

So, here is how I proceeded to get existing Linux Debian from another laptop (very old HP Compaq laptop) working on the new laptop (configuration above) - please, if you decide to go this way, be careful not to lose any data, like expected I post here this description without any warranty :

0. I tested the preinstalled Windows are working properly

1. Before the start of the system I put the Debian Live-USB into some USB slot and connected network via ethernet cable.

2. After the pressing the power-on button I pressed the key F2 to start the setup

3. In setup I changed the boot order so the boot will now run from the USB, saved and exited setup

4. Booted with debian live on usb, shrunk the preinstalled windows partition (there was approximately 53GB of data to cca 70 GB and moved into the end of the disk (now still the preinstalled Windows 10 were bootable after the move). I have used gparted installed somehow like

```
sudo apt-get update && sudo apt-get install gparted
```

(it was not in my default debian live cd installation, there are other tools, which you can possibly use, like parted, is pretty much the same, but you have to go through the text interface ;-)

4.1. Useful hint 1: if your screen in live USB system user session is locked, because you did leave the computer for slightly longer time, do not panic and use the password „live“ for the default live-USB debian user to unlock the session again, you do not have to reboot.

4.2. Useful hint 2: I have connected the laptop to the network with ethernet cable. The debian Live-USB-System could not connect to wifi by default, so I postponed the solution of this problem to later (see points 21 and 23 below)

5. I put into the other usb port the clonezillas live USB, reboot, changed in setup the boot order again to boot the clonezillas USB and boot the clonezillas USB system.

6. With clonezilla I backed up (in the device-to-image mode) the partitions sda1 (EFI), sda2 (Windows special partition – Microsoft reserved) and sda3 (windows recovery) as a partition images to the shrunk sda4 partition (Windows system), which will be in clonezilla mounted as /home/partimag (clonezillas default functionality) and used to store the partitions backup.

7. In clonezillas command prompt I saved to mounted sda4 partition the output of  

```
sudo parted unit B print > /pathmountedwindowspartition/original-partitions-data-bytes.txt
```

  
and  

```
sudo parted unit s print > /pathmountedwindowspartition/original-partitions-data-sectors.txt
```

  
for the possibility of the recovery of the partitions exactly like they were before

8. In clonezillas command prompt I saved with dd all bytes before the start of the first partition – this is the GPT area, containing the data for the booting, etc (see in the internet how to recognize if there is used MBR type or GPT type of booting information, but today

and on newly installed windows are always GPT to expect), so it was something like  
dd if=/dev/sda of=/pathmountedwindowspartition/original-gpt-data.img bs=1048576  
count=1

where the number 1048576 is the start byte of the first partition (EFI-partiton) which can be found in the file /pathmountedwindowspartition/original-partitions-data-bytes.txt – I had there 1048576, so everything before this had to be backed up, since the bytes are numbered from 0, the size of the data before the partition matches the numerical start of the byte partition.

9. I deleted the backed up partitions sda2 and sda3 and created my linux partitions system (100GB) and data (150 GB) – so I left there the EFI partition, which will be used with Linux Debian too

10. I Used clonezilla (in device-to-device mode) to copy my Debian Stretch partitions from the old disk to the newly created partitions on new laptop from my old computer's disk (I used the external disk docking for this purpose)

11. I rebooted to the Debian live USB system (again, possibly the boot order in setup has to be changed) and mounted the copied linux partition with something like  
sudo mkdir -p /mnt/linux && sudo mount /dev/sda2 /mnt/linux  
(you will have to repeat this after every reboot to live-USB-system if you want to access the partition on harddisk)

12. Now we will chroot into the copied installation and proceed there. For this some preparation steps are necessary:

12.1. I started

```
cat /mnt/linux/etc/sources.list
```

and looked from which site I normally download the debian packages – it was ftp.cz.debian.org. So I pinged this site and obtained the IP address.

12.2. I started statement

```
sudo kate /mnt/linux/etc/hosts
```

and added the recovled IP and DNS name to the hosts – you can use any text editor in the live USB to edit the file

12.3. prepare the special devices for chroot by issuing the statements

```
sudo mount -B /dev /mnt/linux/dev
```

```
sudo mount -B /dev/pts /mnt/linux/dev/pts
```

```
sudo mount -B /proc /mnt/linux/proc
```

```
sudo mount -B /sys /mnt/linux/sys
```

```
sudo mount -B /run /mnt/linux/run
```

12.4 I changed finally root by

```
sudo chroot /mnt/linux
```

13. I restored the Grub on EFI in the chrooted environment, like described in <https://wiki.debian.org/GrubEFIReinstall>, so:

13.1. Mount the EFI partition as /boot/efi:

```
mkdir /boot/efi && mount /dev/sda1 /boot/efi
```

13.2. if you now run  
`ls /boot/efi`  
you see the empty drive with the only directory EFI

13.3. I Installed (refreshed if previously installed) grub-efi:  
`apt-get install --reinstall grub-efi`  
(in my case I had not installed grub-efi, so I issued also the previous statement  
`sudo apt-get install grub-efi-amd64`  
but I think this should not be necessary, the grub-efi package installation should be enough)

13.4. I have put the debian bootloader in /boot/efi and created the appropriate entry in computers NVRAM by issuing  
`grub-install /dev/sda`

13.5. Then I created new grub-config based on current partitioning schema by issuing:  
`update-grub`

13.6. I checked if the debian efi has been created correctly:  
file /boot/efi/EFI/debian/grubx64.efi  
should bring something like  
/boot/efi/EFI/debian/grubx64.efi: PE32+ executable (EFI application) x86-64 (stripped to external PDB), for MS Windows

13.7 I checked the NVRAM items have been generated properly:  
`efibootmgr --verbose | grep debian`  
this should bring you one (new) line.

14. Now I was able to boot again Windows without problems and the Linux system will generally boot too (but be aware if you reboot the system and you need the chrooted environment again, you have to repeat some from the steps above, like, for example the remounting of live-cd special directories to be available in chrooted environment etc.), but there are still some points, which prevent Linux from being started (at least in my case) which have to be solved first:

14.1. I modified the mount points listed in /etc/fstab in the chrooted system (or /mnt/linux/etc/fstab if not chrooted) according to the current partition situation (if you mounted root on /dev/sda1 before in the old system, now you have to mount it on /dev/sda2 and similar changes) and/or possibly according to the current UUID of the partitions (if you have the mounting based on partition UUID rather than just on the device naming, use the tool blkid to retrieve the currently valid UUID partition values and replace with them the Ids of the partitions in fstab – this does not affect the copied partitions, where the UUID is the same, but possible other, you want to mount from linux). Because I rebooted and lost all the setup of the chrooted environment, I used following statements in the not-chrooted environment  
`sudo kate /mnt/linux/etc/fstab`  
and in my case, I originally removed swap – I hoped I will never need it again with 64GB of RAM ;-)- but I have been forced to create it later in the size cca 70GB because of the installation of the hibernation functionality, the hibernation does not work if you do not have

the swap partition - and also removed the non-existent cdrom device and saved the modified fstab.

15. I rebooted, started BIOS and selected now as the default EFI boot the new created option debian, which now should appear in your boot menu in BIOS

16. Remove the USB with the live-debian system and reboot again

17. The grub menu appeared now, on the first place with debian grub boot options, I had the possibility to boot default linux, go to submenu for debian kernel selection, boot Windows or start the setup (great, isn't it? :-)

18. I have tested the setup start, then if my Windows start still well and I can log in - everything was OK.

19. After I rebooted, I finally started the Debian system.

20. Voila - My Debian system started, I could log in to KDE and enjoy the speed of the new hardware.

Hint: I used the vanilla kernel of Debian (version 4.9.0-0.bpo.3-amd64), in the case you have some specially compiled kernel and some from your devices does not work correctly, you have to perform your own kernel tweaking.

21. In my case there was remaining problem with the Wifi. Because I was still running old Debian Stretch, I had to install additional packages which would support the Intel AC-9260 wifi & bluetooth device – wifi, as well as bluetooth were not accessible, but even this has not helped in the first stage. So I performed altogether following steps, which might not be necessary, if you use a different kernel or generally different configuration in your old system:

21.1. I started

```
lspci | grep Network
```

what brought me information I have on the system the Intel Corporation Wireless-AC 9260 rev. 29

21.2. I have digged more deeply and figured out, my package firmware-iwlwifi is already very old, so I upgraded it from backports, but it did not help.

21.3. Starting

```
lspci -k
```

turned out, the adapter is found, but uses no kernel module – possibly because in the old kernel is no support for such an old device or because some additional packages need to be installed.

21.4. I also found some other issues even in brand new kernels with this device, the users described sometimes the device did not come up after reboot to windows in the dual booting environment, etc. So I decided to advance the next step and upgrade first from Debian Stretch to Debian Buster, what I wanted to do in every case too later. But you can try to get working wifi on Stretch with the way, described in the point 23 even without the upgrade to Buster – it might be only the question, if there will be available the latest drivers in backports, see the

point 23.

22. For the debian upgrade i followed the steps on the page <https://tecadmin.net/upgrade-debian-9-to-debian-10-buster/> but there are a lot of similar pages handling this issue.

22.1. Useful hint: The upgrade has been executed without major problems – if you do it and use Sophos or any other antivirus, turn it better off before the upgrade otherwise some libs may not be installed and the installation can freeze and you have to fix then the broken upgrade, what was my case and took me some time till I got to the cause of the problems

(To turn OFF Sophos on-the-fly scanning: `sudo /opt/sophos-av/bin/savconfig set LiveProtection false`

To turn ON Sophos on-the-fly scanning: `sudo /opt/sophos-av/bin/savconfig set LiveProtection true`)

22.2. After upgrade the X-Windows have not started, there was necessary to install the packages `nvidia-detect` and `nvidia-driver`, which I did not have installed because my old computer used a different graphic card (note: use the latest version from backports, the older did not support the type of the given graphic card RTX 2060).

23. To get wifi working following steps were necessary:

23.1. Upgrade of the packages `firmware-iwlwifi`, `intel-microcode` (non-free) and `firmware-misc-nonfree` to the last version from backports (version at least from July 2019)

23.2. Activate wifi in the sniplet for the network management, coming after the click on the tray icon

23.3. Go to the defined Wifi networks and change the MAC of the wifi, to which the connection is limited, to the new MAC of the wifi of the new computer – with the copy of the partition all settings have been taken over, together with the MAC addresses from old computer, what is, of course, invalid on the new laptop.

24. I have activated hibernation functionality by following steps:

24.1. I created new linux swap partition of the size 70GB (recommendation from the `hibernate` package says 2x so much like memory amount, but I feel this is not necessary with this amount of memory

24.2 I inserted the new swap into the `fstab` (UUID can be get again with `blkid` started from root console)

24.3 I updated the file `/etc/initramfs-tools/conf.d/resume` with the UUID of the new swap partition

24.4. I installed the packages `hibernate` and `uswsusp`.

24.5. Now the `hibernate` functionality has been working correctly

25. Do not forget to perform any cleanups of temporary data you have saved during the problem solutions and should no longer be on the clean system.