UEFI - informacje

Położenie: (nie dotyczy)

Ogólnie

UEFI zastępuje BIOS, wpisy w MBR oraz zasadność stosowania GRUB-a. Do zalet płyt głównych z funkcją UEFI (*Unified Extensible Firmware Interface* – posiadają chip NVRAM) należy możliwość przeprowadzenia zdalnej diagnostyki bez użycia systemu operacyjnego, większa ilość pamięci (BIOS ma tylko 1MB), szyfrowanie i zdalne / lokalne uwierzytelnianie (*secure boot*), niezależność od architektury procesora.

Fast Startup w Windows

Od *Windowsa 8*, przy zamykaniu systemu, stosuje się tzw. *hybrid shutdown* (funkcja nazywa się „*Fast Startup*"), co skutkuje później szybszym uruchamianiem się systemu. W czasie zamykania systemu, zapisywana jest sesja jądra oraz sterowniki urządzeń do pliku *hiberfil.sys*.
Aby włączyć lub wyłączyć „*Fast Startup*": *Panel sterowania / Opcje zasilania / Wybierz działanie przycisków zasilania / Zmień ustawienia, które są obecnie niedostępne / Włącz szybkie uruchamianie*

Secure Boot

Funkcja blokuje bootowanie komputera z zewnętrznych źródeł: DVD, pendrive, sieć.

UEFI

UEFI tworzy alternatywę wobec MBR – jest nią GPT (*Guid Partition Table*), która likwiduje limit partycji podstawowych (w MBR było do 4 partycji na dysk), a także likwiduje limit ich rozmiaru (w MBR było to do 2TB, zaś w GPT aż do 8ZB). UEFI jest wstecznie kompatybilne i potrafi bootować w stylu BIOS z dysków mających partycje MBR. Wykorzystuje do tego CSM (*Compatibility Support Module*), które symuluje środowisko BIOS.
Uwaga: Aby udało się zainstalować UEFI, system musi być już zbootowany w stylu UEFI. Przed chrootowaniem, należy sprawdzić:
modprobe efivars (lub *efivarfs*)
oraz
mount | grep efivarfs
efivars on /sys/firmware/efi/efivars type efivarfs (rw,nosuid,nodev,noexec,relatime)
Uwaga: Musi być parametr „rw", nie zaś „ro". Jeśli będzie „ro", należy odmontować i zamontować ponownie:
mount -o remount,rw -t efivarfs efivarfs /sys/firmware/efi/efivars

Gdisk

Zaleca się stosowanie do partycjonowania narzędzia *gdisk* (pakiet *gptfdisk*), zamiast *fdisk*.
gdisk /dev/nvme0n1 (m.in. sprawdza czy jest obecna partycja MBR czy też GPT)

Wszelkie partycje powinny zaczynać się co najmniej od 2048 sektora. Partycje „*BIOS boot partition*" (co najmniej 1024KiB, typ 0xEF02), a także „*EFI System*" (8MiB, typ 0xEF00) mogą być utworzone w dowolnym miejscu po tym sektorze nawet na końcu dysku.

Konfiguracja jądra

Aby Linux był obsługiwany w ramach EFI należy uaktywnić w jądrze opcje:
Enable the block layer / Partition types *--->*
[*] PC BIOS (MSDOS partition tables) support
[*] EFI GUID Partition support

Processor type and features *--->*
[*] EFI runtime service support
[*] EFI stub support
[ ] EFI mixed-mode support (opcjonalnie, tylko jeśli wiesz, co robisz)
[*] Built-in kernel command string (konieczne!)
(root=/dev/sdaX) lub w przypadku GPT (root=PARTUUID= [numerID wyświetlany po wydaniu komendy „blkid" lub „gdisk /dev/nvme0n1" (z pakietu „gptfdisk"), a potem komendy „p" oraz „i"], w moim przypadku: 45dd16cb-4f52-46ef-834c-0b5b35b65bd6. Należy pamiętać, że ma to być numer partycji, a nie dysku... ma to być PARTUUID, a nie UUID... i że dotyczy to głównego systemu root /, a nie /boot)
[ ] Built-in command line overrides boot loader arguments (opcjonalnie, tylko jeśli wiesz, co robisz)

Device Drivers */* ***Graphics support*** *--->*
<> Frame buffer devices --->*
[*] EFI-based Framebuffer Support

File systems */* ***Pseudo filesystems*** *--->*
-- /proc file system support*
[*] /proc/kcore support
[*] Tmpfs virtual memory file system support (former shm fs)
[*] Tmpfs POSIX Access Control Lists
-- Tmpfs extended attributes*
[*] HugeTLB file system support
<> Userspace-driven configuration filesystem*
<> EFI Variable filesystem*

Następnie mountujemy partycję EFI w tym miejscu:
 # mount /boot/efi
W pliku */etc/fstab* powinien być wpis:
/dev/nvme0n1p1 /boot/efi vfat noauto,noatime 1 2
Jeśli nie mamy partycji EFI, musimy ją stworzyć (patrz: *gdisk lub gparted*) i sformatować:
 # mkdosfs -F 32 -n efi-boot /dev/nvme0n1p1 (z pakietu dosfstools z flagą „compat")
Instalujemy narzędzie do bezpośredniej zmiany wpisów w UEFI:
 # emerge -vp efibootmgr
Przeglądamy tablicę EFI:
 # efibootmgr -v
Uwaga: Obraz „*EFI/BOOT/bootx64.efi*" uruchamiany jest jako ostatni, gdy żaden inny nie istnieje lub nie będzie działał (jest to tzw. *fallback*). Jeśli chcemy, aby to Gentoo było *fallback*, to po prostu kopiujemy tu jądro i nadajemy mu nazwę „*bootx64.efi*".

 # cp /boot/kernel-6.1.6-2023-01-20 /boot/efi/EFI/gentoo/gentoo.efi
Przykład utworzenia nowego wpisu:
 # efibootmgr -c -d /dev/nvme0n1 -p 1 -L "*Gentoo***" -l "***\EFI\gentoo\gentoo.efi***"** (to jest małe „L")
Przykład usunięcia wpisu:
 # efibootmgr -b 2 -B

Opcje:
--create (-c) – tworzenie nowego wpisu / rekordu;
--part (-p) – numer partycji na której znajduje się EFI;
--disk (-d) – dysk, na którym znajduje się EFI (Uwaga! Dysk! Nie partycja!);
--label (-L) – treść wpisu / rekordu bootowania;
--loader (-l) – ścieżka do obrazu jądra systemu (stosuje się ukośniki „\")
--bootnum (-b) – numer rekordu, np. *boot0002* (czyli 2)
--delete-bootnum (-B) – usuwa wskazany rekord
--timeout (-t) – ustawia czas oczekiwania na reakcję użytkownika

UEFI Script

To kolejny menedżer tekstowy UEFI. Aby z niego skorzystać, należy ściągnąć paczkę „*Shell2.zip*”, która zawiera binarną wersję programu „*UefiShellX64.efi*”. Program kopiujemy na partycję UEFI:
cp UefiShellX64.efi /boot/efi/EFI/boot/shellx64.efi
Program można wywołać w opcjach BIOS, gdzie znajduje się polecenie:
Save & Exit \ Launch EFI Shell from filesystem device...
Niestety, BIOS poszukuje pliku o nazwie „*Shell.efi*”, a nadanie mu takiej nazwy niczego nie zmienia. Program zawiera kilka komend (przykłady):
Shell> **bcfg ?**
Shell> **bcfg boot dump -b** *(pokazuje wpisy)*
Shell> **bcfg boot rm 3** *(usuwa wpis nr 3)*
Shell> **bcfg boot add 3 FS0:\EFI\ubuntu\ubuntu.efi "Ubuntu"** *(dodaje wpis)*
Shell> **mkdir FS1:\EFI\gentoo**
Shell> **cp jakiśPlik.efi miejsce**
Shell> **edit jakiśPlik.txt**

Usuwanie wpisów UEFI w Windows

C:\Windows\system32**diskpart**
DISKPART> **list disk**
DISKPART> **select disk 0**
DISKPART> **list part**
DISKPART> **select part** *numerPartycjiUEFI*
DISKPART> **assign letter-M** *(czyli partycję UEFI oznaczamy jako dysk logiczny M, który powinien pojawić się w menedżerze plików)*
W osobnym oknie uruchom jako administrator *PowerShell*:
PS C:\Windows\system32> **cd M:**
PS M:\> **dir**
PS M:\> **cd EFI**
PS M:\> **dir**
PS M:\> **remove-item** *nazwaSystemuDoUsunięcia*
PS M:\> **exit**

Wracamy do diskpart:
DISKPART> **remove letter-M**
DISKPART> **exit**

Problemy i tipsy

Nie wykrywa bootowalnej partycji po restarcie

Należy ponownie sformatować partycję z EFI, usunąć wszystkie wpisy i ponownie je wprowadzić.

Ostatnia aktualizacja: 24 stycznia 2023.