das Linux endlich unter dem UEFI-Mode.

Wenige Tage später lief ein automatisches Update ("apt-get update" und "apt-get upgrade") gemäß Standardeinstellungen durch und führte dazu, dass der Stick nicht mehr hochfuhr an den Testrechnern. Mit obigen Methoden konnte der Stick auch nicht mehr zum Laufen gebracht werden. Die Version 16.04.1 vom Juli 2016 war in dem Falle nach einem durchgeführten Update innerhalb der laufenden Live-Umgebung ab dem Monat Oktober 2016 nicht mehr verwendbar. Im Februar 2017 kam die Version 16.04.2 heraus und wurde als Live-Linux verwendet.

Um solchen Fällen vorzubeugen oder den Stick wieder herstellen zu können, wird empfohlen die wichtigsten Dateien als Sicherungskopie in einem Unterverzeichnis anzulegen, aus dem diese wieder zurückkopiert werden können. Danach funktioniert der nicht-persistente Mode in der Regel wieder. Unter Umständen funktioniert der persistente Mode nicht, da es zu Inkompatibilitäten zu den Versionen der Systemdateien im Caspar-Overlay-Dateisystem kommen kann.

Skript zur Erzeugung von md5-Summen um diese später mit "diff", "uniq" oder anderen Shell-Werkzeugen zu vergleichen bei Bedarf oder Fehlersuche.

```
IFS=$'\n\t'; for i in $(find -type f) ; do md5sum "$i" ; done | tee md5sumss_$(date +%Y%m%d_%H%M).md5.txt
```

Bei dem folgenden Vorgang wurden ungefähr insgesamt 50-60 MByte kopiert.

```
for i in $(find -type f -size -50M) ; do cp --parents "$i" DOKU/INSTALL/SIK170222/ ; done
# oder
for i in $(find -type f -size -50M | grep -v "DOKU/INSTALL" | grep -v "Trash-1000") ; do cp --parents "$i" DOKU/INSTALL/SIK_$(date +%Y%m%d_%H%M)/ ; done
```

Liste der kritischen Dateien:

```
./EFI/BOOT/BOOTx64.EFI
./EFI/BOOT/grubx64.efi
./boot/grub/efi.img
./casper/initrd.lz
./casper/vmlinuz.efi
```

Die verantwortlichen Konfigurationsdateien befinden sich im Pfad "/etc/apt/apt.conf.d/". Diese wurden folgendermaßen geändert:

```
sudo nano /etc/apt/apt.conf.d/10periodic
# Origin was before changes:
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "0";
APT::Periodic::AutocleanInterval "0";
# Content changed to:
APT::Periodic::Update-Package-Lists "0";
APT::Periodic::Download-Upgradeable-Packages "0";
APT::Periodic::AutocleanInterval "0";

sudo nano /etc/apt/apt.conf.d/20auto-upgrades
# Origin was before changes:
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Unattended-Upgrade "1";
# Content changed to:
APT::Periodic::Update-Package-Lists "0";
APT::Periodic::Unattended-Upgrade "0";
```

Eine einfache LogiLink-Maus führte übrigens auch zu dem bekannten Fehler "(initramfs) unable to

find a medium containing a live-file-system" und zu einem anderen Fehler, dass ein Medium /cow gefunden wurde, welches nicht unterstützt würde. Ohne angeschlossene Maus funktionierte wieder alles.

Weitere Änderungen am USB-Stick

Installierte Anwendungen:

```
$ sudo apt-get install mplayer
Need to get 4-5 MB of archives.
```

```
$ sudo apt-get install pmount
Need to get 73.8 kB of archives.
After this operation, 744 kB of additional disk space will be used.
```

mc

htop

vnstat

etherape

2.3.2 Untersuchung des Inhaltes der eingebauten Festplatte im Netbook

Mittels einiger Konsolenbefehle wurde die Belegung der Festplatte ins Visier genommen und die Ergebnisse in eine Datei geschrieben. Gegenüber "lsblk" und "gparted" liefert "fdisk" oder "parted" wesentlich mehr Informationen über die Dateipartitionen.

```
$ lsblk > 20160928_lsblk.txt
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda   8:0    0 465.8G  0 disk
├─sda1 8:1    0   260M  0 part
├─sda2 8:2    0    16M  0 part
├─sda3 8:3    0 419.7G  0 part
├─sda4 8:4    0    25G  0 part
├─sda5 8:5    0 1000M  0 part
├─sda6 8:6    0  18.9G  0 part
└─sda7 8:7    0 1000M  0 part
```

```
# Use fdisk or parted
$ sudo fdisk -l /dev/sda > 20160928_lsblk.txt
$ sudo parted -l /dev/sda > 20160928_lsblk.txt
Disk /dev/sda: 465.8 GiB, 500107862016 bytes, 976773168 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: B3xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx

Device          Start          End      Sectors   Size Type
/dev/sda1       2048          534527   532480    260M EFI System
/dev/sda2       534528        567295    32768     16M Microsoft reserved
/dev/sda3       567296      880652287 880084992 419.7G Microsoft basic data
/dev/sda4       880652288    933081087 52428800  25G Microsoft basic data
```

/dev/sda5	933081088	935129087	2048000	1000M	Windows recovery environment
/dev/sda6	935129088	974725119	39596032	18.9G	Windows recovery environment
/dev/sda7	974725120	976773119	2048000	1000M	Lenovo boot partition

Als Ergebnis der Festplattenuntersuchung konnte festgehalten werden, dass viele Partitionen vorhanden waren, die nicht gelöscht werden sollten. Vor einer Installation des Linux auf die Festplatte sollten die Daten solcher Partitionen gesichert werden.

2.3.3 Sicherung der Partitionen

Die folgenden Schritte dienen der Sicherung des Inhaltes wichtiger Partitionen für die Wiederherstellung des Systems im Falle eines Zerschießens der Festplattenbelegung auf Grund eines Fehlers oder anderen Defektes. Die gleichen Schritte werden benötigt, wenn man kein zweites Betriebssystem aufspielen möchte, aber die Festplatte irgendwann tauschen muss auf Grund eines Ausfalles, eine größerer Festplatte oder eine Solid State Disk (Flash-Festplatte) einbauen möchte.

Ausgegangen wird hierbei von folgender Eingangsbelegung:

Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	534527	532480	260M	EFI System
/dev/sda2	534528	567295	32768	16M	Microsoft reserved
/dev/sda3	567296	880652287	880084992	419.7G	Microsoft basic data
/dev/sda4	880652288	933081087	52428800	25G	Microsoft basic data
/dev/sda5	933081088	935129087	2048000	1000M	Windows recovery environment
/dev/sda6	935129088	974725119	39596032	18.9G	Windows recovery environment
/dev/sda7	974725120	976773119	2048000	1000M	Lenovo boot partition

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/sda1	262144	23196	238948	9%	/home/xubuntu/Documents/Changes/tmp
/dev/sda3	440042492	24583788	415458704	6%	/home/xubuntu/Documents/Changes/tmp
/dev/sda3	440042492	24583788	415458704	6%	/media/xubuntu/Windows
/dev/sda4	26214396	3611780	22602616	14%	/home/xubuntu/Documents/Changes/tmp
/dev/sda4	26214396	3611780	22602616	14%	/media/xubuntu/LENOVO
/dev/sda5	1023996	373524	650472	37%	/home/xubuntu/Documents/Changes/tmp
/dev/sda6	19798012	11127428	8670584	57%	/home/xubuntu/Documents/Changes/tmp
/dev/sda7	1019904	547636	472268	54%	/home/xubuntu/Documents/Changes/tmp
/dev/sda7	1019904	547636	472268	54%	/media/xubuntu/LRS_ESP

Dieser vordere Teil sollte mittels "dd" in eine Datei kopiert werden. Mit dieser Datei wäre eine Wiederherstellung nur auf einer Festplatte mit mindestens der gleichen Größe möglich, wenn diese auf eine Festplatte kopiert würde. Bei Profiwerkzeugen könnte nach dem Kopieren auch auf eine kleinere Platte mitgegeben werden alle Partitionen zu Lasten der Partition sda3 zu modifizieren.

Device	Start	End	Sectors	Size	Type
/dev/sda??	0 / 1	2047	2047/8	1M	unknown MBR/ GPT/GUID-Partitionen

Diese Partitionen muss auf jeden Fall gesichert werden.

Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	534527	532480	260M	EFI System
/dev/sda2	534528	567295	32768	16M	Microsoft reserved
/dev/sda7	974725120	976773119	2048000	1000M	Lenovo boot partition

Bei diesen Partitionen reicht es den Inhalt inklusive versteckter Dateien zu sichern.

Device	Start	End	Sectors	Size	Type
/dev/sda3	567296	880652287	880084992	419.7G	Microsoft basic data
/dev/sda4	880652288	933081087	52428800	25G	Microsoft basic data
/dev/sda5	933081088	935129087	2048000	1000M	Windows recovery environment
/dev/sda6	935129088	974725119	39596032	18.9G	Windows recovery environment

Von folgender Partition kann Speicherplatz abgezogen werden für eine Linux-Installation oder wenn das System auf eine kleinere Flash-Disk umgezogen werden soll:

Device	Start	End	Sectors	Size	Type
/dev/sda3	567296	880652287	880084992	419.7G	Microsoft basic data

Mit "pmount" konnten die Partitionen nicht geladen werden. Warum dies sich so verhielt, konnte nicht erörtert werden.

```
$ pmount /dev/sda1
Error: device /dev/sda1 is not removable
$ pmount /dev/sda2
Error: device /dev/sda2 is not removable
```

Mit folgendem kleinen Skript wurden die Partitionen nacheinander eingebunden und der Inhalt kurz untersucht.

```
$ for i in 1 2 3 4 5 6 7 ; do sudo mount /dev/sda$i tmp ; sleep 3 ; df ; sleep 2 ; ls tmp ; sleep 2 ; sudo umount /dev/sda$i ; sleep 5 ; done > sda_full.txt
```

```
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/sda1        262144        23196   238948    9% /home/xubuntu/Documents/Changes/tmp
```

Inhalt des Wurzelverzeichnis:

```
BOOT
EFI
```

```
Filesystem /dev/sda2: Nur Fehlermeldung (kein Superblock, usw.) Größe 16MB
```

```
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/sda3        440042492    24583788  415458704    6% /home/xubuntu/Documents/Changes/tmp
```

Inhalt des Wurzelverzeichnis:

```
Boot
bootmgr
BOOTNXT
Dokumente und Einstellungen
hiberfil.sys
Intel
pagefile.sys
PerfLogs
ProgramData
Program Files
Program Files (x86)
Programme
Recovery
$Recycle.Bin
swapfile.sys
System Volume Information
UserGuidePDF
Users
Windows
```

```
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/sda4        26214396    3611780  22602616   14% /home/xubuntu/Documents/Changes/tmp
```

Inhalt des Wurzelverzeichnis:

```
Applications
Drivers
$RECYCLE.BIN
```

System Volume Information

```

Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda5        1023996    373524    650472    37% /home/xubuntu/Documents/Changes/tmp
Inhalt des Wurzelverzeichnis:
Recovery
$RECYCLE.BIN
System Volume Information

```

```

Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda6        19798012  11127428    8670584    57% /home/xubuntu/Documents/Changes/tmp
Inhalt des Wurzelverzeichnis:
OKRBackup
onekey
System Volume Information

```

```

Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda7        1019904    547636    472268    54% /home/xubuntu/Documents/Changes/tmp
Inhalt des Wurzelverzeichnis, hier stark gekürzt:
bg-bg
Boot
bootmgr
bootmgr.efi
de-de
EFI
el-gr
sl-si
sources
uk-ua
Version.txt
zh-tw

```

```

#!/bin/bash
# Kleines Programm, das Dateilisten in Dateien schreibt.
# Liste mit allen Attributen und md5-Summen.
IFS=$'\n\t'
for i in 1 2 3 4 5 6 7
do sudo mount /dev/sda$i tmp ; sleep 3
  ( ( df | grep Filesystem ; df | grep /sda$i ) | tee -a sda_alle_kompakt.txt) ; sleep 2
  ( ls -la tmp | tee -a sda_alle_kompakt.txt ) ; sleep 2
  for j in $(find tmp/ -type f)
  do ( md5sum "$j" | tee -a md5sum_sda$i._$(date +%Y%m%d_%H%M).md5.txt )
  done ; sleep 1
  for j in $(find tmp/ -type f)
  do ( ls -la "$j" | tee -a files_sda$i._$(date +%Y%m%d_%H%M).details.txt )
  done ; sleep 5
  sudo umount /dev/sda$i ; sleep 5
done

```

```

#!/bin/bash
# Hier mit Aufteilung. Grund siehe weiter unten.
IFS=$'\n\t'
for i in 1 4 5 6 7
do sudo mount /dev/sda$i tmp ; sleep 3
  ( ( df | grep Filesystem ; df | grep /sda$i ) | tee -a sda_alle_kompakt.txt) ; sleep 2
  ( ls -la tmp | tee -a sda_alle_kompakt.txt ) ; sleep 2
  for j in $(find tmp/ -type f)
  do ( md5sum "$j" | tee -a md5sum_sda$i._$(date +%Y%m%d_%H%M).md5.txt )
  done ; sleep 1
  for j in $(find tmp/ -type f)
  do ( ls -la "$j" | tee -a files_sda$i._$(date +%Y%m%d_%H%M).details.txt )
  done ; sleep 5

```

```
sudo umount /dev/sda$i ; sleep 5
done
# Die Partitionen, die nicht eingebunden werden können
# oder zu viele Dateien hätten (hier sda3 über 30000 Dateien):
for i in 2 3
do sudo mount /dev/sda$i tmp ; sleep 3
  ( ( df | grep Filesystem ; df | grep /sda$i ) | tee -a sda_alle_kompakt.txt) ; sleep 2
  ( ls -la tmp | tee -a sda_alle_kompakt.txt ) ; sleep 2
  sudo umount /dev/sda$i ; sleep 5
done
```

3 Unterstützte Hardware des Lenovo Yoga 300 unter Linux

3.1 Bei Verwendung von xubuntu 16.04

Eigentlich funktionierte die meiste Hardware mit xubuntu 16.04 sofort ohne weitere Handgriffe anzulegen. Die Bildschirmauflösung stimmte mit der tatsächlichen Zeilen und Punktezahl überein. Das WLAN inklusive WPA2 funktionierte. Die 2D/3D-Beschleunigung wurde mit dem Spiel "Extreme Tuxracer" getestet und funktionierte auch zufriedenstellend. Mit xrandr konnten Verstellungen des Bildschirmes vorgenommen werden. Das Display wird mit "xrandr --display eDT1" angesprochen. Die Touch-Funktion des Displays funktionierte ebenfalls. Nur das Bluetooth konnte mangels geeignetes Zubehör nicht ausreichend getestet werden.

Nicht funktionierte nur die Batterieanzeige. Die Anwendung acpi konnte auch keine Ausgabe generieren. Die Batterielaufzeit betrug nach eigenen Schätzungen ungefähr vier Stunden. Beim mobilen Betrieb ist das ein großer Nachteil.

Dies gilt für xubuntu 16.04.1 bis 16.04.2. Letztere verwendet einen Kernel 4.8.0-36-generic. Eine neuere Version stand nicht zur Verfügung.

Für den Batterieanzeigefehler gibt es einen Patch seit Dezember 2016:

Hier sind noch ein paar kleinere Probleme geschildert mit Stand vom August 2016, die in 2017 größtenteils behoben sein sollten:

<https://forums.lenovo.com/t5/Linux-Discussion/Yoga-300-battery-percent-indicator/td-p/3501546>

Die Lösung des Problems und wie der Patch eingespielt wird, steht unter diesen Links:

<https://ubuntuforums.org/showthread.php?t=2340949>

<http://askubuntu.com/questions/809393/battery-not-detected-yoga-300-11ibr-ubuntu-lts-14-04>

Problem war: buggy firmware -BIX-method. Batterieanzeige soll funktionieren ab Kernel 4.10.rc5 oder 4.10.0-1.

Auf Grund dieses Ergebnisses konnte nur ein Linux mit einem aktuellen Kernel verwendet werden. Vermutlich enthält erst Ubuntu 16.04.3 einem 4.10er Kernel.

3.2 Bei Verwendung von xubuntu 17.04

Diese Version war zum Stand vom Februar 2017 noch eine Beta-Version, enthält aber den benötigten Kernel 4.10, so dass auch die Batterieanzeige funktioniert. Eine Recherche ergab, dass es zu dem Zeitpunkt nur Beta-Versionen (testing, sid) gibt, die den geeigneten Kernel enthalten.

3.3 Bei Verwendung von opensuse tumbleweed

Diese Version, tumbleweed, Release Date 2017-02-28, enthält den benötigten Kernel 4.10, so dass auch die Batterieanzeige funktioniert. Eine praktische Quelle für solche Recherchen bietet die Internetseite distrowatch:

<https://distrowatch.com/table.php?distribution=opensuse>

<http://distrowatch.com/table.php?distribution=opensuse>

Der verwendete Kernel kann der Zeile "linux" entnommen werden.

Eine Recherche ergab, dass es zu dem Zeitpunkt nur Beta-Versionen (testing, sid) gibt, die den geeigneten Kernel enthalten.

3.4 Weitere interessante Linux-Distributionen

Ebenfalls Gentoo-Linux wäre auch möglich, da diese Distribution jeweils aus den aktuellen Quellen kompiliert werden muss. In dem Falle werden keine binären Distributionspakete zur Verfügung gestellt, sondern nur die Quelldateien. Für den Start gibt es ein Image für eine Grundinstallation eines Linux-Systems mit allen benötigten Komponenten, so dass die Quelldateien für die weitere Komplettierung und Aktualisierung des Systems verarbeitet werden können.

Das gobolinix ist ebenfalls auf die Verwendung von Quelldateien ausgelegt mit der Besonderheit, dass jeweils alle benötigten zusätzlichen Dateien (Libraries) in ein gemeinsames Verzeichnis gepackt werden. Somit sind hier die Anwendungsdateien schön sauber sortiert und die Abhängigkeiten gestalten sich wesentlich einfacher. Es können bei dieser Methode auch einfacher mehrere Versionen einer Anwendung parallel auf dem System verwendet werden.

Es gibt dabei drei Nachteile, die erwähnt werden sollten meiner Ansicht nach. Der erste Nachteil wäre, dass die Anwendungen etwas länger zum Starten benötigen. Der zweite Nachteil wäre, dass die Anwendungen auch etwas mehr Speicherplatz im RAM belegen und verbrauchen, da die Bibliotheken (Libraries, bzw. lib*-Dateien) nicht so intensiv gemeinsam verwendet werden. Der dritte Nachteil wäre, dass natürlich mehr Platz auf der Festplatte belegt wird, da die Dateien auch mehrfach auf dem Laufwerk liegen, aber hier könnte über Verlinkungen gleicher Dateien (es muss die Variante gewählt werden, bei der die Verlinkung gelöst wird, wenn die Datei beim direkten Zugriff auf das Verzeichnis verändert wird) wieder einen Teil des Speichermehrbedarfs freigeben.

4 Installation eines Linux-Betriebssystems als Dual-Boot Betriebsvariante

4.1 Vorbereitungen für eine Dual-Boot Installation

Auf Grund der Untersuchung der Festplattenbelegung konnte bei dem Rechner UEFI und GPT-Partitionierung identifiziert werden. Das Betriebssystem Windows 10 verwendet Secure Boot. Vor einer Installation sollte auf einschlägigen Internetseiten und auch mit Hilfe von Suchmaschinen vor der Durchführung einige Anleitungen gelesen werden, um die meisten möglichen Fehler bei der Installation zu vermeiden.

Literaturliste zur Installation Dual Boot mit UEFI und Secure Boot:

<http://askubuntu.com/questions/221835/installing-ubuntu-alongside-a-pre-installed-windows-with-uefi>

<http://www.pc-magazin.de/ratgeber/multiboot-uefi-festplatte-partitionieren-2785863.html>

<https://wiki.ubuntuusers.de/Dualboot/>

Um ein Live-Linux auf einem USB-Stick zu starten, wurden bereits einige zwingend notwendige abzustellende Funktionen im BIOS und Windows durchgeführt.

Eine wichtige Information aus der Literaturrecherche sollte noch berücksichtigt werden, dass die Windows-Partition unter Windows mit dessen Partitionswerkzeug verkleinert wird. In dem freien Bereich werden die anzulegenden Partitionen mit dem Installationswerkzeug von Linux angelegt. Sinnvollerweise sollten mindestens eine Wurzelpartition "/" und "swap" angelegt werden.

Bei GPT ist die Reihenfolge der Partitionen bei der Anlage nicht vorbestimmt, wie bei MBR. Es kann also durchaus von sieben vorhandenen Partitionen, die dritte Partition verkleinert werden ohne die anderen Partitionen nach unten zu verschieben. In der Lücke zwischen der dritten und vierten Partition können die Partitionen für Linux angelegt werden. Wenn hier drei Partitionen erstellt werden, wären das im hier genannten Fall die Partitionen acht bis zehn.

Je nach Notebookhersteller oder Mainboard-Hersteller wird eine Partition für das UEFI auf der Festplatte angelegt und genutzt. Bei diesem Rechner heißt diese Partition Lenovo. Bei UEFI die das Label der Partition auswerten, kann diese im Prinzip an beliebiger Stelle auf der Festplatte liegen. Die Angaben über Hersteller, die hier weniger flexible Lösungen verwenden, wie zum Beispiel, dass diese am physikalischen Ende der Festplatte liegen muss, es darf nur die Partition mit der höchsten Zählnummer sein, oder sonstige Einschränkungen unterliegt, sind sehr selten. Für den Hersteller des Netbooks konnten solche Einschränkungen nicht gefunden werden, die heutzutage eine sehr seltene Ausnahme sein sollten.

4.2 Änderung der Partitionierung

4.2.1 Geplante Partitionierung

Die Partition für Windows belegte rund 400GB auf der Festplatte. Es wurde entschieden, dass diese Partition auf 100GB-150GB geschrumpft werden sollte, damit genügend Platz aus zum Probieren verschiedener Distributionen verbliebe. Ein Hintergedanke war auch, dass eventuell ein abgeschlossenes Linux-System über das andere installierte Linux-System über die Einbindung als chroot-

Umgebung wieder repariert werden könnte. Früher verwendete ich unter hierfür grub als Startkette auf MBR-Festplatten (grub im MBR --> grub in der Linux-Partition (hierzu war in der separaten Partiton für die Dateien ein Linux im Textmode installiert, das grub im MBR nutzte.)).

Angelegt werden sollten für die Betriebssysteme jeweils Partitionen von ungefähr 50-64GB und eine Swap-Partition von 4-8GB.

4.2.2 Sicherung der Dateien auf dem Rechner

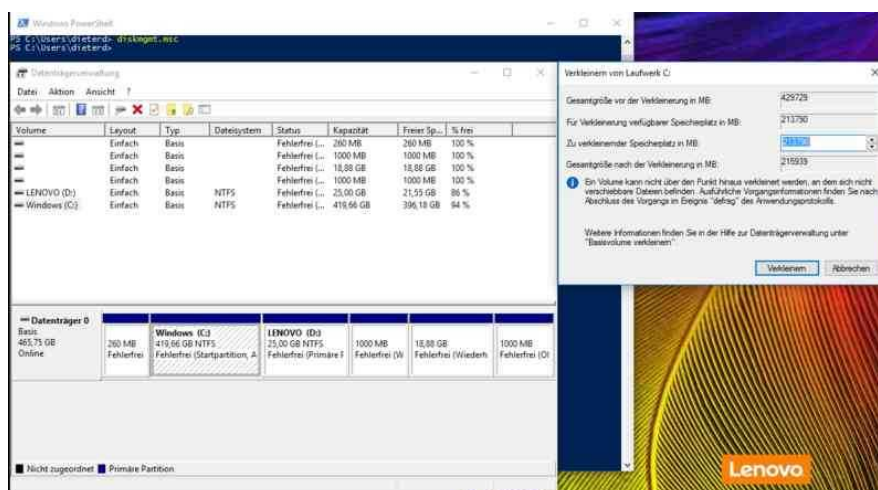
Als erstes sollten unbedingt die Dateien auf dem Rechner für eine spätere Wiederherstellung gesichert werden, bevor die Partitionen verändert werden.

Nachdem auf dem Windows-System noch nichts verändert wurde, so dass durch ein Recovery weder Daten noch Anwendungen ernstlich verloren wären und auch der Zeitaufwand für die Wiederherstellung gering gewesen wäre, wurde die Sicherung verschoben auf den Zeitpunkt zwischen der Verkleinerung der Windowspartition mit dem Diskmanager von Windows und Anlegen der neuen Partitionen.

4.2.3 Verkleinern der Windows-Partition

Hierzu wurde das Windows 10 gestartet, die Power Shell aufgerufen und der Befehl "diskmgmt" eingegeben. Es sollte die Windows-Partition (unter Linux sda3) verkleinert werden.

Die Partion sda3 wurde so weit es Windows 10 mit dem diskmgmt zuließ verkleinert. Hier zeigte sich von Windows bereits wieder eine versteckte Beschränkung. Die Partition ließ sich trotz genügend freien Speichers nur auf 260GB reduzieren. Der Hinweis auf einschlägigen Seiten, man solle das Windows-Werkzeug verwenden, war sinnvoll zu beachten. Würde die Partition anderweitig noch stärker verkleinert, würde das Windows nicht mehr starten. In dem Blogs war allerdings Limit von Seiten Windows nicht entsprechend erklärt, die von mir gefunden wurden.



Das Ergebnis der Verkleinerung ist unter folgendem Screenshot zu sehen.

Volume	Layout	Typ	Dateisystem	Status	Kapazität	Freier Sp...	% frei
	Einfach	Basis		Fehlerfrei (...)	260 MB	260 MB	100 %
	Einfach	Basis		Fehlerfrei (...)	1000 MB	1000 MB	100 %
	Einfach	Basis		Fehlerfrei (...)	18,88 GB	18,88 GB	100 %
	Einfach	Basis		Fehlerfrei (...)	1000 MB	1000 MB	100 %
LENOVO (D:)	Einfach	Basis	NTFS	Fehlerfrei (...)	25,00 GB	21,55 GB	86 %
Windows (C:)	Einfach	Basis	NTFS	Fehlerfrei (...)	210,88 GB	187,46 GB	89 %

Datenträger 0		260 MB	210,88 GB NTFS	208,78 GB	25,00 GB NTFS	1000 MB	18,88 GB	1000 MB
Basis 465,75 GB Online		Fehlerfrei	Fehlerfrei (Startparti)	Nicht zugeordnet	Fehlerfrei (Prim)	Fehlerfrei	Fehlerfrei (Wies)	Fehlerfrei

■ Nicht zugeordnet ■ Primäre Partition

4.2.4 Nachgeholte Sicherung der Daten auf der Festplatte

Spätestens jetzt müssen die Daten für Fall eines gänzlichen Fehlschlags mit Löschung der internen Festplatte (Daten und Partitionen) gesichert werden. Der Rechner wurde mit dem Live-Linux neu gestartet. Hierzu werden auf einer externen Festplatte ein Verzeichnis mit geeigneten Unterverzeichnissen erstellt.

```
mkdir Lenovo
mkdir Lenovo/sda0
...
mkdir Lenovo/sda7
```

Vollständigerhalber wird hier die Partitionierung nach der Verkleinerung angeben:

```
$ sudo fdisk -l /dev/sda
Disk /dev/sda: 465.8 GiB, 500107862016 bytes, 976773168 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: B3.....

Device      Start      End      Sectors  Size Type
/dev/sda1   2048      534527   532480   260M EFI System
/dev/sda2   534528    567295   32768    16M Microsoft reserved
/dev/sda3   567296   442810367 442243072 210.9G Microsoft basic data
/dev/sda4   880652288 933081087 52428800  25G Microsoft basic data
/dev/sda5   933081088 935129087 2048000  1000M Windows recovery environment
/dev/sda6   935129088 974725119 39596032  18.9G Windows recovery environment
/dev/sda7   974725120 976773119 2048000  1000M Lenovo boot partition
```

Durch pmount ließen sich die Partitionen nicht einbinden.

```
$ pmount /dev/sda1
Error: device /dev/sda1 is not removable
```

```
$ sudo mount /dev/sda1 ~/Documents/Changes/tmp

$ mount | grep sda1
/dev/sda1 on /home/xubuntu/Documents/Changes/tmp type vfat
(rw,relatime,fmask=0022,dmask=0022,codepage=437,ioccharset=iso8859-1,shortname=mixed,er-
rors=remount-ro)

$ df | grep sda1
/dev/sda1 262144 23196 238948 9% /home/xubuntu/Documents/Changes/tmp
```

Die Dateien wurden komplett kopiert auf die externe Festplatte.

Anschließend wieder ausgehängt um jeweils die nächste Partition einzuhängen.

```
$ sudo umount /dev/sda1
```

Die kleine Partition von nur 16MB ließ sich nicht einbinden. Hier wird später das Image als Sicherung angelegt.

```
$ pmount /dev/sda2
Error: device /dev/sda2 is not removable

$ sudo mount /dev/sda2 ~/Documents/Changes/tmp
mount: wrong fs type, bad option, bad superblock on /dev/sda2,
missing codepage or helper program, or other error

In some cases useful info is found in syslog - try
dmesg | tail or so.
```

```
$ sudo mount /dev/sda3 ~/Documents/Changes/tmp

$ mount | grep sda3/dev/sda3 on /home/xubuntu/Documents/Changes/tmp type fuseblk
(rw,relatime,user_id=0,group_id=0,allow_other,blksize=4096)
$ df | grep sda3
/dev/sda3 221121532 24550176 196571356 12% /home/xubuntu/Documents/Changes/tmp
```

Die Dateien wurden komplett kopiert auf die externe Festplatte. Der Kopiervorgang benötigte einige Zeit (30-40 Minuten). Es gab das Problem, dass die symbolischen Links nicht kopiert werden konnten. Also wurde für die 32GB an Daten "Skip for all" angegeben. In der Regel muss hier alles über die Recovery-Funktion wieder installiert werden. In dem Falle sollte der Urzustand kopiert werden.

Die Sicherungsfestplatte war vfat und nicht ntfs. Hier sollte später noch einen ntfs angelegt werden, um die Daten korrekt zu kopieren, d.h. mit den Attributen.

```
# Die externe Festplatte enthält ein erweitertes FAT32 Dateisystem.
$ mount | grep sdc1
/dev/sdc1 on /media/xubuntu/INTENSO type vfat
(rw,nosuid,nodev,relatime,uid=999,gid=999,fmask=0022,dmask=0022,codepage=437,ioccharset=i-
so8859-1,shortname=mixed,showexec=utf8,flush,errors=remount-ro,uhelper=udisks2)
```

```
cfdisk -l /dev/sda verrät, dass sda3, sda4 und sda6 ntfs-Partitionen sind. Die anderen Partitionen haben den GUID-Typ 63.
```

Der Kopiervorgang brach, ab wegen überschreiten des maximalen Speicherbereichs eines Direktories (Verzeichniseinträge). Die Anzahl an Dateieinträgen des vfat-Verzeichnisses wurde überschritten.

```
$ sudo umount /dev/sda3
```

```
$ sudo mount /dev/sda4 ~/Documents/Changes/tmp
$ mount | grep sda4
/dev/sda4 on /home/xubuntu/Documents/Changes/tmp type fuseblk
(rw,relatime,user_id=0,group_id=0,allow_other,blksize=4096)
df | grep sda4
/dev/sda4 26214396 3611780 22602616 14% /home/xubuntu/Documents/Changes/tmp
```

Das kopieren dauerte 2-3 Minuten. Verlinkungen der Dateien wurden nicht gemeldet, d.h. es war kein Skip-All wegen Verlinkungen notwendig gewesen.

```
$ sudo umount /dev/sda4
$ sudo mount /dev/sda5 ~/Documents/Changes/tmp
xubuntu@xubuntu:/media/xubuntu/INTENSO/Lenovo$ mount | grep sda5
/dev/sda5 on /home/xubuntu/Documents/Changes/tmp type fuseblk
(rw,relatime,user_id=0,group_id=0,allow_other,blksize=4096)
$ df | grep sda5
/dev/sda5 1023996 373524 650472 37% /home/xubuntu/Documents/Changes/tmp
xubuntu@xubuntu:/media/xubuntu/INTENSO/Lenovo$
```

Die Dateien konnte ohne Probleme kopiert werden.

```
$ sudo umount /dev/sda5
$ sudo mount /dev/sda6 ~/Documents/Changes/tmp
$ mount | grep sda6
/dev/sda6 on /home/xubuntu/Documents/Changes/tmp type fuseblk
(rw,relatime,user_id=0,group_id=0,allow_other,blksize=4096)
df | grep sda6
/dev/sda6 19798012 11127428 8670584 57% /home/xubuntu/Documents/Changes/tmp
```

Das kopieren dauerte 3-4 Minuten. Verlinkungen der Dateien wurden nicht gemeldet, d.h. es war kein Skip-All wegen Verlinkungen notwendig gewesen.

Aber die Datei install.wim war zu groß für vfat und mußte abgebrochen werden.

```
$ sudo umount /dev/sda6
$ sudo mount /dev/sda7 ~/Documents/Changes/tmp
$ mount | grep sda7
/dev/sda7 on /home/xubuntu/Documents/Changes/tmp type vfat
```

```
(rw,relatime,fmtmask=0022,dmask=0022,codepage=437,ioccharset=iso8859-1,shortname=mixed,errors=remount-ro)

$ df | grep sda7
/dev/sda7      1019904    547636    472268   54% /home/xubuntu/Documents/Changes/tmp
xubuntu@xubuntu:/media/xubuntu/INTENSO/Lenovo$
```

Die Dateien konnte ohne Probleme kopiert werden.

```
$ sudo umount /dev/sda7
```

Nun mußte noch auf der externen Festplatte eine ntfs-Partiton erstellt werden. Das erweiterte FAT32 Dateisystem konnte von gparted 0.19 nicht verkleinert werden. Auch nachinstallieren von mtools gemäß einer Fehlermeldung half nicht weiter.

Letztendlich wurde auf den harten Weg mittels parted auf der Konsole die Partition verkleinert.

```
(parted) resizepart 1
End? [1000GB]? 700GB
Warning: Shrinking a partition can cause data loss, are you sure you want to
continue?
Yes/No? yes
```

Trotz Fehlermeldung bei Verwendung von fsck konnte die Partiton eingebunden, gelesen und geschrieben werden. Unter Windows konnte das Dateisystem nicht mit chkdisk geprüft werden, da diese Anwendung monierte das Format "RAW" nicht reparieren zu können. Geprüft wurde es über den Aufruf in einem der Unterreiter des Dateimanager/Eigenschaften (rechter Mausklick).

Die ntfs-Partiton war mit gparted sofort angelegt. Angemerkt sei hier, dass es deutlich schneller ginge, wenn ein gparted unter Windows installiert wurde und hierfür verwendet würde.

Nun konnten die ntfs-Partitionen (sda3, sda4, sda6) kopiert werden. Die Details wurden hier weglassen.

Anschließend konnte mittels dem Befehl dd die Images der nicht einbindbaren Bereiche und Partitionen angelegt werden.

```
$ dd if=/dev/sda of=sda0_GPTMBR.img count=2048

$ dd if=/dev/sda2 of=sda2.img
```

Angemerkt sei noch, dass die EFI-Partition und die Partition mit der Bezeichnung (Label) Lenovo, die wichtigsten zu sichernden Dateien enthalten. Die Partition Lenovo enthält auch die Dateien für das mit ausgelieferte Betriebssystem. Eine Wiederherstellung aus den Dateien der Lenovo-Partition setzt in der Regel das Gerät auf "werkszustand" zurück. Eine Wiederherstellung aus der Windows-Recovery-Partition besitzt häufig auch eine Option für einen Reparaturversuch.

4.2.5 Anlegen von zusätzlichen Partitionen für Linux

Die neuen und zusätzlichen Partitionen wurden mit gparted erstellt. Es wurden Linux-Partitionen mit ext4-Dateisystem angelegt.

```
s$ sudo fdisk -l /dev/sda
Disk /dev/sda: 465.8 GiB, 500107862016 bytes, 976773168 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: B3xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Device            Start          End      Sectors   Size Type
/dev/sda1         2048          534527   532480    260M EFI System
/dev/sda2         534528        567295    32768     16M Microsoft reserved
/dev/sda3         567296 442810367 442243072 210.9G Microsoft basic data
/dev/sda4         880652288    933081087 52428800    25G Microsoft basic data
/dev/sda5         933081088    935129087  2048000   1000M Windows recovery environment
/dev/sda6         935129088    974725119 39596032   18.9G Windows recovery environment
/dev/sda7         974725120    976773119  2048000   1000M Lenovo boot partition
/dev/sda8         442810368    546140159 103329792  49.3G Linux filesystem
/dev/sda9         546140160    659206143 113065984  53.9G Linux filesystem
/dev/sda10        659206144    862937087 203730944  97.2G Linux filesystem
/dev/sda11        862937088    880652287  17715200   8.5G Linux swap

Partition table entries are not in disk order.
```

Nun wurde der Rechner heruntergefahren und getestet, ob das andere Betriebssystem noch startete. Der Test verlief erfolgreich.

4.3 *Installation eines Linux-Betriebssystems auf dem Notebook*

4.3.1 **Installation von Ubuntu 16.04**

Es wurde beschlossen ein Ubuntu 16.04 zu installieren und dieses später auf 17.04 hoch zu ziehen. Vor der Installation sollten unbedingt noch einmal die Quellen konsultiert werden.

Quellen:

<https://wiki.ubuntuusers.de/Dualboot/>

https://wiki.ubuntuusers.de/EFI_Installieren/

<http://www.tecmint.com/install-ubuntu-16-04-alongside-with-windows-10-or-8-in-dual-boot/>

<http://linuxbsdos.com/2016/06/05/dual-boot-ubuntu-16-04-and-windows-10-on-a-pc-with-uefi-firmware/>

<http://www.everydaylinuxuser.com/2015/11/how-to-install-ubuntu-linux-alongside.html>

Es wurde für die Installation xubuntu vom USB-Stick gestartet. Es gibt zwar die Option bereits im Startmenü, aber es wurde hier das xubuntu im Persistent-Mode gestartet. Anschließend wurde der Ubuntu-Installer "ubiquity" gestartet.

Zu Beginn wurde die Sprache und die Tastaturbelegung abgefragt. Es wurde im Menü die Punkte Updates und Pakete aus dritten Quellen (Proprietäres) entsprechend nicht ausgewählt. Ein Versuch mit Auswahl der Pakete aus dritten Quellen (Proprietäres) ließ den Vorgang stocken, d. h. auch nach 20 Minuten warten, konnte nicht fortgesetzt werden.

Es wurde hier die Installationsart "etwas anderes" / "something else" gewählt. **Die Windowspartition bei Dual-Boot muss vorher unter Windows 10 verkleinert werden**, da sonst unter Umständen eine Begrenzung von Windows nicht greift und das Betriebssystem unter Umständen nicht mehr startet und funktioniert.

Es wurde im Menü die Punkte Updates und Pakete aus dritten Quellen (Proprietäres) entsprechend nicht ausgewählt. Trotzdem wurden beim Installieren Pakete (Updates) aus dem Internet gezogen. In der Vergangenheit scheiterte bei diesem Punkt der Installation mangels Internetverbindung. Es war überraschend diesen Unfug hier wieder zu finden, da dies eine komplette Offline Installation komplett vereiteln könnte. Unter Nutzung des kleinen Detailfensters des Installers konnte gesehen werden, dass die Paketinformationsdateien (apt-get update) herunter geladen wurden und bei der langsamen Verbindung 2h benötigte. Nach diesem Schritt begann das Herunterladen der Paketinstallationsdateien (apt-get upgrade -d). Bei diesem Schritt wurde dem Rechner das Internet entzogen. Nach wenigen Minuten (ca. 2-3) wurde die Installation ohne Internetverbindung fortgesetzt.

Bei der Auswahl der Partitionen wurde die angelegte Partition sda8 für root gewählt. Da vom USB-Stick gebootet wurde, mußte explizit eingestellt werden die swap-Partition sdb3 des USB-Sticks nicht zu verwenden. Erst danach war es möglich die swap-Partition sda11 der internen HD im Menü einzustellen und zu verwenden. Dies hätte allerdings nach der Installation in der Datei /etc/fstab auch behoben werden können.

Parallel wurde noch geprüft, ob die richtige EFI-Partition angesprochen würde, da in der Anleitung stand, hier müßte keine Auswahl der Partition für den Bootmanager getroffen werden.

```
$ df
/dev/dsa8  /target
/dev/sda1  /target/boot/efi
```

Wenn hier etwas falsches stünde, müßte so schnell wie möglich der Vorgang beendet werden damit das Schreiben des Bootladers nicht noch mehr Schaden anrichten könne.

Nach der Installation erschien während des Startvorganges das Auswahlmenü von grub. Zunächst wurde Windows (Start von der ersten Partition) ausgewählt. Das Betriebssystem Windows 10 funktionierte ohne Probleme. Anschließend wurde Ubuntu gestartet und funktionierte auch ohne Probleme. Unter dem Menüpunkt Ausschalten konnte die wählbare Option des Schlafmodes unter Ubuntu auch erfolgreich getestet werden.

4.4 Installation eines weiteren Betriebssystems

Mit einem dritten Betriebssystem läge nun ein "Tripple-Boot" System vor. Als weitere Systeme kämen zum Beispiel ein rpm-basiertes Linux, Slackware Linux oder ein openBSD in Betracht.

