

Multiboot UEFI-vel, Windows 10-zel és további disztrókkal

Írta: Frank Denissen

(Megjelent: FullCircle Magazine 102. sz. - 2015. november)

A terv

Van egy PC-m, 1 TB-os HDD-vel, 250 GB-os SSD-vel és egy CD-DVD meghajtóval. Windows 10-et és négy Linux disztribúciót akarok telepíteni rá UEFI-t és secure boot-ot használva.

Korábban sok problémám volt a multiboot-os Linuxokkal, mert a boot loader-t (**Grub2**) és könyvtárát (**/boot**) meg kellett osztani. A gond az, hogy minden egyes disztró eltérő grub2 verziót telepít. Ez könnyen odavezethet, hogy a PC nem hajlandó elindulni, mivel egy frissítés után a disztró által használt **/boot** könyvtár és a merevlemezre telepített boot loader között inkonzisztencia léphet fel.

Ezt elkerülendő a [Full Circle Magazine 88-as számában](#) megjelent „Hogyan: GRUB2 és multiboot 4. rész” ([a kiadvány magyarul itt](#)) cikkben leírt kétlépcsős boot-olást fogom használni. A központi boot loader önálló partícióra kerül az adataival, ahol a felhasználó menüből választhat disztrót. Választás után a központi boot loader a kiválasztott disztró saját boot loader-ét indítja el. Az önálló boot loader saját partícióján tárolja az adatokat, és egy másik menüt ad fel a felhasználónak, hogy kernel verziót és esetenként boot opciókat válasszon. A fő grub2 chainloader modulja arra való, hogy egy másik boot loader-t indítson.

Emellett LVM-et (Logical Volume Manager: logikai kötetkezelő) használok, amennyire lehet. Az LVM-ben egy, vagy több fizikai partíciót, ún. kötetcsoportokat, és a kötetcsoportokban logikai partíciókat lehet felállítani. A logikai partíciókat logikai köteteknek hívják az LVM nyelvén. Nagyon könnyű a logikai partíciók méretét igény esetén megváltoztatni. Másik nagy előnye, hogy a logikai kötet neve nem változik meg (formázásnál), így nyugodtan lehet megcímezni egy **lv**-t azaz logikai kötetet a kötetcsoportban, hdvg-ben mint **/dev/mapper/hdvg-lv**, vagy **/dev/hdvg/lv**.

Én egy kötetcsoportot „**hdvg**” készítek a **merevlemezre** (hd) a disztrók változó adatai számára és egy másik „**ssdvg**”-t a **szilárdtest-meghajtón** (SSD). A Windows 10-et a merevlemez első felére telepítem.

A központi boot loader-nek fizikai partíció kell (a boot loader nem beszél az LVM-et), és azt az SSD-re fogom rakni. A disztribúciók között megosztott további partíciók a következők:

- egy fizikai partíció cserehelynek (swap) és
- egy logikai partíció a Linux indításakor törlődő fájlok számára (**/tmp**).

Mindkét partíció a merevlemezen lesz, mivel változó adatokat tartalmaznak,

Minden disztrónak készítek egy saját:

- fizikai partíciót az SSD-n a boot loader-e (**/boot**) számára,
- logikai kötetet az SSD-n a disztró root fájlrendszere számára (**/**) és végül
- logikai kötetet az állandó adatoknak (**/var**) a merevlemezen.

Szoktak még önálló partíciót készíteni a felhasználói adatoknak (**/home**), de én nem teszem, mert van egy NAS-em (Network Access Storage: hálózati elérésű tároló), ahová csatolom a kedvenc disztróm (Kubuntu) home könyvtárait. A többi disztró csak kísérleti célú, így sosem tartalmaznak fontos adatokat.

1. lépés: felkészülés a Windows 10 telepítésére

Előbb fizikailag válaszd le a szilárdtest-meghajtót az alaplapról, hogy a Windows 10 mindenképpen a merevlemezt használja.

A Windows 10-et USB-kulcson árulják, így dugd be azt az egyik USB-csatlakozóba és boot-olj be a PC-be. Lépj be a BIOS-ba az UEFI és a secure boot engedélyezéséhez. Az alaplapomnál a Del billentyűt kell közvetlenül a PC boot-olása után néhányszor lenyomni.

2. lépés: készítsd elő a BIOS-t

A gond evvel az, hogy a pontos menete az alaplap típusától függ, vagyis nagy valószínűséggel kísérletezned kell, hogy az alaplapod beállításait megtaláld. Az enyém egy **ASUS A88X-Plus**. Ennél a típusú alaplapnál az „**Advanced Mode**”-ba kell belépni, és ezután kiválasztani a „**Boot**” fület.

Az **UEFI engedélyezéséhez** lépj a „**Compatibility Support Mode**”-ba és ezeket állítsd be:

- “**Launch CSM**” Enabled,
- “**Boot Device Control**”-t “UEFI only”-ra,
- “**Boot from storage devices**”-t “Both, UEFI first”-re és
- “**Boot from PCIe/PCI expansion devices**”-t “UEFI drive first”.

A **secure bott engedélyezéséhez** állítsd a „**Secure boot**”-ot „**Windows UEFI mode**”-ra.

Állítsd be az USB-kulcsot az első boot-eszköznek. Nekem az „UEFI: KDI-MSFTWindows10”-t kellett választanom. Ha több opció is van, akkor a UEFI változatot válaszd. Mentsd a beállításokat és lépjél ki.

3. lépés: telepítsd a Windows-t

A PC most újraindul és elkezdődik a Windows 10 telepítése.

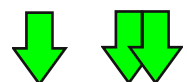
A telepítő felajánlja a merevlemez négy partícióra osztását. A legnagyobb, 500 GB-s partíció méretét úgy csökkentettem, hogy a jövőben létrehozandó Linux-partícióknak elég hely legyen.

A telepítés során, meg kell válaszolnod néhány kérdést, de végül a PC újraindul Windows 10-zel.

A multiboot-os környezetben **ki kell kapcsolni a gyors indítást (fast boot)**, hogy elkerüld a Windows 8.1, vagy 10 sérülését.

Menj a „**Vezérlőpult** → **Rendszer** → „**Bekapcsolási opciók**”-hoz és válaszd „**A bekapcsolási opciók kiválasztása**”-t. Kattints a „A jelenleg elérhetetlen beállítások változtatása”-ra és végül **távolítsd el a jelzést a „Gyors indítás bekapcsolása (ajánlott)”** elől. Mentsd a változásokat és indítsd újra a PC-t.

Most távolítsd el a Windows 10-es USB-kulcsot.



4. lépés: telepítsd a Kubuntu-t

Töltsd le a Kubuntu 14.04 LTS 64-bit iso képfájlt és írd ki DVD-re, azt helyezd be a CD/DVD meghajtóba és kapcsolj ki a gépet.

Csatlakoztasd vissza a szilárdtest-meghajtót az alaplaphoz és indítsd el a PC-t.

Menj be a BIOS-ba és válaszd ki a DVD-t indító eszköznél. Nekem az „UEFI: P3 TSTST corp CDDVDW SH-224 DB”-t kellett. Mindig az UEFI-változatot válaszd, ha több opció van. Mentsd a beállításokat és lépj ki.

Előbb válaszd a „Start Kubuntu”-t, várj egy kicsit, amíg kiválaszthatod a nyelvet és „Kubuntu telepítése”-t. A „Lemez beállítása”-nál a „Telepítés típusa” legyen „Kézi”.

Banyek: a telepítő nem enged LVM kötetcsoporthoz és logikai köteteket készíteni, de képes dolgozni velük, ha már előre készen vannak.

Így most választanod kell:

vagy elfelejted az LVM-et és csak fizikai partíciókat hozol létre,
vagy a Kubuntu live-ba belépsz és ott hozod létre a fizikai és logikai partíciókat.

Én az utóbbi opciót választom.

5. lépés: partíciók és logikai kötetek létrehozása Kubuntu live alatt

Zárd be a telepítőt és automatikusan elindul a Kubuntu live.

Most **nyiss egy terminált**: az „Alkalmazások → Rendszer” alatt „Konsol terminál”-nál találsz.

Telepítsd a gparted-et a következő paranccsal:

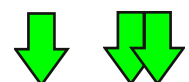
```
sudo apt-get install gparted
```

Indítsd el a gparted-et:

```
sudo gparted
```

Válaszd ki a megfelelő lemezt: a méretből kitalálhatod, hogy melyik a merevlemez és melyik a szilárdtest-meghajtó. Mindenképpen jegyezd meg az általad létrehozott partíciók azonosítóit (például: /dev/sda7).

A merevlemezen már van partíciós tábla. **Jelöld ki a lefoglalatlan területet** és válaszd „Partíció → Új”-at, hogy létrehozz egyet: a **mérete legyen 1 GB** és a fájlrendszerének válaszd a **swap**-ot. Ezt használja majd cseréhelyként. **Ismételd meg a lépéseket további partíciók létrehozásához**, lvm2pv fájlrendszerrel a teljes lefoglalatlan területre. Nyomd le az „Alkalmaz” gombot.



Megjegyzés: a partíciókat nem kell formázni: hagyd a telepítőre.

Az **SSD-meghajtón** még nincs partíciós tábla. Válaszd ki az eszközt és „Eszköz → **Partíciós tábla létrehozása...**”-t. A „Szakértői beállítások”-nál a **típust** msdos-ról váltsd **gpt-re**, majd **Alkalmaz**.

Most készíts 6 partíciót:

egy 100 MB-ost ext2 fájlrendszerrel a **központi boot loader** számára, **négy 1 GB**-ost szintén ext2 fájlrendszerrel **az egyedi boot loader-ek** számára, és a **fennmaradó** partíció egy **lvm2pv** fájlrendszer lesz a teljes lefoglalatlan területet felhasználva.

Nyomd le az „**Alkalmaz**” gombot. **Zárd be a gparted-et és lépj vissza a terminálba.**

Add hozzá a merevlemezen éppen most létrehozott lvm2pv fájlrendszerű fizikai partíciót (esetemben **/dev/sdb6**-nak hívják) az LVM-hez:

```
sudo pvcreate /dev/sdb6
```

Hozd létre a **hdvg** kötetcsoporthoz és kapcsold hozzá a partíciót:

```
sudo vgcreate hdvg /dev/sdb6
```

Készítsd el a **var1** 30 GB-s logikai kötetet a hdvg kötetcsoporthoz:

```
sudo lvcreate -n var1 -L 30G hdvg
```

Ezen a módon készítsd el a **var2**, **var3** és **var4** logikai köteteket és egy 10 GB-s **tmp** logikai kötetet.

Most készítsd el az **ssdvg kötetcsoporthoz** a szilárdtest-meghajtón és készíts négy 20 GB-os logikai kötetet **root1**, **root2**, **root3** és **root4** néven.

Lépj vissza a telepítőbe a „**Kubuntu telepítése**”-re kattintva az asztalon.

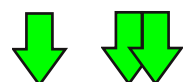
6. lépés: a Kubuntu telepítőjének újraindítása

Válaszd ismét a „**Kézi**” telepítési típust a lemez beállításánál.

Használd központi boot partícióként a **/boot/central**-t és az első saját boot partícióként a **/boot**-ot. Mindkét esetben válaszd az ext2-t fájlrendszerként.

Legyen a **/dev/ssdvg/root1** a **/**, a **/dev/hdvg/var1** a **/var** és a **/dev/hdvg/tmp** a **/tmp**, mindegyik ext4 fájlrendszer. Végül a **swap** partíciót használd cserehelyként. **Hagyd a telepítőt a partíciókat formázni.**

Folytasd a telepítést. **Telepítsd a boot loader-t a szilárdtest-meghajtóra** és végül indítsd újra a PC-t.



Ha a Windows töltődne be, lépj be a BIOS-ba és gondoskodj arról, hogy az Ubuntu legyen az első boot-eszköz. Nekem az „Ubuntu (P4 : WDC WD10EAVS-00D7B1)”-t kellett választanom.

Most már képesnek kell lenned a Windows és a Kubuntu elindítására a grub menüjével.

7. lépés: állítsd be a Linuxot az SSD használatára

Megjegyzés: ennél a lépésnél némi változtatást kell végrehajtani a konfigurációs fájlokban. Óvatosan ezzel: mindig készíts másolatot az eredeti fájlról (`sudo cp config-file config-file.bak`) és írd bele megjegyzéseket a neveddel, dátummal és a változtatás okával. Így könnyen visszacsinálhatod a változásokat, vagy megtalálhatod az összes módosított fájlt. Használd a live DVD-t javításra, ha valami elromlana.

Csökkentsd a szilárdtest-meghajtóra írások számát, amennyire lehet, hogy meghosszabbítsd az élettartamát.

A fájlrendszerek alaphelyzetben feljegyzik a fájlok és könyvtárak elérésének idejét. Ezt az információt szinte sosem használod, tehát biztonságos ennek a tulajdonságnak a kikapcsolása. Ezt legalább a szilárdtest-meghajtóra meg kell csinálni, de nem okozol semmi kárt, ha a merevlemezre is megteszed, az elérés sokkal gyorsabb lesz.

Szerkeszd meg az `/etc/fstab`-ot root-ként és add a „`noatime`”-ot a partíciók, vagy logikai kötetek opciós mezőjéhez (ne feledd a vesszőt).

Például:

```
UUID=8482863b-d04e-40d2-be10-f5f3df88b8cd / ext4 errors=remount-ro 0 1
UUID=f65f89ac-b2b0-4345-949a-6965e3513db3 /boot ext2 defaults 0 2
```

erre változik:

```
UUID=8482863b-d04e-40d2-be10-f5f3df88b8cd / ext4 errors=remount-ro,noatime 0 1
UUID=f65f89ac-b2b0-4345-949a-6965e3513db3 /boot ext2 defaults,noatime 0 2
```

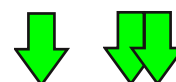
Ellenőrizd a megfelelő szintaxist úgy, hogy futtatom: `sudo mount -a`.

A második módosítás, a trim-parancs a boot idején fusson és ne cron feladatként. Szerkeszd a `/etc/rc.local`-t root-ként és abban add az „`fstrim -v <partíció>`” parancsot az egyes partíciókhoz, amik a szilárdtest-meghajtón vannak, amit sokszor a disztró ír. Én hozzáadtam:

```
fstrim -v /
```

Szerkeszd az `/etc/cron-weekly/fstrim`-et root-ként és rakj egy „`#`”-t az „`fstrim-all`” elé: Például:

```
#exec fstrim-all
```



Ha a cache partíció a szilárdtest-meghajtón van, csökkenteni kell a cache Linux általi használatának számát úgy, hogy **/etc/sysctl.conf**-hoz, root-ként szerkesztve hozzáadod a következő sorokat:

```
vm.swappiness=1
vm.vfscachepressure=50
```

Néhány alkalmazás, mint a Firefox és a Java gyakran ír a home könyvtárba. Ez is fölöslegesen csökkenti a szilárdtest-meghajtó élettartamát. Keress rá a „firefox és SSD” szavakra, hogy hogyan változtathatod az alkalmazásokat SSD-barátokká.

8. lépés: állítsd be a kétlépéses boot menüt

Az ötletet a Full Circle Magazine 88-as száma „Hogyan: GRUB2 és multiboot 4. rész” című cikkből merítettem. A központi boot partíciónk a **/boot/central** megegyezik a cikkbeli **/mnt/GRUBpart/boot**-tal. Vedd figyelembe, hogy az UEFI-ben sok mindent meg kell változtatni ehhez.

Először add a chainloader (lánc-betöltő) bejegyzéseket a **/etc/grub.d/40_custom**-hoz.

Az UEFI nélküli használathoz a chainloader bejegyzések nagyon egyszerűek. Ha a partíció grub neve **hd0,msdos1**, ahová ugrani akarsz (a partíció a **/boot**-ba, vagy a **/**-ba van csatolva, amennyiben nincs önálló boot partíció), akkor a chainloader-bejegyzés így néz ki :

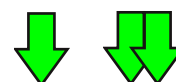
```
menuentry 'Ubuntu' {
    set root='hd0,msdos1'
    chainloader +1
}
```

Javaslat: ellenőrizd a **/boot/grub/grub.cfg**-t, hogy a partíciók milyen nevet kaptak a grub-ban. Használd a grub-héjat, ha kétségeid vannak a pontos nevekkel kapcsolatban: indítsd újra a PC-t, menj a BIOS-ba, kapcsold ki a secure boot-ot, mentsd és lépj ki, majd nyomj Esc-et, amikor a grub menüt látod. Most már bevihetsz olyan parancsot, mint **dir (hd0,msdos1)**, hogy lásd az adott partíció tartalmát: ez segíthet ellenőrizni a feltételezett grub-név helyességét.

Egy UEFI-s chain loader bejegyzés sokkal bonyolultabb. A következőképpen néz ki, ha a partíció neve **hd0,gpt5**, ahová ugrani akarsz (a partíció a **/boot**-ba, vagy a **/**-ba van csatolva, ha nincs önálló boot partíció) és ha **hd2,gpt** az EFI partíció grub-neve:

```
menuentry 'Kubuntu 14.04 amd64 op /dev/sda9' {
    insmod partgpt
    insmod chain
    set root='hd0,gpt5'
    set prefix=($root)/grub
    configfile $prefix/grub.cfg
    set efiroot='hd2,gpt2'
    chainloader ($efi_root)/EFI/ubuntu/grubx64.efi
}
```

Megjegyzés: ellenőrizd és javítsd a többi fájlhoz mutató útvonalat, ha más disztrót, illetve partíciós sémát használasz!



A grub-ban talán jobb a lemezek számozása (hdx) helyett uuid-t használni, mivel az előbbi megváltozhat, ha például USB-kulccsal indítasz. Használd a **sudo blkid /dev/sda5** parancsot, hogy megkapd a /dev/sda5 partíció uuid-jét. A chainloader bejegyzés a következőképpen néz ki (ne feledd a megfelelő hint értékek bevitelét se):

```
menuentry 'Kubuntu 14.04 amd64 op /dev/sda9' {
    insmod part_gpt
    insmod chain
    set root='hd0,gpt5'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,gpt5 --hint-efi=hd0,gpt5 --hint-baremetal=ahci0,gpt5 f65f89ac-b2b0-4345-949a-6965e3513db3
    else
        search --no-floppy --fs-uuid --set=root f65f89ac-b2b0-4345-949a-6965e3513db3
    fi
    set prefix=($root)/grub
    configfile $prefix/grub.cfg
    set efi_root='hd2,gpt2'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=efi_root --hint-bios=hd2,gpt2 --hint-efi=hd2,gpt2 --hint-baremetal=ahci2,gpt2 EC4E-2E34
    else
        search --no-floppy --fs-uuid --set=efi_root EC4E-2E34
    fi
    chainloader ($efi_root)/EFI/ubuntu/grubx64.efi
}
```

Adj egy chainloader bejegyzést a Kubuntu-hoz ugráshoz, és másold be a „Windows 10” és „System Setup” bejegyzéseket /boot/grub/grub.cfg-ből a /etc/grub.d/40_custom-ba.

Futtasd a **sudo update-grub**-t, indítsd újra és ellenőrizd, hogy az új bejegyzések működnek-e. Javítsd, ha kell.

Készítsd el a központi boot könyvtárat

```
sudo mkdir /boot/central/efi
sudo mount -o bind /boot/efi /boot/central/efi
sudo cp /boot/efi/EFI/ubuntu/grub.cfg /boot/efi/EFI/ubuntu/grub.cfg.bak
sudo grub-install /dev/sda --boot-directory=/boot/central
```

Készíts egy másolatot a grub.cfg-ről, generáld először a központi boot loader-t és azután ismét a Kubuntu-t.

```
sudo cp /boot/grub/grub.cfg /boot/grub/grub.cfg.bak
cd /etc/grub.d
sudo chmod -x linux
        mem
        prober
        uefi
sudo update-grub
sudo cp /boot/grub/grub.cfg /boot/central/grub/
sudo cp /boot/grub/unicode.pf2 /boot/central/grub/
sudo chmod +x linux
sudo chmod -x custom
sudo update-grub
```



Készítsd el a központi boot loader-t és 2 új boot bejegyzést, a **centralgrub**-ot és a **centralshim**-et. A **-d** opció jelzi a lemezt, ami az efi partíciót és a **-p** adja az efi partíció számát. Az efi partíció esetembe a **/dev/sdb2**, így ezt kapjuk:

```
sudo cp -R /boot/efi/EFI/ubuntu /boot/efi/EFI/central
sudo mv /boot/efi/EFI/ubuntu/grub.cfg.bak /boot/efi/EFI/ubuntu/grub.cfg
sudo rm /boot/efi/EFI/central/grub.cfg.bak
sudo efibootmgr -c -l \EFI\central\grubx64.efi -L centralgrub -d /dev/sdb -p 2
sudo efibootmgr -c -l \EFI\central\shimx64.efi -L centralshim -d /dev/sdb -p 2
```

Ellenőrizd az **/boot/efi/EFI/central/grub.cfg** tartalmát és a **/boot/efi/EFI/ubuntu/grub.cfg**-et. Ezen fájlok tartalma a következő:

```
search.fs_uuid 5b686b70-7fdf-495c-afa8-33847392b06f root hd0,gpt1
set prefix=($root)'/grub'
configfile $prefix/grub.cfg
```

Gondoskodj arról, hogy az **uuid** és a **root** megfelelően a központi boot partícióra, illetve a **kubuntu** boot partíciójára mutasson. Javítsd, ha kell.

9. lépés: telepítsd a többi disztrót

A következő telepített partícióm Debian volt, de ennél előbb ki kellett kapcsolni a BIOS-ban a **secure boot**-ot.

A telepítés hasonlít az első disztró telepítéséhez, attól eltekintve, hogy most már nem kell a partíciók elkészítésével bajlódni többé. Gondoskodj arról, hogy a **/boot/central**, a **tmp** és a **swap** partíciókat **ne formázza** le újból.

Egyes disztrók mindenképpen formázzák a **swap** partíciót. Ekkor a partíció új **uuid**-t kap. Ebben az esetben javítani kell a **swap** partíció **uuid**-jét a **/etc/fstab** fájlban a többi disztróban.

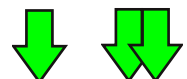
Most add a **/boot/central/grub/grub.cfg**-hez az új telepítés **chainloader** bejegyzését. Figyelj oda, hogy a Debian-hoz az útvonalat megfelelően le kell cserélni: az **ubuntu/grubx64.efi**-t **debian/grubx64.efi**-re.

Indíts újra, lépj be a BIOS-ba, engedélyezd a **secure boot**-ot és állítsd a **central_shim**-et boot loadernek.

Vége

Most már könnyen kell tudnod váltani **Kubuntu**, **Debian** és **Windows 10** között. Élvezd ki.

Utolsó megjegyzés: *grub splash* kép a **Secure Boot**-tal nem kombinálható, mivel a képfájlokat a BIOS aláírás hiányában nem tekinti biztonságosnak.



Frank 1986-ban végzett mérnökként és most nagy Internet router-ek számára fejleszt szoftvert. Otthon és munkában is egyaránt Linuxot használ. Nagyon érdekli olyasmi, mint a Linux tényleges működése a felszín alatt.

