UEFI-RAMOS and BIOS-RAMOS tutorial based on svbus driver + grub4dos or grub2

Author: **liuzhaoyzz** http://wuyou.net/forum.php?mod=viewthread&tid=423275&page=1&extra=#pid4192803

Translated by alacran from reboot.pro http://reboot.pro/ by means of Google Translate.

With the grub4dosUEFI or grub22020-12-10 version with the svbus driver, with the efforts of wintoflash, 2011yaya2007777, sunsea and other great gods, we finally overcome the UEFI-RAMOS production, which is a full-memory UEFI that does not rely on Microsoft's ramdisk.sys driver -RAMOS brand new solution, in fact, the production method is very simple, it is to install the driver + write the boot menu, but there are still many aspects that have not been verified. It is hoped that the release of this post can further lead everyone to study the full-memory UEFI-RAMOS solution based on primo driver and play a role in attracting new ideas. This article also talks about the boot menu under BIOS-RAMOS.

1. Software and hardware requirements

UEFI boot, the current UEFI firmware is generally 64-bit, generally requires 64-bit WIN7, 8, 10 operating system, currently I have successfully produced svbus-based WIN7, WIN8, WIN10X64-RAMOS. (Note that WIN7 Enterprise Edition and Ultimate Edition natively support VHD startup, WIN7 Home Edition and Professional Edition do not natively support VHD startup; WIN8 and above seem to support VHD startup natively) It is recommended to use WINNTSETUP to install in a fixed-size VHD, and optimize during installation In the settings, it is recommended to tick off virtual memory and hibernation, physical memory ≥ 50% to 70% of the total space of the VHD disk + 4GB, physical memory is recommended to be 8G or more (the larger the better), and a streamlined system is recommended.

BIOS booting, WIN7 Enterprise Edition and Ultimate Edition and above are more convenient to make because it supports VHD boot natively; XP/2003 does not support VHD boot natively, you need to inject the svbus driver offline (I didn't try this), or after installing the driver online Copy to VHD (I didn't try this), BIOS-RAMOS production, it is recommended that you directly use the "RAMOS One-Key Batch Processing All-in-One V3.8.7" program outside of Mi.

2. Tools and software that may be used

grub4dos version after UEFI2020-12-09 or version after grub22020-12-10, bootice, WINPE, WINNTSETUP single file version, Windows installation package (may be iso, esd, wim format), double drive or driver wizard, for WIN7 May need ceomsx of Hongmao Sakuragi or usb3 driver injection tool of sinoxer to spare (not required for WIN8 and above)

I share some of the tools on the network disk. https://cloud.189.cn/t/m6J3EvFnmIvu

Scan the QR code to log in on WeChat to download. (Free)

3. VHD system installation

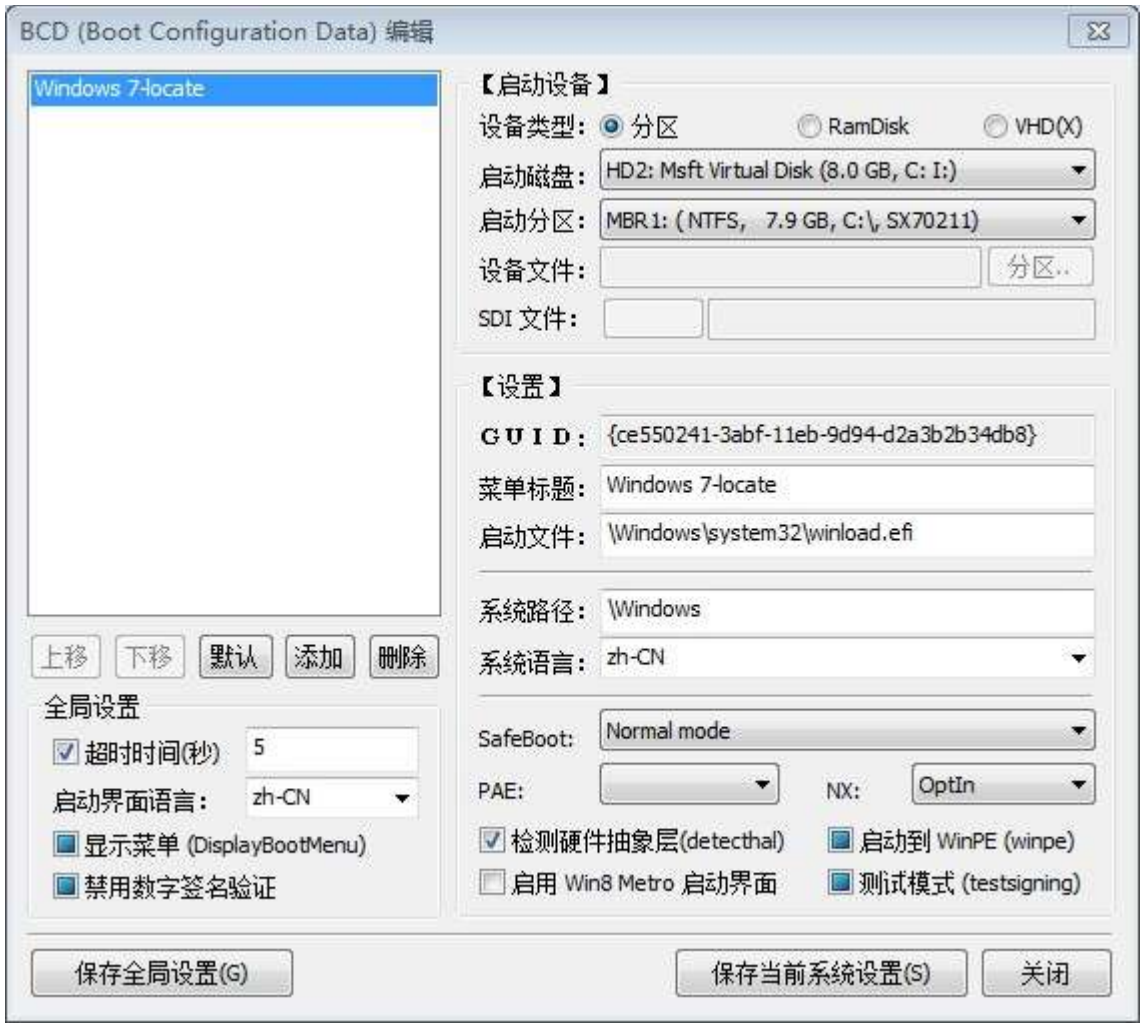
WINNTSETUP installation system tutorial (ramos newbie entry post)-RAMOS-worry-free startup forum http://wuyou.net/forum.php?mod=viewthread&tid=411864&extra=page%3D1

Note (1) A fixed-size VHD is recommended. (2) The VHD of WIN7 and WIN8 needs to use FAT32+NTFS dual partitions. For the VHD of WIN10, I tested NTFS single partition. The partition format I used is MBR, which can ensure dual boot of BIOS/UEFI. (3) Both the internal BCD of the FAT32+NTFS dual partition VHD and the external BCD of the VHD need to be adjusted accordingly, otherwise the VHD and RAMOS cannot be started. details as follows:

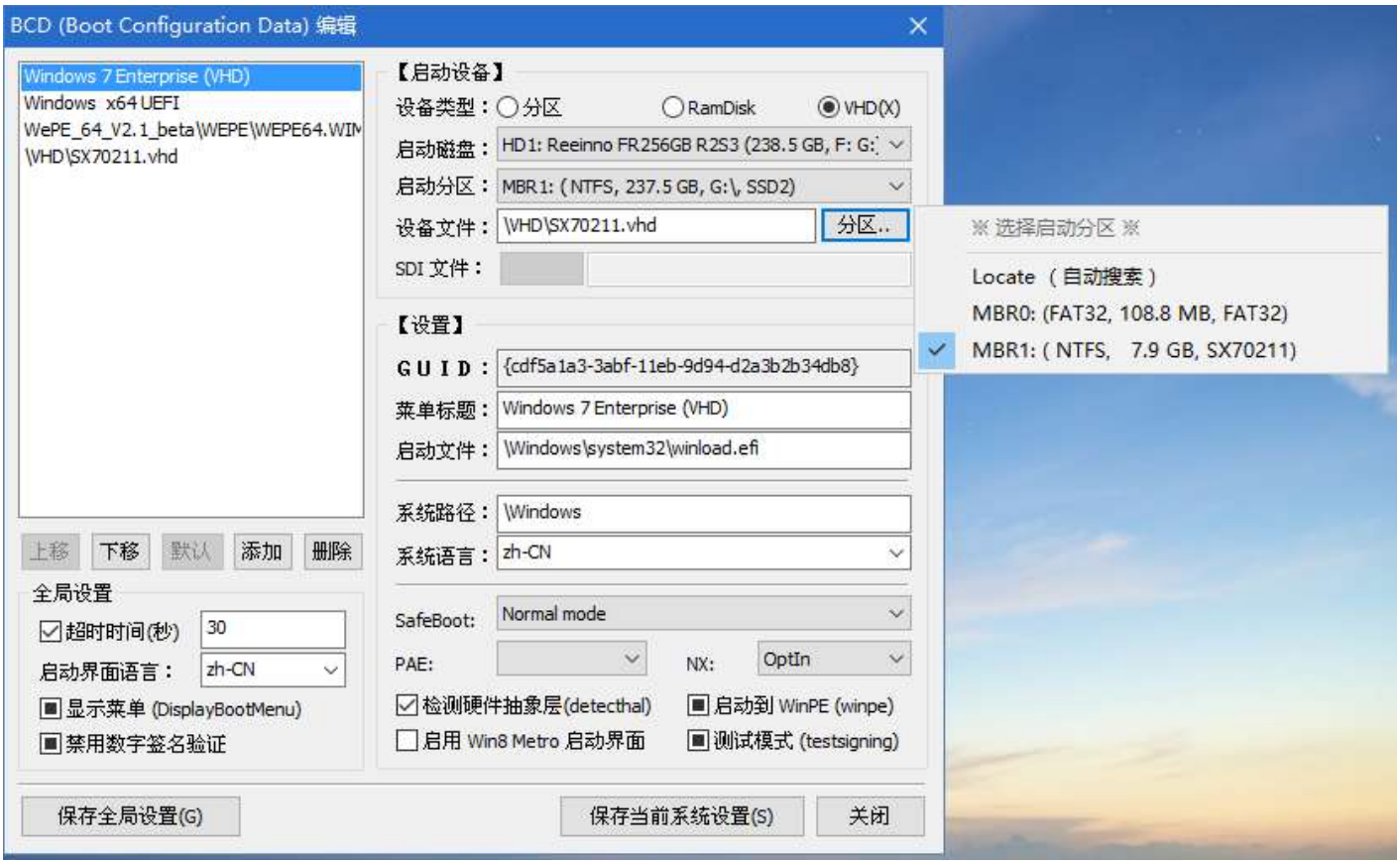
(1) If the original VHD is a single partition, you can enter PE, mount the VHD, use AOMEI Partition Assistant to divide a partition on the front of the hard disk, create a new FAT32 partition, format it, and then repair it with bcdboot Boot, bcdboot%windisk%:\windows /s %bootdisk%: /l zh-CN /f %bootmode%, pay attention to the drive letter where windows is located, and choose the drive letter of the partition where the NTFS of the VHD is located. There is a batch network disk.

Then delete the EFI and boot directories in the NTFS partition (you can change the name), otherwise, for WIN7, WIN8, grub4dos UEFI version will still appear "boot_image_handle not found" error when loading.

(2) After the partition is adjusted, adjust the BCD inside the VHD (FAT32/ESP for VHD)\EFI\Microsoft\Boot\BCD, manually select the boot disk and boot partition, the boot partition is the NTFS partition inside the VHD:



(3) After adjusting the internal boot partition of the VHD, adjust the external BCD of the VHD, otherwise the error 0xc000000f will be displayed when booting the VHD with bootmgfw.efi, (Main hard disk FAT32/ESP)\EFI\Microsoft\Boot\BCD:



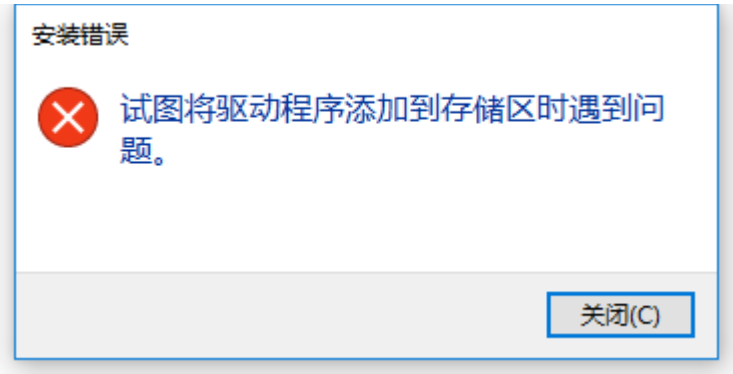
If your original VHD system is booted in BIOS mode and want to increase the UEFI boot mode, you can use the bcdboot command to repair the boot with one key after mounting the VHD, because I remember not to order the parameters, for convenience, I provide a "one key repair Guide the batch processing of .bat.

bcdboot %windisk%:\windows/s %bootdisk%: /l zh-CN /f %bootmode%

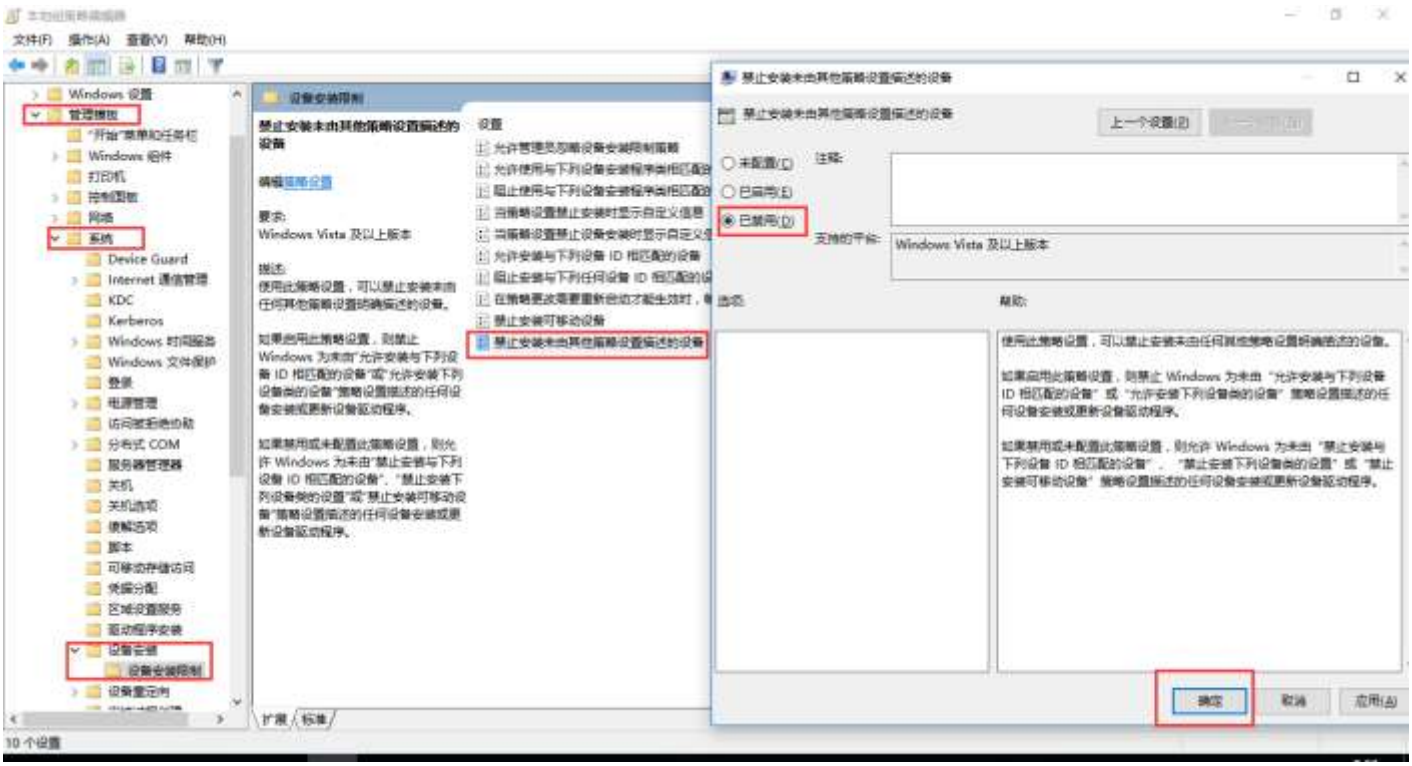
4. Installation of svbus driver

After the VHD system is installed, enter the VHD system and install the svbus driver. Because WIN10 has a strict driver signature mechanism, the driver signature verification in the BCD is turned off. The actual test is invalid (you may need to press F8 to turn off the signature verification, there is no actual test), the original open source version svbus did not pass the win10 signature. I contacted Zhuma and added the HT Srl certificate signature to the svbus driver.

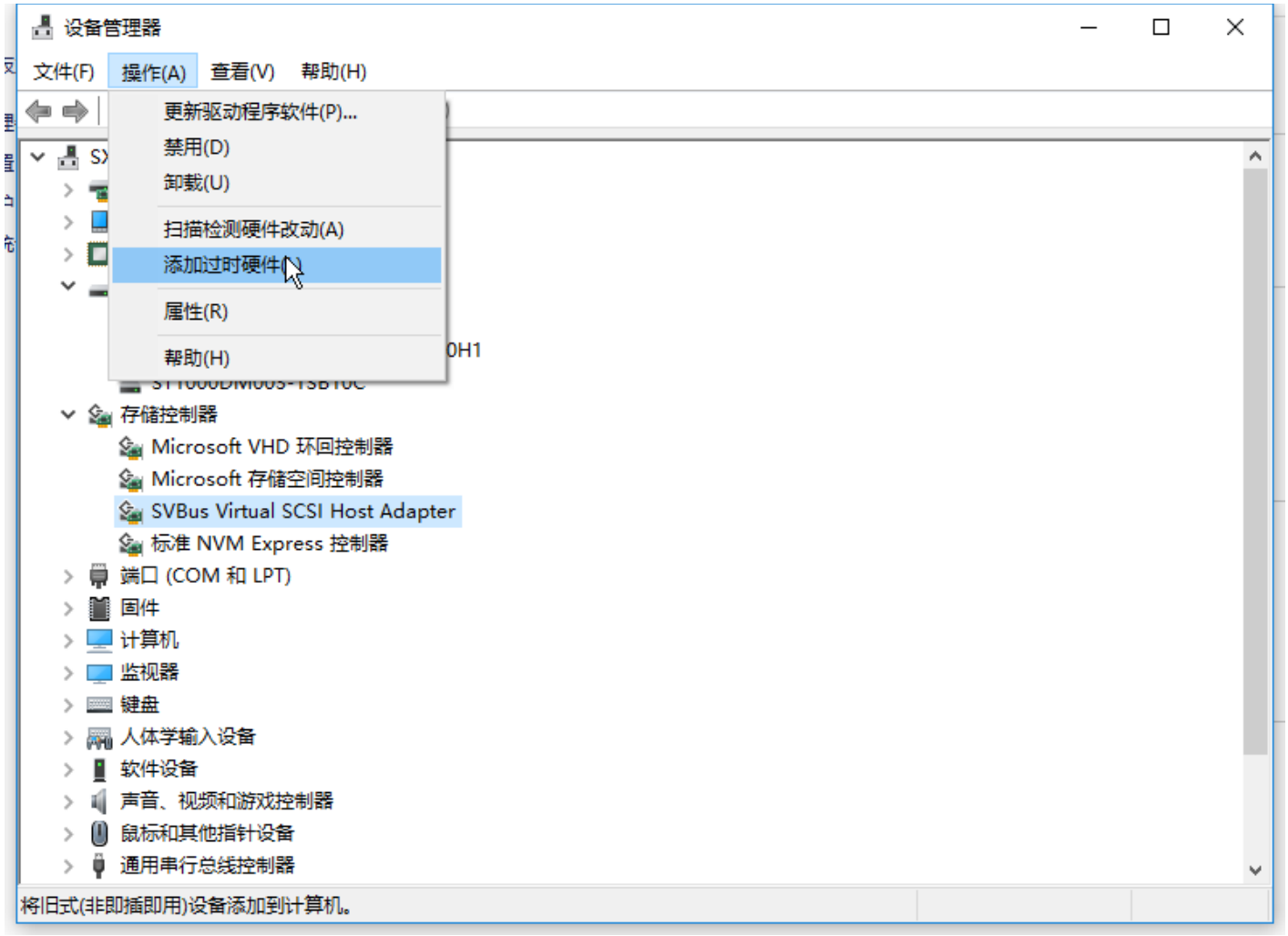
(1) First, double-click the EVRootCA.reg registry file to import it into the registry, otherwise the error "I encountered a problem when trying to add the driver to the storage area" may appear when svbus is installed.



(2) Right-click to install svbus.inf, select Install, you may be prompted "System Policy Group prohibits installation of this device", Baidu clicked, start-run, enter gpedit.msc to confirm-computer configuration-management template- —System—Device Installation—Device Installation Restriction—Prohibit the installation of devices that are not described by other policy groups (on the right), double-click and set it to "unconfigured" or "disabled"-OK. Restart the computer and install the signed svbus driver again.



(3) If the above method fails to install, use the method of adding obsolete hardware to add, right-click on my computer → properties → device manager → select this machine → operation → add obsolete hardware → install the hardware I manually selected from the list (advanced)→Install from disk→Select svbus.inf, trust the HT Srl certificate to confirm.



[Reminder] After svbus is installed, it is best to restart the computer to ensure that the driver takes effect.

5. VHD fragment elimination (not necessarily required)

After the VHD is installed, there may be fragments. If there are more than 32 fragments in the BIOS, it may not work; it is not verified under UEFI. The way to eliminate the fragments is not to mount the VHD. In the PE:

Method one, copy and paste the vhd, the copy is free of fragments, provided that the hard disk has enough free space for moving.

Method two, directly use wincontig to analyze and organize files. The prerequisite is that the hard disk has enough free space.

Generally speaking, the above method is effective for mechanical hard drives. It is not necessarily effective for SSD, because SSD has an intermediate layer of FTL logic conversion circuit, and SSD directly copies files into the grid, which should be continuous.

6. Copy the boot file and build the boot menu

The above network disk contains the directory structure of grub4dos and grub2 bootloader and its boot menu example.rar, unzip it to the ESP or FAT32 boot partition, and then modify it according to your own vhd path, the latest version of grub4dos and grub2 You can download the update from the official website.

(1) UEFI can use grub4dosUEFI as the primary boot. From grub4dosUEFI official website: http://grub4dos.chenall.net/categories/0-4-6a-for-UEFI/ , download the version after grub4dos UEFI2020-12-09, and put bootx64.efi in the (ESP or FAT32 boot partition))\EFI\boot\ directory, and use Bootice to add the UEFI boot sequence, pointing to (ESP or FAT32 boot partition)\EFI\boot\bootx64.efi (some motherboards can automatically add this UEFI boot sequence).

The default boot menu of grub4dos UEFI is the file \EFI\grub\menu.lst, and the file should be encoded in utf-8. The example is as follows:

timeout 5

default 0

graphicsmode-1 800

#graphicsmode-1 -1 -1 24:32

#graphicsmode-1 640:800 480:600 24:32 || graphicsmode -1 -1 -1 24:32

find--ignore-floppies --set-root /efi/grub/unifont.hex.gz

font/efi/grub/unifont.hex.gz

#splashimage/efi/grub/lt.jpg

colornormal=0x55ffff highlight=0xff00ff helptext=0xffff55 standard=0x55ffffborder=0xaaaaaa

#colornormal=0x07 highlight=0xE1 helptext=0x07 heading=0x02

setmenu--box x=4 w=60 y=6 h=9 l=2

setmenu--keyhelp=1 --lang=zh

setmenu--auto-num-on

setmenu--keyhelp=1=0x66ff00

setmenu--string=m=2=0x0000000000ffff="G4D maintenance menu"

setmenu--string=s=1=0x8800000000ffff="date&time=yyyy-MM-dd HH:mm:ss"

setmenu--timeout=90=2=0x88000000ffff

setmenu--hotkey -A [F4] commandline

title WIN10X64-SVBUS (/VHD/ltsb-svbus.vhd)

find --ignore-floppies --ignore-cd --set-root/VHD/SX10090329.vhd

map --mem --top /VHD/SX10090329.vhd (hd)

chainloader (hd-1)

(2) UEFI can also be booted with grub2. Wintoflash has made a lot of changes on the basis of the official grub2.04 to support map. The code has been open source https://github.com/a1ive/grub/releases/tag/latest , if the download is slow, you can download it from http://gitd.cc/. Put grubx64.efi under (ESP or FAT32 boot partition)\EFI\boot\ directory, and use Bootice to add UEFI boot sequence, pointing to (ESP or FAT32 boot partition)\EFI\boot\grubx64.efi.

The default boot menu of grub2 is the file \boot\grub\grub.cfg, and the file should be encoded in utf-8. The example is as follows:

```
set default=0

set fallback=1

set timeout=2

set pager=20

set grub_draw_border=1


#Set the menu font and background color

set menu_color_normal=white/black

set menu_color_highlight=white/blue


loopback -m ramdisk/boot/grub/unicode.xz

loadfont(ramdisk)/grub2/fonts/unicode.pf2

set locale_dir=(ramdisk)/grub2/locale

set lang=zh_CN

terminal_output gfxterm


#loadfont /boot/grub2/fonts/unicode.pf2

#set locale_dir=/boot/grub2/locale

#set lang=zh_CN

#set gfxmode=auto,800x600,1024x768

#terminal_output gfxterm


menuentry "0.Windows""/EFI/Microsoft/Boot/bootmgfw.efi" --hotkey=0 {

        search--no-floppy --set --file $2

        chainloader$2
```

}

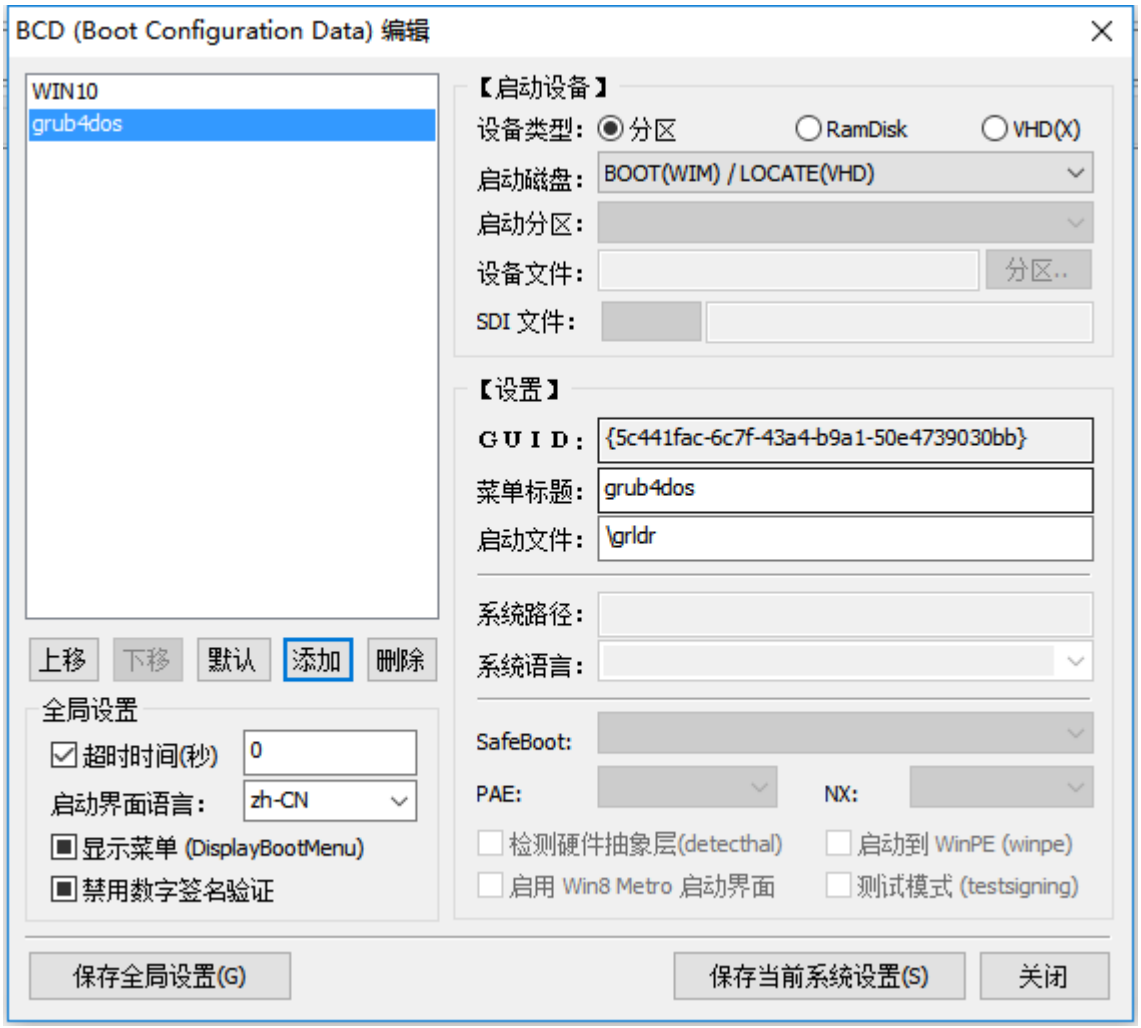
menuentry"UEFIltsb-svbus.vhd" "/VHD/UEFIltsb-svbus.vhd" {

search --no-floppy --set --file $2

map --mem --rt $2

}

(3) The grub4dosBIOS version can be used to boot under BIOS. Grub4dosBIOS version download address: http://grub4dos.chenall.net/categories/0-4-6a/ , take the commonly used boot process as an example, usually bootmgr→BCD→grldr→menu.lst of windows, extracted from the compressed package To the root directory of any partition, it is recommended to put it directly under the root directory of the boot partition, edit \boot\bcd with bootice, and add "real mode boot item grub", as shown in the figure below.



grldr will find and load the \menu.lst→\boot\menu.lst→\boot\grub\menu.lst file in turn. The \menu.lst file is recommended to be encoded in utf-8. The menu example is as follows:

```
graphicsmode-1 640:800 480:600 24:32 || graphicsmode -1 -1 -1 24:32

#foregroundFFFFFF

#background0000AD

find--ignore-floppies --set-root /EFI/grub/unifont.hex.gz

font/EFI/grub/unifont.hex.gz

#colorwhite/blue blue/yellow light-red/blue 10

#colornormal=0x07 highlight=0xF1 helptext=0x07 heading=0x02 standard=0x07 border=0x09

colornormal=0x07 highlight=0xE1 helptext=0x07 heading=0x02

timeout5

default0


title0 boot Windows by zhaohj

find--set-root --devices=h /ntldr || find --set-root --devices=h /bootmgr

map() (hd0)

map(hd0) ()

map--rehook

find--set-root --devices=h /ntldr || find --set-root --devices=h /bootmgr

#rootnoverify(hd0,0)

chainloader/ntldr || chainloader /bootmgr


title WIN10X64-SVBUS (/VHD/ltsb-svbus.vhd)

find --ignore-floppies --ignore-cd --set-root /VHD/ltsb-svbus.vhd

map --mem --top /VHD/ltsb-svbus.vhd (hd0)

map (hd0) (hd1)

map --e820cycles=-1
```

map --hook

root (hd0,0)

chainloader (hd0,0)/bootmgr

7. BIOS settings

In the previous installation system, we used windows bootmgr/bootmgfw.efi as the main boot. Microsoft's boot system can be "secure boot". The essence of "secure boot" is that Microsoft suppresses Linux and charges protection fees in order to strengthen its monopoly. Dirty means, grub4dos/grub2 does not pay Microsoft a protection fee, so it cannot be directly started via secure boot. Generally, it is recommended to disable secure boot in the BIOS settings. I randomly found a screenshot setting from the forum to illustrate the problem. Start → Secure Boot → Choose OtherOS as the operating system type. Some BIOSes directly have the disable option of Secure Boot, which is more direct. There are ways to bypass the secure boot in the forum. I feel that it is only a matter of time before being blocked by Microsoft. I don't have much research, so I skip it (ashamed).

8. Restart into RAMOS

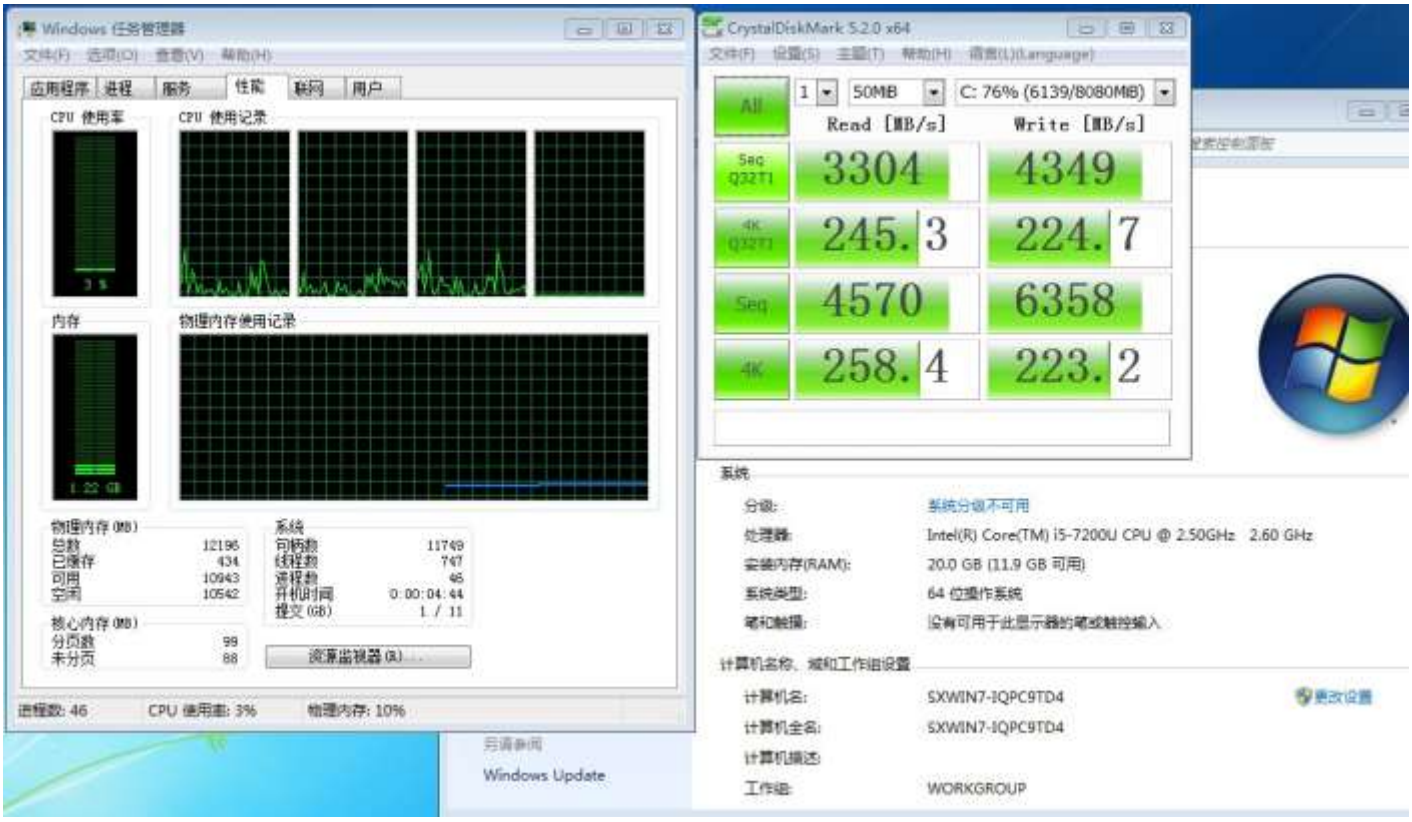
When the logo appears on boot, press the F12/F8/F9/F10 startup shortcut key to select bootx64.efi or grubx64.efi, and then select the VHD-RAMOS you want to start.

Shortcut keys for U disk startup items of computers of different brands
http://wuyou.net/forum.php?mod=viewthread&tid=410907

组装机主板		品牌笔记本		品牌台式机	
主板品牌	启动按键	笔记本品牌	启动按键	台式机品牌	启动按键
华硕主板	F8	联想笔记本	F12	联想台式机	F12
技嘉主板	F12	宏基笔记本	F12	惠普台式机	F12
微星主板	F11	华硕笔记本	ESC	宏基台式机	F12
映泰主板	F9	惠普笔记本	F9	戴尔台式机	ESC
梅捷主板	ESC或F12	联想Thinkpad	F12	神舟台式机	F12
七彩虹主板	ESC或F11	戴尔笔记本	F12	华硕台式机	F8
华擎主板	F11	神舟笔记本	F12	方正台式机	F12
斯巴达克主板	ESC	东芝笔记本	F12	清华同方台式机	F12
昂达主板	F11	三星笔记本	F12或F2	海尔台式机	F12
双敏主板	ESC	IBM笔记本	F12	明基台式机	F8
翔升主板	F10	富士通笔记本	F12		
精英主板	ESC或F11	海尔笔记本	F12		
冠盟主板	F11或F12	方正笔记本	F12		
富士康主板	ESC或F12	清华同方笔记本	F12		
顶星主板	F11或F12	微星笔记本	F11		
铭瑄主板	ESC或F11	明基笔记本	F9		
盈通主板	F8	技嘉笔记本	F12		
捷波主板	ESC	Gateway笔记本	F12		
Intel主板	F12	eMachines笔记本	F12		
杰微主板	ESC或F8	索尼笔记本	ESC		
致铭主板	F12	苹果笔记本	长按“option”键		
磐英主板	ESC				
磐正主板	ESC				
冠铭主板	F9				

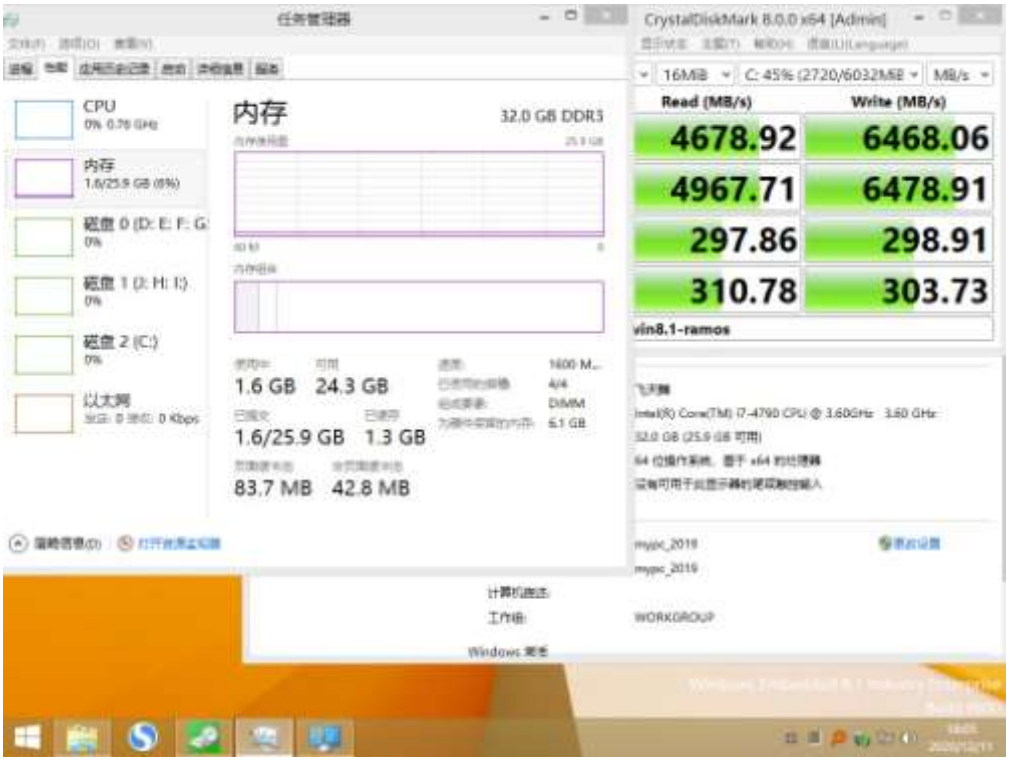
Svbus-UEFI-WIN10-RAMOS speed measurement:

g4e+UEFI-WIN7-RAMOS



g4e/grub2+svbus+win8.1RAMOS

32GB RAM = 6GB occupied by vhd + 1.6GB used by Windows + 24.3GB remaining available



9. The future technical outlook of UEFI-RAMOS

The boot principle of UEFI-RAMOS based on svbus is that grub4dos_UEFI or grub2_UEFI uses the EFI_RESERVED_MEMORY type map --mem --top xxx.vhd (hd0) to load vhd into the memory, and cooperate with the svbus driver to let windows recognize this memory disk, and from this memory disk Start above. This startup process is similar to the map--mem+firadisk/winvblock/svbus disk simulation startup of grub4dos_BIOS.

(1) There is still room for optimization and improvement of svbus drive speed. I tested about 4~5GB/S for continuous reading and writing of svbus memory disk, while primo6.3.1 drive can reach 16GB/S, which is 4 times that of svbus.

(2) RAMOS based on svbus divides the physical memory into three parts: the first part is grub4dos/grub2 emulation to the memory disk; the second part is the memory occupied by windows itself and its applications; the third part is the remaining available memory. The remaining part of the first part of the memory disk can only be used to install the program, and cannot be converted into the third part of the remaining available memory, causing waste; and the third part of the remaining available memory cannot be converted into the first part of the memory disk space, and I want to install the program on the RAMOS_C drive , The remaining space is too small.

Compare the RAMOS made by the primo driver under the BIOS. The first part and the third part of the memory can be transformed into each other. The memory is the hard disk, and the hard disk is the memory. You can build as much memory disk as you have. For example, if you have 32GB of memory, A 32GB memory disk can be created. After deducting the 1.5GB memory required for the operation of the windows itself, the C drive space of RAMOS and the remaining available memory can be converted into each other while keeping the total 32-1.5=30.5GB unchanged. C Disk big can install and test many softwares to make full use of memory.

Future prospects (unrealized): grub4dos_UEFI or grub2_UEFI's map xxx.vdf (hd0) loads vdf to virtual disk C: disk, and windows starts from this virtual disk C: disk and then loads, such as primo memory disk drive, which is driven by primo xxx.vdf is loaded to the memory disk such as R: disk. In this xxx.vdf, the mounted device of C: disk and R: disk are interchanged. Windows can continue to boot from the C: disk of this memory disk. This primo driver is ready-made and speed quickly. The problem with this path is that in the first step, the map xxx.vdf (hd0) of grub4dos_UEFI or grub2_UEFI loads vdf to the virtual disk C: disk. Windows does not recognize the virtual disk C: disk. It is stuck in the BCD, and the subsequent boot process Unable to continue, this still requires a great god who understands the operating system startup process and understands the principles of RAMOS startup to do further research.

In the Svbus driver solution, the memory disk is created by grub4dos/grub2, while the primo driver solution memory disk is created by the primo driver. In comparison, the Svbus driver program has been implemented, and the primo driver program seems more promising, because the driver Already, all is ready except for the opportunity,to let windows recognize the virtual disk emulated by grub4dos_UEFI or grub2_UEFI map xxx.vdf (hd0), and boot the BCD normally.

10. Special thanks

tinybit-the initiator of the grub4dos project

Bean-grub4dos developer

Chenall-grub4dos developer

2011yaya2007777-grub4dos developer

Wintoflash-grub2 developer

Pauly-author of bootice

Kai Schtrom-author of the open source SVBus driver, from Germany.

Sunsea-svbus source code level interpretation support.

Zhuma 12345678-super version, made a signature for the svbus driver to ensure smooth installation.

Sinoxer——USB3_Drivers_Smart_Install_For_Win7

Red Sakuragi-ceomsx (drive injection tools such as USB3 and NVME)

alacran-svbus-UEFI-RAMOS a lot of testing and feedback, from Mexico.

Liuzhaoyzz-super version, grub4dos/grub2+svbus-UEFI-RAMOS a lot of testing and feedback.

Wuwuzz-grub4dos/grub2 problem feedback.

t5481194 moderator-hot keys for starting computers of different brands.

Etc., etc……