

2018\_11\_22

//-----

The latest "standard" SSD layout is as follows.

The "BIOS boot partition" may well not be required as grub2 data is probably located in sectors 34 -&gt; 2047

The separate /boot partition (sda3) may well not be used.

[root@gtx:~]\$ gdisk -l /dev/sda

Found valid GPT with protective MBR; using GPT.

Disk /dev/sda: 250069680 sectors, 119.2 GiB

Model: Voyager GTX

Main partition table begins at sector 2 and ends at sector 33

Partitions will be aligned on 2048-sector boundaries

Number	Start (sector)	End (sector)	Size	Code	Name				
1	2048	4095	1024.0 KiB	EF02	BIOS boot partition	1MiB	none	????	
2	4096	1028095	500.0 MiB	EF00	EFI System	500MiB	vfat	/boot/efi	
3	1028096	2097152	522.0 MiB	8300	Linux filesystem	1GiB	ext4	/boot	
4	2099200	18876415	8.0 GiB	8200	Linux swap	8GiB	swap		
5	18876416	134219775	55.0 GiB	8300	Linux filesystem		ext4	/	
6	134219776	250069646	55.2 GiB	8300	Linux filesystem		ext4	spare	

//-----

If the UEFI does not "see" the SSD then check the status of the Protective MBR boot flag using gdisk/fdisk

To CLEAR the flag gdisk /dev/sda p ; v ; x ; n ; w ;

To SET/CLEAR the flag fdisk /dev/sda M ; i ; a ; i ; w ; q ;

//-----

USB SSD TRIM - As root

/home/ja/bin/wiper.sh --verbose --commit /dev/sda1

//-----

To discover how the machine was booted

1. Check grub menu entries for linux16 or linuxefi

2. When booted in UEFI mode efibootmgr will provide relevant information

//-----

To ensure that the device will run on the widest range of machines

dnf install dracut-config-generic To force a generic initrd

To force an existing kernel to use a "fully configured" initramfs file then

dracut --regenerate-all --force

//-----

Creation of an SSD (USB or SATA) device that will boot on both BIOS and UEFI based machines.

References:

<https://blog.heckel.xyz/2017/05/28/creating-a-bios-gpt-and-uefi-gpt-grub-bootable-linux-system/><https://superuser.com/questions/801515/is-a-hybrid-linux-usb-stick-for-uefi-legacy-bios-possible>

It is now known how the fedora installer determines which boot mechanism to use.

The type of install is determined by the way the distribution stick is booted.

Once a bootable installation has been achieved (BIOS or UEFI) the "alternative" boot mechanism can be installed. Both of the possibilities described below have been well tested.

Triple check the correct device name at all stages, /dev/sda is assumed below.

Case 1: The machine has booted from a BIOS install	Install the UEFI boot loader
mkfs -t vfat /dev/sda2	if required (double check sda2)
mount /dev/sda2 /boot/efi	(double check sda2)
dnf [re]install grub2-efi-x64 shim-x64 efibootmgr	this should populate /boot/efi/EFI/fedora
grub2-mkconfig -o /boot/efi/EFI/fedora/grub.cfg	create the UEFI .cfg file*
geany /boot/efi/EFI/fedora/grub.cfg	change linux16 > linuxefi, initrd16 > initrddefi*
grub2-install --target=x86_64-efi /dev/sda	This is unnecessary - DO NOT USE - why?
Case 2: The machine has booted from a UEFI install	Install the BIOS boot loader
mount /dev/sda3 /boot	probably not required
dnf [re]install grub2-pc	only installs 3 files
grub2-mkconfig -o /boot/grub2/grub.cfg	create the BIOS .cfg file*
geany /boot/grub2/grub.cfg	change linuxefi > linux16, initrddefi > initrd16*
grub2-install --target=i386-pc /dev/sda	install grub on the MBR (double check sda)
double check that /boot/grub2/grub.cfg is not over written by grub2-install	

\*When a kernel update occurs the appropriate grub.cfg may require re-generation.

This has been tested when booted using both UEFI &amp; BIOS. No changes are necessary as both grub.cfg files are updated and include the correct entries for linux[efi | 16] and initrd[efi | 16].

<https://forums.opensuse.org/showthread.php/509106-linux-or-linux-efi-commands-generated-by-grub2-mkconfig>

"When you use "linuxefi" it loads the kernel in a different way,

and does the signature checking that secure-boot requires."

F29 Nov 2018: (gtx1 memory stick)

BIOS: /boot/grub2/grub.cfg linux &amp; initrd

EFI: /boot/efi/EFI/fedora/grub.cfg linux &amp; initrddefi [/dev/sdb2 is mounted at /boot/efi]

This changes after a dnf kernel update!

BIOS: /boot/grub2/grub.cfg linux &amp; initrd [for the latest kernel]

EFI: /boot/efi/EFI/fedora/grub.cfg linuxefi &amp; initrddefi [for the latest kernel]

//-----

If you want to manually provide redundant EFI system partitions (no RAID1), all you need to do is run efibootmgr to add a second entry. You don't need to install grub2-efi-modules or run grub2-install. Use: efibootmgr -c -w -L 'CentOS Linux' -d /dev/sdb -p 1 -l 'EFI\centos\shim.efi'

... where the option to -d is the name of the drive containing the EFI system partition and the

option to -p is the EFI system partition number.

For reference, that command will appear in /var/log/anaconda/program.log after any normal installation.